Foundations and Trends® in Signal Processing

1

Safety at scale: a comprehensive survey of large model and agent safety

Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, Hengyuan Xu, Yunhao ChenYunhan Zhao, Hanxun Huang, Yige Li, Yutao Wu, Jiaming Zhang, Xiang Zheng, Yang Bai, Yiming Li, Zuxuan WuXipeng Qiu, Jingfeng Zhang, Xudong HanHaonan Li, Jun Sun, Cong Wang, Jindong Gu, Baoyuan Wu, Siheng Chen, Tianwei ZhangYang Liu, Mingming Gong, Tongliang Liu, Shirui Pan, Cihang Xie, Tianyu Pang, Yinpeng Dong, Ruoxi Jia, Yang Zhang, Shiqing Ma, Xiangyu Zhang, Neil Gong, Chaowei Xiao, Sarah Erfani, Tim Baldwin, Bo Li, Masashi Sugiyama, Dacheng Tao, James Bailey and Yu-Gang Jiang

(Author affiliations can be found at the end of the article)

ABSTRACT

The rapid advancement of large models, driven by their exceptional abilities in learning and generalization through large-scale pre-training, has reshaped the landscape of Artificial Intelligence (AI). These models are now foundational to a wide range of applications, including conversational AI, recommendation systems, autonomous driving, content generation, medical diagnostics, and scientific discovery. However, their widespread deployment also exposes them to significant safety risks, raising concerns about robustness, reliability, and ethical implications. This survey provides a

Foundations and Trends® in Signal Processing
Vol. 8 No. 3-4, 2025
p. 000
Emerald Publishing Limited
2474-1558
DOI 10.1561/3300000051

systematic review of current safety research on large models, covering Vision Foundation Models (VFMs), Large Language Models (LLMs), Vision-Language Pre-training (VLP) models, Vision-Language Models (VLMs), Diffusion Models (DMs), and large-model-powered Agents. Our contributions are summarized as follows: (1) We present a comprehensive taxonomy of safety threats to these models, including adversarial attacks, data poisoning, backdoor attacks, jailbreak and prompt injection attacks, energy-latency attacks, data and model extraction attacks, and emerging agent-specific threats. (2) We review defense strategies proposed for each type of attack, if available and summarize the commonly used datasets and benchmarks for safety research. (3) Building on this, we identify and discuss the open challenges in large model safety, emphasizing the need for comprehensive safety evaluations, scalable and effective defense mechanisms, and sustainable data practices. More importantly, we highlight the necessity of collective efforts from the research community and international collaboration. Our work can serve as a useful reference for researchers and practitioners, fostering the ongoing development of comprehensive defense systems and platforms to safeguard AI models. GitHub: https://github.com/xingjunm/Awesome-Large-Model-Safety.

1. Introduction

Artificial Intelligence (AI) has entered the era of large models, exemplified by Vision Foundation Models (VFMs), Large Language Models (LLMs), Vision-Language Pre-Training (VLP) models, Vision-Language Models (VLMs), and image/video generation diffusion models (DMs). Through large-scale pre-training on massive datasets, these models have demonstrated unprecedented capabilities in tasks ranging from language understanding and image generation to complex problem-solving and decision-making. Their ability to understand and generate human-like content (e.g., texts, images, audios, and videos) has enabled applications in customer service, content creation, healthcare, education, and more, highlighting their transformative potential in both commercial and societal domains.

However, the deployment of large models comes with significant challenges and risks. As these models become more integrated into critical applications, concerns regarding their vulnerabilities to adversarial, jailbreak, and backdoor attacks, data privacy breaches, and the generation of harmful or misleading content have intensified. These issues pose substantial threats, including unintended system behaviors, privacy leakage, and the dissemination of harmful information. Ensuring the safety of these models is paramount to prevent such unintended consequences, maintain public trust, and promote responsible AI usage. The field of AI safety research has expanded in response to these challenges, encompassing a diverse array of attack methodologies, defense strategies, and evaluation benchmarks designed to identify and mitigate the vulnerabilities of large models. Given the rapid development of safety-related techniques for various large models, we aim to provide a

comprehensive survey of these techniques, highlighting strengths, weaknesses, and gaps, while advancing research and fostering collaboration.

Given the broad scope of our survey, we have structured it with the following considerations to enhance clarity and organization:

- *Models*. We focus on six widely studied model categories, including *VFMs*, *LLMs*, *VLPs*, *VLMs*, *DMs*, and *Agents*, and review the attack and defense methods for each separately. These models represent the most popular large models across various domains.
- *Organization*. For each model category, we classify the reviewed works into attacks and defenses, and identify *10* attack types: *adversarial*, *backdoor*, *poisoning*, *jailbreak*, *prompt injection*, *energy-latency*, *membership inference*, *model extraction*, *data extraction*, and *agent* attacks. When both backdoor and poisoning attacks are present for a model category, we combine them into a single *backdoor & poisoning* category due to their similarities. We review the corresponding defense strategies for each attack type immediately after the attacks.
- *Taxonomy.* For each type of attack or defense, we use a two-level taxonomy: *Category* → *Subcategory.* The *Category* differentiates attacks and defenses based on the threat model (e.g., white-box, gray-box, black-box) or specific subtasks (e.g., detection, purification, robust training/tuning, and robust inference). The *Subcategory* offers a more detailed classification based on their techniques.
- *Granularity.* To ensure clarity, we simplify the introduction of each reviewed paper, highlighting only its key ideas, objectives, and approaches, while omitting technical details and experimental analyses.

Our survey methodology is structured as follows. First, we conducted a keyword-based search targeting specific model types and threat types to identify relevant papers. Next, we manually filtered out non-safety-related and non-technical papers. For each remaining paper, we categorized its proposed method or framework by analyzing its settings and attack/defense types, assigning them to appropriate categories and subcategories. In total, we reviewed *574* technical papers, with their distribution across years, model types, and attack/defense strategies illustrated in Figure 1. As shown, safety research on large models has surged significantly since 2023, following the release of ChatGPT. Among the model types, LLMs, DMs and Agents have garnered the most attention, accounting for *71.32%* of the surveyed papers.

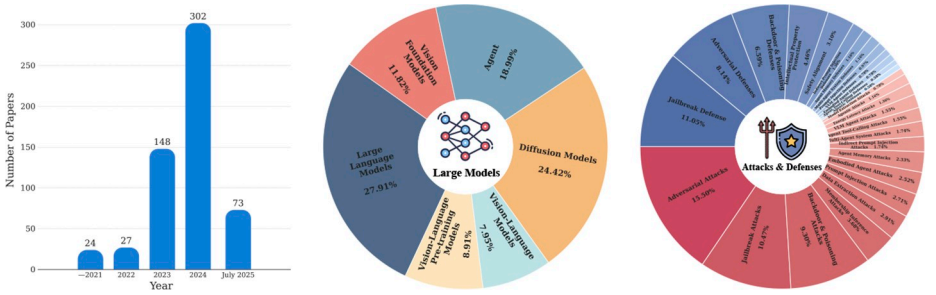

Figure 1. *Left*: The number of surveyed technical papers on attacks, defenses, and benchmarks/datasets. *Middle*: Distribution of surveyed technical papers by model type. *Right*: Distribution of surveyed technical papers by attack and defense type

Regarding attack types, *jailbreak*, *adversarial*, and *backdoor* attacks were the most extensively studied. On the defense side, *jailbreak defenses* received the highest focus, followed by *adversarial defenses*. Figure 2 presents a cross-view of temporal trends across model types and attack/defense categories, offering a detailed breakdown of the reviewed works. Notably, research on attacks constitutes $\sim$ *60%* of the studied. In terms of defense, while defense research accounts for only $\sim$ *40%*, underscoring a

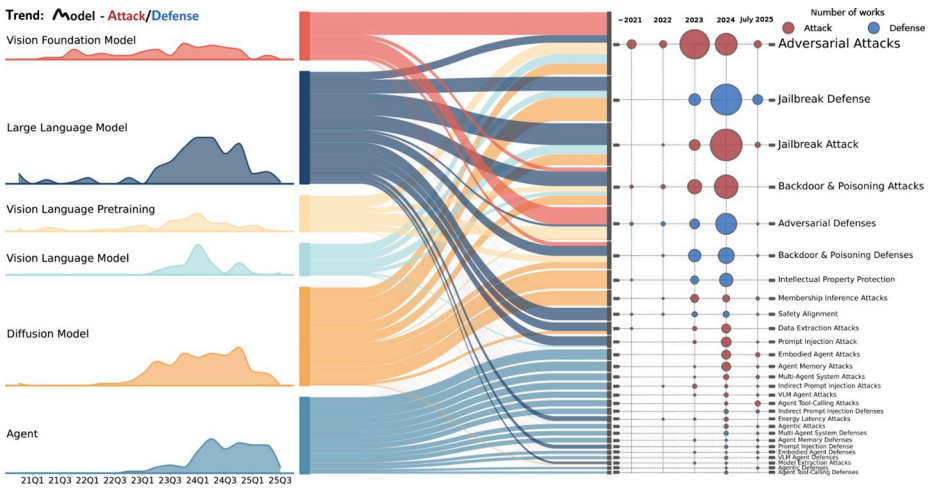

Figure 2. *Left*: The quarterly trend in the number of surveyed safety papers across different models; *Middle*: Proportional distribution of attack and defense studies associated with large models. *Right*: Annual trend in the number of surveyed safety papers on various attacks and defenses, ordered from most to least studied

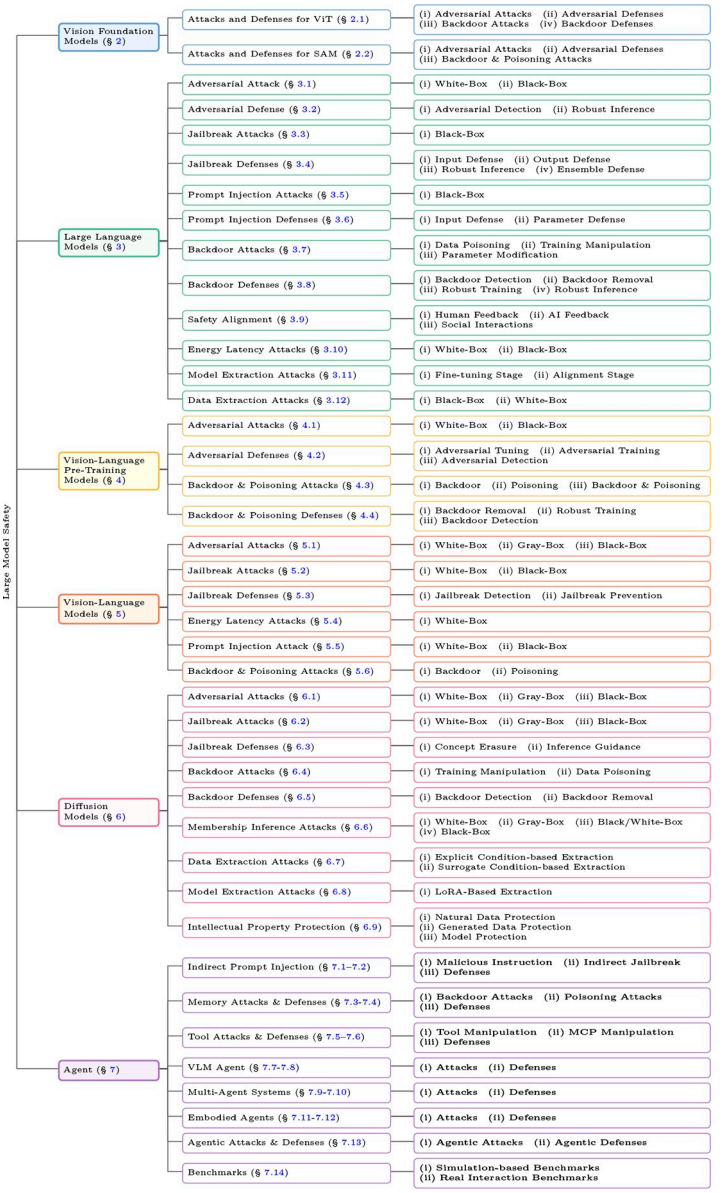

Figure 3. Organization of this survey

Table 1. A summary of existing surveys

Survey	Year	VFM	VLP	LLM	VLM	DM	LLM-Agent	VLM-Agent
Zhang *et al.* 2024c	2024	✓	✓	✗	✗	✓	✗	✗
Truong *et al.* 2024	2024	✗	✗	✗	✗	✓	✗	✗
Zhao *et al.* 2024b	2024	✗	✗	✓	✗	✗	✗	✗
Yi *et al.* 2024	2024	✗	✗	✓	✗	✗	✗	✗
Jin *et al.* 2024	2024	✗	✗	✓	✓	✗	✗	✗
Liu *et al.* 2024o	2024	✗	✗	✗	✓	✗	✗	✗
Liu *et al.* 2024c	2024	✗	✗	✗	✓	✗	✓	✗
Cui *et al.* 2024a	2024	✗	✗	✓	✗	✗	✓	✗
Gan *et al.* 2024	2024	✗	✗	✓	✓	✗	✓	✓
Deng *et al.* 2024c	2025	✗	✗	✓	✗	✗	✓	✗
Ye *et al.* 2025	2025	✗	✗	✗	✓	✗	✗	✗
Wang *et al.* 2025a	2025	✗	✗	✓	✗	✗	✓	✗
Our survey	2025	✓	✓	✓	✓	✓	✓	✓

significant gap that warrants increased attention toward defense strategies. The overall structure of this survey is outlined in Figure 3.

Difference to Existing Surveys. Large-model safety is a rapidly evolving field, and several surveys have been conducted to advance research in this area. Recently, Slattery *et al.* (2024) introduced an AI risk framework with a systematic taxonomy covering all types of risks. In contrast, our focus is on the technical aspects, specifically the attack and defense techniques proposed in the literature. Table 1 summarizes the technical surveys we identified, each concentrating on a few model types or threat categories (e.g., LLMs, VLMs, agents, or jailbreak attacks/defenses). Compared with these works, our survey provides both a broader scope by covering a wider range of model types and threats, and a more high-level perspective that focuses on overarching methodologies rather than specific technical details.

2. Vision foundation model safety

This section surveys safety research on two types of VFMs: per-trained Vision Transformers (ViTs) (Dosovitskiy *et al.*, 2021) and the Segment Anything Model (SAM) (Kirillov *et al.*, 2023). We focus on ViTs and SAM because they are among the most widely deployed VFMs and have garnered significant attention in recent safety research.

2.1 Attacks and defenses for ViTs

Pre-trained ViTs are widely employed as backbones for various downstream tasks, frequently achieving state-of-the-art performance through efficient adaptation and fine-tuning. Unlike traditional CNNs, ViTs process images as sequences of tokenized patches, allowing them to better capture spatial dependencies. However, this patch-based mechanism also brings unique safety concerns and robustness challenges. This section explores these issues by reviewing ViT-related safety research, including adversarial attacks, backdoor & poisoning attacks, and their corresponding defense strategies. Table 2 provides a summary of the surveyed attacks and defenses, along with the commonly used datasets.

2.1.1 Adversarial attacks. Adversarial attacks on ViTs can be classified into *white-box attacks* and *black-box attacks* based on whether the attacker has full access to the victim model. Based on the attack strategy, white-box attacks can be further divided into 1) *patch attacks*, 2) *position embedding attacks* and 3) *attention attacks*, while black-box attacks can be summarized into 1) *transfer-based attacks* and 2) *query-based attacks*.

Table 2. A summary of attacks and defenses for ViTs and SAM

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
Attacks and defenses for ViT (Section 2.1)						
Adversarial Attack	Patch-Fool (Fu *et al.*, 2022)	2022	White-box	Patch Attack	DeiT, ResNet	ImageNet
	SlowFormer (Navaneet *et al.*, 2024)	2024	White-box	Patch Attack	ATS, AdaViT	ImageNet
	PE-Attack (Gao *et al.*, 2024g)	2024	White-box	Position Embedding Attack	ViT, DeiT, BEiT	ImageNet, GLUE, wmt13/16, Food-101, CIFAR100, etc.
	Attention-Fool (Lovisotto *et al.*, 2022)	2022	White-box	Attention Attack	ViT, DeiT, DETR	ImageNet
	AAS (Jain and Dutta, 2024)	2024	White-box	Attention Attack	ViT-B	ImageNet, CIFAR10/100
	SE-TR (Naseer *et al.*, 2021)	2022	Black-box	Transfer-based Attack	DeiT, T2T, TnT, DINO, DETR	ImageNet
	ATA (Wang *et al.*, 2022a)	2022	Black-box	Transfer-based Attack	ViT, DeiT, ConViT	ImageNet
	PNA-PatchOut (Wei *et al.*, 2022)	2022	Black-box	Transfer-based Attack	ViT, DeiT, TNT, LeViT, PiT, CaiT, ConViT, Visformer	ImageNet
	LPM (Wei and Zhao, 2023)	2023	Black-box	Transfer-based Attack	ViT, PiT, DeiT, Visformer, LeViT, ConViT	ImageNet

(*continued*)

Table 2. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	MIG (Ma *et al.*, 2023)	2023	Black-box	Transfer-based Attack	ViT, TNT, Swin	ImageNet
	TGR (Zhang *et al.*, 2023c)	2023	Black-box	Transfer-based Attack	DeiT, TNT, LeViT, ConViT	ImageNet
	VDC (Zhang *et al.*, 2024k)	2024	Black-box	Transfer-based Attack	CaiT, TNT, LeViT, ConViT	ImageNet
	FDAP (Gao *et al.*, 2024a)	2024	Black-box	Transfer-based Attack	ViT, DeiT, CaiT, ConViT, TNT	ImageNet
	SASD-WS (Wu *et al.*, 2024d)	2024	Black-box	Transfer-based Attack	ViT, ResNet, DenseNet, VGG	ImageNet
	CRFA (Li *et al.*, 2024aa)	2024	Black-box	Transfer-based Attack	ViT, DeiT, CaiT, TNT, Visformer, LeViT, ConvNeXt, RepLKNet	ImageNet
	FPR Ren *et al.* 2025	2025	Black-box	Transfer-based Attack	ViT, CaiT, PiT, Visformer, Swin, DeiT, CoaT, ResNet, VGG, DenseNet	ImageNet
	PAR (Shi *et al.*, 2022)	2022	Black-box	Query-based Attack	ViT	ImageNet
Adversarial Defense	AGAT (Wu *et al.*, 2022a)	2022	Adversarial Training	Efficient training	ViT, CaiT, LeViT	ImageNet

(*continued*)

Table 2. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	ARD-PRM (Mo *et al.*, 2022)	2022	Adversarial Training	Efficient training	ViT, DeiT, ConViT, Swin	ImageNet, CIFAR10
	Patch-Vestiges (Li, 2022)	2022	Adversarial Detection	Patch-based Detection	ViT, ResNet	CIFAR10
	ViTGuard (Sun *et al.*, 2024a)	2024	Adversarial Detection	Attention-based Detection	ViT	ImageNet, CIFAR10/100
	ARMRO (Liu *et al.*, 2023d)	2023	Adversarial Detection	Attention-based Detection	ViT, DeiT	ImageNet, CIFAR10
	Smoothed-Attention (Gu *et al.*, 2022)	2022	Robust Architecture	Robust Attention	DeiT, ResNet	ImageNet
Adversarial Defense	TAP (Guo *et al.*, 2023a)	2023	Robust Architecture	Robust Attention	RVT, FAN	ImageNet, Cityscapes, COCO
	RSPC (Guo *et al.*, 2023b)	2023	Robust Architecture	Robust Attention	RVT, FAN	ImageNet, CIFAR10/100
	FViT (Hu *et al.*, 2024b)	2024	Robust Architecture	Robust Attention	ViT, DeiT, Swin	ImageNet, Cityscapes, COCO
	SATA Nikzad *et al.* 2025	2025	Robust Architecture	Robust Attention	ViT, DeiT	ImageNet
	ADBM Li *et al.* 2024q	2024	Adversarial Purification	Diffusion-based Purification	WideResNet, ViT	CIFAR-10, ImageNet, SVHN

(*continued*)

Table 2. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	CGDMP (Bai *et al.*, 2024b)	2024	Adversarial Purification	Diffusion-based Purification	ResNet, XciT	CIFAR 10/100, GTSRB, ImageNet
	ADBM (Li *et al.*, 2024q)	2024	Adversarial Purification	Diffusion-based Purification	WideResNet, ViT	CIFAR-10, ImageNet, SVHN
	OSCP (Lei *et al.*, 2024)	2024	Adversarial Purification	Diffusion-based Purification	ViT, Swin, WideResNet	ImageNet, CelebA-HQ
Backdoor Attack	BadViT (Yuan *et al.*, 2023b)	2023	Data Poisoning	Patch-level Attack	DeiT, LeViT	ImageNet
	TrojViT (Zheng *et al.*, 2023)	2023	Data Poisoning	Patch-level Attack	DeiT, ViT, Swin	ImageNet, CIFAR10
	SWARM (Yang *et al.*, 2024d)	2024	Data Poisoning	Token-level Attack	ViT	VTAB-1k
	DBIA (Lv *et al.*, 2023)	2023	Data Poisoning	Data-free Attack	ViT, DeiT, Swin	ImageNet, CIFAR10/100, GTSRB, GGFace
	MTBA (Li *et al.*, 2024x)	2024	Data Poisoning	Multi-trigger Attack	ViT	ImageNet, CIFAR10
Backdoor defense	PatchDrop (Doan *et al.*, 2023)	2023	Robust Inference	Patch Processing	ViT, DeiT, ResNet	ImageNet, CIFAR10
	Image Blocking (Subramanya *et al.*, 2024)	2023	Robust Inference	Image Blocking	ViT, CaiT	ImageNet

(*continued*)

Table 2. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
Attacks and defenses for SAM (Section 2.2)						
Adversarial Attack	S-RA (Shen *et al.*, 2024b)	2024	White-box	Prompt-agnostic Attack	SAM	SA-1B
	Croce and Hein (2024)	2024	White-box	Prompt-agnostic Attack	SAM, SEEM	SA-1B
	Attack-SAM (Zhang *et al.*, 2023b)	2023	Black-box	Transfer-based Attack	SAM	SA-1B
	PATA++ (Zheng and Zhang, 2023)	2023	Black-box	Transfer-based Attack	SAM	SA-1B
	UAD (Lu *et al.*, 2024b)	2024	Black-box	Transfer-based Attack	SAM, FastSAM	SA-1B
	T-RA (Shen *et al.*, 2024b)	2024	Black-box	Transfer-based Attack	SAM	SA-1B
	UMI-GRAT (Xia *et al.*, 2024)	2024	Black-box	Transfer-based Attack	Medical SAM, Shadow-SAM, Camouflaged-SAM	CT-Scans, ISTD, COD10K, CAMO, CHAME
Adversarial Attack	Han *et al.* (2023b)	2023	Black-box	Universal Attack	SAM	SA-1B
	DarkSAM (Zhou *et al.*, 2024i)	2024	Black-box	Universal Attack	SAM, HQ-SAM, PerSAM	ADE20K, Cityscapes, COCO, SA-1B

(*continued*)

Table 2. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
Adversarial Defense	ASAM (Li *et al.*, 2024a)	2024	Adversarial Tuning	Diffusion Model-based Tuning	SAM	Ade20k, VOC2012, COCO, DOORS, LVIS, etc.
	Robust SAM Long *et al.* 2025	2024	Adversarial Tuning	Parameter-Efficient Fine-Tuning	SAM, MedSAM, SAM-Adapter	SA-1B, VOC, COCO, DAVIS
Backdoor& Poisoning Attack	BadSAM (Guan *et al.*, 2024)	2024	Data Poisoning	Visual trigger	SAM	CAMO
	UnSeg (Sun *et al.*, 2024b)	2024	Data Poisoning	Unlearnable Examples	HQ-SAM, DINO, Rsprompter, UNet++, Mask2Former, DeepLabV3	Cityscapes, VOC, COCO, Lung, Kvasir-seg, WHU, etc.

White-box Attacks. Patch Attacks exploit the modular structure of ViTs, aiming to manipulate their inference processes by introducing targeted perturbations in specific patches of the input data. Joshi *et al.* (2021) proposed an adversarial token attack method leveraging block sparsity to assess the vulnerability of ViTs to token-level perturbations. Expanding on this, *Patch-Fool* (Fu *et al.*, 2022) introduces an adversarial attack framework that targets the self-attention modules by perturbing individual image patches, thereby manipulating attention scores. Different from existing methods, *SlowFormer* (Navaneet *et al.*, 2024) introduces a universal adversarial patch can be applied to any image to increases computational and energy costs while preserving model accuracy.

Position Embedding Attacks aim to attack the spatial or sequential position of tokens in transformers. For example, *PE-Attack* (Gao *et al.*, 2024g) explores the common vulnerability of positional embeddings to adversarial perturbations by disrupting their ability to encode positional information through periodicity manipulation, linearity distortion, and optimized embedding distortion.

Attention Attacks target vulnerabilities in the self-attention modules of ViTs. *Attention-Fool* (Lovisotto *et al.*, 2022) manipulates dot-product similarities to redirect queries to adversarial key tokens, exposing the model's sensitivity to adversarial patches. Similarly, *AAS* (Jain and Dutta, 2024) mitigates gradient masking in ViTs by optimizing the pre-softmax output scaling factors, enhancing the effectiveness of attacks.

Black-box Attacks. *Transfer-based Attacks* first generate adversarial examples using fully accessible surrogate models, which are then transferred to attack black-box victim ViTs. In this context, we first review attacks specifically designed for the ViT architecture. *SE-TR* (Naseer *et al.*, 2021) enhances adversarial transferability by optimizing perturbations on an ensemble of models. *ATA* (Wang *et al.*, 2022a) strategically activates uncertain attention and perturbs sensitive embeddings within ViTs. *LPM* (Wei and Zhao, 2023) mitigates the overfitting to model-specific discriminative regions through a patch-wise optimized binary mask. Chen *et al.* (2023f) introduced an Inductive Bias Attack (*IBA*) to suppress unique biases in ViTs and target shared inductive biases. *TGR* (Zhang *et al.*, 2023c) reduces the variance of the backpropagated gradient within internal blocks. *VDC* (Zhang *et al.*, 2024k) employs virtual dense connections between deeper attention maps and MLP blocks to facilitate gradient backpropagation. *FDAP* (Gao *et al.*, 2024a) exploits feature collapse by reducing high-frequency components in feature space. *CRFA* (Li *et al.*, 2024aa) disrupts only the most crucial image regions using approximate attention maps. *SASD-*

WS (Wu *et al.*, 2024d) flattens the loss landscape of the source model through sharpness-aware self-distillation and approximates an ensemble of pruned models using weight scaling to improve target adversarial transferability. *FPR* (Ren *et al.*, 2025) improves the adversarial transferability on ViTs by refining forward propagation instead of backward gradients. It consists of two components: (1) Attention Map Diversification (AMD), which applies controlled randomness to diversify attention maps, mitigating overfitting and implicitly inducing gradient vanishing; and (2) Momentum Token Embedding (MTE), which stabilizes token embedding updates by accumulating historical embeddings across iterations.

Other strategies are applicable to both ViTs and CNNs, ensuring broader applicability in black-box settings. Wei *et al.* (2022, 2023) proposed a dual attack framework to improve transferability between ViTs and CNNs: 1) a Pay No Attention (*PNA*) attack, which skips the gradients of attention during backpropagation, and 2) a *PatchOut* attack, which randomly perturbs subsets of image patches at each iteration. *MIG* (Ma *et al.*, 2023) uses integrated gradients and momentum-based updates to precisely target model-agnostic critical regions, improving transferability between ViTs and CNNs.

Query-based Attacks generate adversarial examples by querying the black-box model and levering the model responses to estimate the adversarial gradients. The goal is to achieve successful attack with a minimal number of queries. Based on the type of model response, query-based attacks can be further divided into score-based attacks, where the model returns a probability vector, and decision-based attacks, where the model provides only the top-k classes. Decision-based attacks typically start from a large random noise (to achieve misclassification first) and then gradually find smaller noise while maintaining misclassification. To improve the efficiency of the adversarial noise searching process in ViTs, *PAR* (Shi *et al.*, 2022) introduces a coarse-to-fine patch searching method, guided by noise magnitude and sensitivity masks to account for the structural characteristics of ViTs and mitigate the negative impact of non-overlapping patches.

2.1.2 Adversarial defenses. Adversarial defenses for ViTs follow four major approaches: 1) *adversarial training*, which trains ViTs on adversarial examples via min-max optimization to improve its robustness; 2) *adversarial detection*, which identifies and mitigates adversarial attacks by detecting abnormal or malicious patterns in the inputs; 3) *robust architecture*, which modifies and optimizes the architecture (e.g., self-attention module) of ViTs to improve their resilience against adversarial attacks; and 4) *adversarial purification*, which pre-processes the input

(e.g., noise injection, denoising, or other transformations) to remove potential adversarial perturbations before inference.

Adversarial Training is widely regarded as the most effective approach to adversarial defense; however, it comes with a high computational cost. To address this on ViTs, *AGAT* (Wu *et al.*, 2022a) introduces a dynamic attention-guided dropping strategy, which accelerates the training process by selectively removing certain patch embeddings at each layer. This reduces computational overhead while maintaining robustness, especially on large datasets such as ImageNet. Due to its high computational cost, research on adversarial training for ViTs has been relatively limited. *ARD-PRM* (Mo *et al.*, 2022) improves adversarial robustness by randomly dropping gradients in attention blocks and masking patch perturbations during training.

Adversarial Detection methods for ViTs primarily leverage two key features, i.e., patch-based inference and activation characteristics, to detect and mitigate adversarial examples. Li (2022) proposed the concept of *Patch Vestiges*, abnormalities arising from adversarial examples during patch division in ViTs. They used statistical metrics on step changes between adjacent pixels across patches and developed a binary regression classifier to detect adversaries. Alternatively, *ARMOR* (Liu *et al.*, 2023d) identifies adversarial patches by scanning for unusually high column scores in specific layers and masking them with average images to reduce their impact. *ViTGuard* (Sun *et al.*, 2024a), on the other hand, employs a masked autoencoder to detect patch attacks by analyzing attention maps and CLS token representations. As more attacks are developed, there is a growing need for a unified detection framework capable of handling all types of adversarial examples.

Robust Architecture methods focus on designing more adversarially resilient attention modules for ViTs. For example, *Smoothed Attention* (Gu *et al.*, 2022) employs temperature scaling in the softmax function to prevent any single patch from dominating the attention, thereby balancing focus across patches. *ReiT* (Gong *et al.*, 2024b) integrates adversarial training with randomization through the II-ReSA module, optimizing randomly entangled tokens to reduce adversarial similarity and enhance robustness. *TAP* (Guo *et al.*, 2023a) addresses token overfocusing by implementing token-aware average pooling and an attention diversification loss, which incorporate local neighborhood information and reduce cosine similarity among attention vectors. *FViTs* (Hu *et al.*, 2024b) strengthen explanation faithfulness by stabilizing top-k indices in self-attention and robustify predictions using denoised diffusion smoothing combined with Gaussian noise. *RSPC* (Guo *et al.*, 2023b) tackles

vulnerabilities by corrupting the most sensitive patches and aligning intermediate features between clean and corrupted inputs to stabilize the attention mechanism. Collectively, these advancements underscore the pivotal role of the attention mechanism in improving the adversarial robustness of ViTs. *SATA* (Nikzad *et al.*, 2025) robustifies ViT models without retraining by injecting a spatial autocorrelation-based module between attention and FFN layers. It splits tokens by their spatial autocorrelation scores and selectively merges high/low-score tokens before FFN, reducing redundancy and improving feature aggregation. Residual tokens are later concatenated to preserve information, offering strong robustness against adversarial and corrupted inputs.

Adversarial Purification refers to a model-agnostic input-processing technique that is broadly applicable across various architectures, including but not limited to ViTs. *DiffPure* (Nie *et al.*, 2022) introduces a framework where adversarial images undergo noise injection via a forward stochastic differential equation (SDE) process, followed by denoising with a pre-trained diffusion model. *CGDMP* (Bai *et al.*, 2024b) refines this approach by optimizing the noise level for the forward process and employing contrastive loss gradients to guide the denoising process, achieving improved purification tailored to ViTs. *ADBM* (Li *et al.*, 2024q) highlights the disparity between diffused adversarial and clean examples, proposing a method to directly connect the clean and diffused adversarial distributions. While these methods focus on ViTs, other approaches demonstrate broader applicability to various vision models, e.g., CNNs. *Purify++* (Zhang *et al.*, 2023a) enhances DiffPure with improved diffusion models, *DifFilter* (Chen *et al.*, 2024b) extends noise scales to better preserve semantics, and *MimicDiffusion* (Song *et al.*, 2024b) mitigates adversarial impacts during the reverse diffusion process. For improved efficiency, *OSCP* (Lei *et al.*, 2024) and *LightPure* (Khalili *et al.*, 2024) propose single-step and real-time purification methods, respectively. *LoRID* (Zollicoffer *et al.*, 2024) introduces a Markov-based approach for robust purification. These methods complement ViT-related research and highlight diverse advancements in adversarial purification.

2.1.3 Backdoor attacks. Backdoors can be injected into the victim model via data poisoning, training manipulation, or parameter editing, with most existing attacks on ViTs being data poisoning-based. We classify these attacks into four categories: 1) *patch-level attacks*, 2) *token-level attacks*, and 3) *multi-trigger attacks*, which exploit ViT-specific data processing characteristics, as well as 4) *data-free attacks*, which exploit the inherent mechanisms of ViTs.

Patch-level Attacks primarily exploit the ViT's characteristic of processing images as discrete patches by implanting triggers at the patch level. For example, *BadViT* (Yuan *et al.*, 2023b) introduces a universal patch-wise trigger that requires only a small amount of data to redirect the model's focus from classification-relevant patches to adversarial triggers. *TrojViT* (Zheng *et al.*, 2023) improves this approach by utilizing patch salience ranking, an attention-targeted loss function, and parameter distillation to minimize the bit flips necessary to embed the backdoor.

Token-level Attacks target the tokenization layer of ViTs. *SWARM* (Yang *et al.*, 2024d) introduces a switchable backdoor mechanism featuring a "switch token" that dynamically toggles between benign and adversarial behaviors, ensuring high attack success rates while maintaining functionality in clean environments.

Multi-trigger Attacks employ multiple backdoor triggers in parallel, sequential, or hybrid configurations to poison the victim dataset. *MTBAs* (Li *et al.*, 2024x) utilize these multiple triggers to induce coexistence, overwriting, and cross-activation effects, significantly diminishing the effectiveness of existing defense mechanisms.

Data-free Attacks eliminate the need for original training datasets. Using substitute datasets, *DBIA* (Lv *et al.*, 2023) generates universal triggers that maximize attention within ViTs. These triggers are fine-tuned with minimal parameter adjustments using PGD (Madry *et al.*, 2018), enabling efficient and resource-light backdoor injection.

2.1.4 Backdoor defenses. Backdoor defenses for ViTs aim to identify and break (or remove) the correlation between trigger patterns and target classes while preserving model accuracy. Two representative defense strategies are: 1) *patch processing*, which disrupts the integrity of image patches to prevent trigger activation, and 2) *image blocking*, which leverages interpretability-based mechanisms to mask and neutralize the effects of backdoor triggers.

Patch Processing strategy disrupts the integrity of patches to neutralize triggers. Doan *et al.* (2023) found that clean-data accuracy and attack success rates of ViTs respond differently to patch transformations before positional encoding, and proposed an effective defense method by randomly dropping or shuffling patches of an image to counter both patch-based and blending-based backdoor attacks. *Image Blocking* utilizes interpretability to identify and neutralize triggers. Subramanya *et al.* (2022) showed that ViTs can localize backdoor triggers using attention maps and proposed a defense mechanism that dynamically masks potential trigger regions during inference. In a subsequent work, Subramanya *et al.* (2024) proposed to integrate trigger neutralization into the training phase to improve the robustness of ViTs to backdoor attacks. While these two methods are promising, the field requires a holistic defense

framework that integrates non-ViT defenses with ViT-specific characteristics and unifies multiple defense tasks including backdoor detection, trigger inversion, and backdoor removal, as attempted in Li *et al.* (2024v).

2.1.5 Datasets. Datasets are crucial for developing and evaluating attack and defense methods. Table 2 summarizes the datasets used in adversarial and backdoor research.

Datasets for Adversarial Research As shown in Table 2, adversarial researches were primarily conducted on ImageNet. While attacks were tested across various datasets like CIFAR-10/100, Food-101, and GLUE, defenses were mainly limited to ImageNet and CIFAR-10/100. This imbalance reveals one key issue in adversarial research: attacks are more versatile, while defenses struggle to generalize across different datasets.

Datasets for Backdoor Research Backdoor researches were also conducted mainly on ImageNet and CIFAR-10/100 datasets. Some attacks, such as DBIA and SWARM, extend to domain-specific datasets like GTSRB and VGGFace, while defenses, including PatchDrop, were often limited to a few benchmarks. This narrow focus reduces their real-world applicability. Although backdoor defenses are shifting towards robust inference techniques, they typically target specific attack patterns, limiting their generalizability. To address this, adaptive defense strategies need to be tested across a broader range of datasets to effectively counter the evolving nature of backdoor threats.

2.2 Attacks and defenses for SAM

SAM is a foundational model for image segmentation, comprising three primary components: a ViT-based image encoder, a prompt encoder, and a mask decoder. The image encoder transforms high-resolution images into embeddings, while the prompt encoder converts various input modalities into token embeddings. The mask decoder combines these embeddings to generate segmentation masks using a two-layer Transformer architecture. Due to its complex structure, attacks and defenses targeting SAM differ significantly from those developed for CNNs. These unique challenges stem from SAM's modular and interconnected design, where vulnerabilities in one component can propagate to others, necessitating specialized strategies for both attack and defense. This section systematically reviews SAM-related adversarial attacks, backdoor & poisoning attacks, and adversarial defense strategies, as summarized in Table 2.

2.2.1 Adversarial attacks. Adversarial attacks on SAM can be categorized into: (1) *white-box attacks*, exemplified by *prompt-agnostic attacks*, and (2) *black-box attacks*, which can be further divided into *universal attacks* and *transfer-based attacks*. Each category employs distinct strategies to compromise segmentation performance.

White-box Attacks. Prompt-Agnostic Attacks are white-box attacks that disrupt SAM's segmentation without relying on specific prompts, using either *prompt-level* or *feature-level* perturbations for generality across inputs. For prompt-level attacks, Shen *et al.* (2024b) proposed a grid-based strategy to generate adversarial perturbations that disrupt segmentation regardless of click location. For feature-level attacks, Croce and Hein (2024) perturbed features from the image encoder to distort spatial embeddings, undermining SAM's segmentation integrity.

Black-box Attacks. Universal Attacks generate UAPs (Moosavi-Dezfooli *et al.*, 2017) that can consistently disrupt SAM across arbitrary prompts. Han *et al.* (2023b) exploited contrastive learning to optimize the UAPs, achieving better attack performance by exacerbating feature misalignment. *DarkSAM* (Zhou *et al.*, 2024i), on the other hand, introduces a hybrid spatial-frequency framework that combines semantic decoupling and texture distortion to generate universal perturbations.

Transfer-based Attacks exploit transferable representations in SAM to generate perturbations that remain adversarial across different models and tasks. *PATA++* (Zheng and Zhang, 2023) improves transferability by using a regularization loss to highlight key features in the image encoder, reducing reliance on prompt-specific data. *Attack-SAM* (Zhang *et al.*, 2023b) employs ClipMSE loss to focus on mask removal, optimizing for spatial and semantic consistency to improve cross-task transferability. *UMI-GRAT* (Xia *et al.*, 2024) follows a two-step process: it first generates a generalizable perturbation with a surrogate model and then applies gradient robust loss to improve across-model transferability. Apart from designing new loss functions, optimization over transformation techniques can also be exploited to improve transferability. This includes *T-RA* (Shen *et al.*, 2024b), which improves cross-model transferability by applying spectrum transformations to generate adversarial perturbations that degrade segmentation in SAM variants, and *UAD* (Lu *et al.*, 2024b), which generates adversarial examples by deforming images in a two-stage process and aligning features with the deformed targets.

2.2.2 Adversarial defenses. Adversarial defenses for SAM are currently limited, with existing approaches focusing primarily on adversarial tuning, which integrates adversarial training into the prompt tuning process of SAM. For example,

ASAM (Li *et al.*, 2024b) utilizes a stable diffusion model to generate realistic adversarial samples on a low-dimensional manifold through diffusion model-based tuning. ControlNet (Zhang *et al.*, 2023d) is then employed to guide the re-projection process, ensuring that the generated samples align with the original mask annotations. Finally, SAM is fine-tuned using these adversarial examples. *RobustSAM* (Long *et al.*, 2025) defends against adversarial attacks by adapting only 512 singular values in SAM's convolutional layers via Singular Value Decomposition (SVD), effectively altering feature distributions. This few-parameter approach achieves strong robustness–accuracy trade-off with minimal computational overhead.

2.2.3 Backdoor and poisoning attacks. Backdoor and poisoning attacks on SAM remain underexplored. Here, we review one backdoor attack that leverages perceptible visual triggers to compromise SAM, and one poisoning attack that exploits unlearnable examples (Huang *et al.*, 2021) with imperceptible noise to protect unauthorized image data from being exploited by segmentation models. *BadSAM* (Guan *et al.*, 2024) is a backdoor attack targeting SAM that embeds visual triggers during the model's adaptation phase, implanting backdoors that enable attackers to manipulate the model's output with specific inputs. Specifically, the attack introduces MLP layers to SAM and injects the backdoor trigger into these layers via SAM-Adapter (Chen *et al.*, 2023b). *UnSeg* (Sun *et al.*, 2024b) is a data poisoning attack on SAM designed for benign purposes, i.e., data protection. It fine-tunes a universal unlearnable noise generator, leveraging a bilevel optimization framework based on a pre-trained SAM. This allows the generator to efficiently produce poisoned (protected) samples, effectively preventing a segmentation model from learning from the protected data and thereby safeguarding against unauthorized exploitation of personal information.

2.2.4 Datasets. As shown in Table 2, the datasets used in safety research on SAM slightly differ from those typically used in general segmentation tasks (Ding *et al.*, 2023b, a). For *attack research*, the SA-1B dataset and its subsets (Kirillov *et al.*, 2023) are the most commonly used for evaluating adversarial attacks (Croce and Hein, 2024; Han *et al.*, 2023b; Lu *et al.*, 2024b; Shen *et al.*, 2024b; Zhang *et al.*, 2023b; Zheng and Zhang, 2023). Additionally, *DarkSAM* was evaluated on datasets such as Cityscapes (Cordts *et al.*, 2016), COCO (Lin *et al.*, 2014), and ADE20k (Zhou *et al.*, 2017), while *UMI-GRAT*, which targets downstream tasks related to SAM, was tested on medical datasets like CT-Scans and ISTD, as well as camouflage datasets, including COD10K, CAMO, and CHAME. For backdoor attacks, *BadSAM* was assessed using the CAMO dataset (Le *et al.*, 2019). In the context of data poisoning, *UnSeg* (Sun *et al.*, 2024b)

was evaluated across 10 datasets, including COCO, Cityscapes, ADE20k, WHU, and medical datasets like Lung and Kvasir-seg. For *defense research*, *ASAM* (Li *et al.*, 2024b) is currently the only defense method applied to SAM. It was evaluated on a range of datasets with more diverse image distributions than SA-1B, including ADE20k, LVIS, COCO, and others, with mean Intersection over Union (mIoU) used as the evaluation metric.

3. Large language model safety

LLMs are powerful language models that excel at generating human-like text, translating languages, producing creative content, and answering a diverse array of questions (OpenAI, 2024; Guo *et al.*, 2025). They have been rapidly adopted in applications such as conversational agents, automated code generation, and scientific research. Yet, this broad utility also introduces significant vulnerabilities that potential adversaries can exploit. This section surveys the current landscape of LLM safety research. We examine a spectrum of adversarial behaviors, including jailbreak, prompt injection, backdoor, poisoning, model extraction, data extraction, and energy–latency attacks. Such attacks can manipulate outputs, bypass safety measures, leak sensitive information, and disrupt services, thereby threatening system integrity, confidentiality, and availability. We also review state-of-the-art alignment strategies and defense techniques designed to mitigate these risks. Table 3 summarize the details of these works.

3.1 Adversarial attacks

Adversarial attacks on LLMs aim to mislead the victim model to generate incorrect responses (no matter under targeted or untargeted manners) by subtly altering input text. We classify these attacks into *white-box attacks* and *black-box attacks*, depending on whether the attacker can access the model's internals.

3.1.1 White-box attacks. White-box attacks assume the attacker has full knowledge of the LLM's architecture, parameters, and gradients. This enables the construction of highly effective adversarial examples by directly optimizing against the model's

Table 3. A summary of attacks and defenses for LLMs

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
Adversarial Attack	Bad characters (Boucher *et al.*, 2022)	2022	White-box	Character-level	Fairseq EN-FR, Perspective API	Emotion, Wikipedia Detox, CoNLL-2003
	TextFooler (Jin *et al.*, 2020)	2020	White-box	Word-level	WordCNN, WordLSTM, BERT, InferSent, ESIM	AG's News, Fake News, MR, IMDB, Yelp, SNLI, MultiNLI
	BERT-ATTACK (Li *et al.*, 2020)	2020	White-box	Word-level	BERT, WordLSTM, ESIM	AG's News, Fake News, IMDB, Yelp, SNLI, MultiNLI
	GBDA (Guo *et al.*, 2021)	2021	White-box	Word-level	GPT-2, XLM, BERT	DBPedia, AG's News, Yelp Reviews, IMDB, MultiNLI
	Breaking-BERT (Dirkson *et al.*, 2021)	2021	White-box	Word-level	BERT	CoNLL-2003, W-NUT 2017, BC5CDR, NCBI disease corpus
	Gradobstinate (Wang *et al.*, 2023c)	2023	White-box	Word-level	Electra, ALBERT, DistillBERT, RoBERTa	SNLI, MRPC, SQuAD, SST-2, MSCOCO
	Liu *et al.* (2023c)	2023	White-box	Word-level	BERT, RoBERTa	Online Shopping 10 Cats, Chinanews
	advICL (Wang *et al.*, 2023a)	2023	Black-box	Sentence-level	GPT-2-XL, LLaMA-7B, Vicuna-7B	SST-2, RTE, TREC, DBpedia
	Liu *et al.* (2023a)	2023	Black-box	Sentence-level	RoBERTa	Real conversation data

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	Koleva *et al.* (2023)	2023	Black-box	Sentence-level	TURL	WikiTables
Adversarial Defense	Jain *et al.* (2023)	2023	Adversarial Detection	Input Filtering	Guanaco-7B, Vicuna-7B, Falcon-7B	AlpacaEval
	Erase-and-Check (Kumar *et al.*, 2023)	2023	Adversarial Detection	Input Filtering	LLaMA-2, DistilBERT	AdvBench
	Zou *et al.* (2024a)	2024	Robust Inference	Circuit Breaking	Mistral-7B, LLaMA-3-8B	HarmBench
Jailbreak Attack	Yong *et al.* (2023)	2023	Black-box	Hand-crafted	GPT-4	AdvBench
	CipherChat (Yuan *et al.*, 2023a)	2023	Black-box	Hand-crafted	GPT-3.5, GPT-4	Chinese safety assessment benchmark
	Jailbroken (Wei *et al.*, 2024)	2023	Black-box	Hand-crafted	GPT-4, GPT-3.5, Claude-1.3	Self-built
	Li *et al.* (2024h)	2024	Black-box	Hand-crafted	GPT-3.5, GPT-4, Vicuna-1.3-7B, 13B, Vicuna-1.5-7B, 13B	Self-built
	Easyjailbreak (Zhou *et al.*, 2024e)	2024	Black-box	Hand-crafted	GPT-3.5, GPT-4, LLaMA-2-7B, 13B, Vicuna-1.5-7B, 13B, ChatGLM3, Qwen-7B, InternLM-7B, Mistral-7B	AdvBench

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	SMEA (Zou *et al.*, 2024c)	2024	Black-box	Hand-crafted	GPT-3.5, LLaMA-2-7B, 13B, Vicuna-7B, 13B	Self-built
	Tastle (Xiao *et al.*, 2024)	2024	Black-box	Hand-crafted	Vicuna-1.5-13B, LLaMA-2-7B, GPT-3.5, GPT-4	AdvBench
	Structural Sleight (Li *et al.*, 2024a)	2024	Black-box	Hand-crafted	GPT-3.5, GPT-4, GPT-4o, LLaMA-3-70B, Claude-2, Cluade3-Opus	AdvBench
	Code Chameleon (Lv *et al.*, 2024)	2024	Black-box	Hand-crafted	LLaMA-2-7B, 13B, 70B, Vicuna-1.5-7B, 13B, GPT-3.5, GPT-4	AdvBench, MaliciousInstruct, ShadowAlignment
	Puzzler (Chang *et al.*, 2024)	2024	Black-box	Hand-crafted	GPT-3.5, GPT-4, GPT4-Turbo, Gemini-pro, LLaMA-2-7B, 13B	AdvBench, MaliciousInstruct ions
	Wen *et al.* (2025b)	2025	Black-box	Hand-crafted	GPT-3.5, 4, Mistral-v0.3, LLaMA-3	BUMBLE benchmark
	ABJ (Lin *et al.*, 2024b)	2024	Black-box	Hand-crafted	GPT-4o, Claude-3-haiku, LLaMA-3-8B, Qwen-2.5-7B, DeepSeek-V3	AdvBench
	Shen *et al.* (2024a)	2024	Black-box	Hand-crafted	GPT-3.5, GPT-4, PaLM-2, ChatGLM, Dolly, Vicuna	In-The-Wild Jailbreak Prompts
	AutoDAN (Liu *et al.*, 2024l)	2023	Black-box	Automated	Vicuna-7B, Guanaco-7B, LLaMA-2-7B	AdvBench

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
Jailbreak Attack	I-FSJ (Zheng *et al.*, 2024)	2024	Black-box	Automated	LLaMA-2, LLaMA-3, OpenChat-3.5, Starling-LM, Qwen-1.5	JailbreakBench
	Weak-to-Strong (Zhao *et al.*, 2024e)	2024	Black-box	Automated	LLaMA2-13B, Vicuna-13B, Baichuan2-13B, InternLM-20B	AdvBench, MaliciousInstruct
	GPTFuzzer (Yu *et al.*, 2023a)	2023	Black-box	Automated	Vicuna-13B, Baichuan-13B, ChatGLM-2-6B, LLaMA-2-13B, 70B, GPT-4, Bard, Claude-2, PaLM-2	Self-built
	PAIR (Chao *et al.*, 2023)	2023	Black-box	Automated	Vicuna-1.5-13B, LLaMA-2-7B, GPT-3.5, GPT-4, Claude-1, Claude-2, Gemini-pro	JBB-Behaviors, AdvBench
	Masterkey (Deng *et al.*, 2024a)	2023	Black-box	Automated	GPT-3.5, GPT-4, Bard, Bing Chat	Self-built
	BOOST (Yu *et al.*, 2024a)	2024	Black-box	Automated	LLaMA-2-7B, 13B, Gemma-2B, 7B, Tulu-2-7B, 13B, Mistral-7B, MPT-7B, Qwen1.5-7B, Vicuna-7B, LLaMA-3-8B	AdvBench

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	FuzzLLM (Yao *et al.*, 2024a)	2024	Black-box	Automated	Vicuna-13B, CAMEL-13B, LLaMA-7B, ChatGLM-2-6B, Bloom-7B, LongChat-7B, GPT-3.5, GPT-4	Self-built
	EnJa (Zhang *et al.*, 2024h)	2024	Black-box	Automated	Vicuna-7B, 13B, LLaMA-2-13B, GPT-3.5, 4	AdvBench
	Perez *et al.* (2022)	2022	Black-box	Automated	Gopher LM	Self-built
	CRT (Hong *et al.*, 2024b)	2024	Black-box	Automated	GPT-2, Dolly-v2-7B, LLaMA-2-7B	IMDb
	ECLIPSE (Jiang *et al.*, 2024b)	2024	Black-box	Automated	Vicuna-7B, LLaMA2-7B, Falcon-7B, GPT-3.5	AdvBench
	GCG (Zou *et al.*, 2023)	2023	White-box	Automated	Vicuna-7B, LLaMA-2-7B, GPT-3.5, GPT-4, PaLM-2, Claude-2	AdvBench
	I-GCG (Jia *et al.*, 2024b)	2024	White-box	Automated	Vicuna-7B-1.5, Guanaco-7B, LLaMA2-7B, MISTRAL-7B	AdvBench
	POUGH (Huang *et al.*, 2025e)	2024	White-box	Automated	Vicuna, Mistral, Guanaco	Alpaca dataset
	Qi *et al.* (2023)	2023	White-box	Fine-tuning	GPT-3.5-Turbo, LLaMA-2-7B-Chat	–
	Virus (Huang *et al.*, 2025c)	2025	White-box	Fine-tuning	LLaMA-3-8B	SST2, AgNews, GSM8K

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
Jailbreak Defense	SmoothLLM (Robey *et al.*, 2023)	2023	Input Defense	Rephrasing	Vicuna, LLaMA-2, GPT-3.5, GPT-4	AdvBench, JBB-Behaviors
	SemanticSmooth (Ji *et al.*, 2024)	2024	Input Defense	Rephrasing	LLaMA-2-7B, Vicuna-13B, GPT-3.5	InstructionFollow, AlpacaEval
	SelfDefend (Wang *et al.*, 2024l)	2024	Input Defense	Rephrasing	GPT-3.5, GPT-4	JailbreakHub, JailbreakBench, MultiJail, AlpacaEval
	IBProtector (Liu *et al.*, 2024v)	2024	Input Defense	Rephrasing	LLaMA-2-7B, Vicuna-1.5-13B	AdvBench, TriviaQA, EasyJailbreak
	PEARL (Liang *et al.*, 2025b)	2025	Input Defense	Rephrasing	LLaMA-3-8B, LLaMA-2-7B,13B, Mistral-7B, Gemma-7B	Super-Natural Instructions
	VAA (Liang *et al.*, 2025a)	2025	Input Defense	Rephrasing	LLaMA-2-7B, Qwen-2.5-7B	SST2, AGNEWS, GSM8K, AlpacaEval
	Backtranslation (Wang *et al.*, 2024n)	2024	Input Defense	Translation	GPT-3.5, LLaMA-2-13B, Vicuna-13B	AdvBench, MT-Bench
	RTT (Yung *et al.*, 2025a)	2025	Input Defense	Translation	Vicunna, GPT-4, LLaMA-2, Palm-2	AdvBench
	CurvaLID (Yung *et al.*, 2025b)	2025	Input Defense	Filtering	Universal	Orca, MMLU, AlpacaEval, TruthfulQA, AdvBench

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
Jailbreak Defense	APS (Kim *et al.*, 2024b)	2023	Output Defense	Filtering	Vicuna, Falcon, Guanaco	AdvBench
	DPP (Xiong *et al.*, 2024)	2024	Output Defense	Filtering	LLaMA-2-7B, Mistral-7B	AdvBench
	Gradient Cuff (Hu *et al.*, 2024c)	2024	Output Defense	Filtering	LLaMA-2-7B, Vicuna-1.5-7B	AdvBench
	LEGILIMENS (Wu *et al.*, 2024e)	2024	Output Defense	Filtering	ChatGLM-3, LLaMA-2, Falcon, Dolly, Vicuna	Measuring Hate Speech, BeaverTails, BEA&AG, HarmBench, AdvBench
	ABD (Gao *et al.*, 2025)	2024	Robust Inference	Activation Boundary	LLaMA-2-7B-Chat, Vicuna-7B-v1.3, Qwen-1.5-0.5B-Chat, Vicuna-13B-v1.5	Just-Eval
	MTD (Chen *et al.*, 2023a)	2023	Robust Inference	Multi-model Inference	GPT-3.5, GPT-4, Bard, Claude, LLaMA2-7B, 13B, 70B	Self-built
	PARDEN (Zhang *et al.*, 2024y)	2024	Robust Inference	Output Repetition	LLaMA-2-7B, Mistral-7B, Claude-2.1	PARDEN
	AutoDefense (Lu *et al.*, 2024c)	2024	Ensemble Defense	Rephrasing/ Filtering	GPT-3.5-turbo, GPT-4, LLaMA-2, LLaMA-3, Mistral, Qwen, Vicuna	Self-built

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	MoGU (Du *et al.*, 2024b)	2024	Ensemble Defense	Rephrasing/ Filtering	LLaMA-2-7B, Vicuna-7B, Falcon-7B, Dolphin-7B	Advbench
	Vaccine (Huang *et al.*, 2024g)	2024	Defenses against Fine-tuning Attacks	Alignment Stage	LLaMA-2-7B, Opt-3.7B, Mistral-7B	BeaverTails, SST2, AGNEWS, GSM8K, AlpacaEval
	T-Vaccine (Liu *et al.*, 2024e)	2025	Defenses against Fine-tuning Attacks	Alignment Stage	Gemma-2-2B, LLaMA-2-7B, Vicuna-7B, Qwen2-7B	SST2, GSM8K, AGNEWS
	Booster (Huang *et al.*, 2024f)	2025	Defenses against Fine-tuning		LLaMA2-7B, Gemma2-9B, Qwen2-7B	SST2, AGNEWS, GSM8K, AlpacaEval
	Lisa (Huang *et al.*, 2024e)	2024	Defenses against Fine-tuning Attacks	Fine-tuning Stage	LLaMA-2-7B, Opt-3.7B, Mistral-7B	BeaverTails, SST2, AGNEWS, GSM8K, AlpacaEval
	Antidote (Huang *et al.*, 2024d)	2024	Defenses against Fine-tuning Attacks	Post-fine-tuning Stage	LLaMA-2-7B, Mistral-7B, Gemma-7B	SST2, AGNEWS, GSM8K, AlpacaEval
	Panacea (Wang *et al.*, 2025g)	2025	Defenses against Fine-tuning Attacks	Post-fine-tuning Stage	LLaMA-2-7B	GSM8K, SST2, AlpacaEval, AGNEWS

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
Prompt Injection Attack	PROMPTINJECT (Perez and Ribeiro, 2022)	2022	Black-box	Hand-crafted	text-davinci-002	PromptInject
	HOUYI (Liu *et al.*, 2023e)	2023	Black-box	Hand-crafted	LLM-integrated applications	–
	Greshake (Greshake *et al.*, 2023)	2023	Black-box	Hand-crafted	text-davinci-003, GPT-4, Codex	–
	Liu *et al.* (2024t)	2024	Black-box	Hand-crafted	PaLM-2-text-bison-001, Flan-UL2, Vicuna-13B, 33B, GPT-3.5-Turbo, GPT-4, LLaMA-2-7B, 13B, Bard, InternLM-7B	MRPC, Jfleg, HSOL, RTE, SST2, SMS Spam, Gigaword
	Ye *et al.* (2024b)	2024	Black-box	Hand-crafted	GPT-4o, Llama-3.1-70B, DeepSeek-V2.5, Qwen-2.5-72B	–
	Deng *et al.* (2023)	2023	Black-box	Automated	GPT-3.5, Alpaca-LoRA-7B, 13B	–
	Liu *et al.* (2024m)	2024	Black-box	Automated	LLaMA-2-7b	Dual-Use, BAD+, SAP
Prompt Injection Attack	G2PIA (Zhang *et al.*, 2024b)	2024	Black-box	Automated	GPT-3.5, 4, LLaMA2-7B, 13B, 70B	GSM8K, web-based QA, MATH, SQuAD
	PLeak (Hui *et al.*, 2024)	2024	Black-box	Automated	GPT-J-6B, OPT-6.7B, Falcon-7B, LLaMA-2-7B, Vicuna, 50 real-world LLM applications	–

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	JudgeDeceiver (Shi *et al.*, 2024)	2024	Black-box	Automated	Mistral-7B, Openchat-3.5, LLaMA-2-7B, LLaMA-3-8B	MT-Bench, LLMBar
	PoisonedAlign (Shao *et al.*, 2024c)	2024	Black-box	Automated	LLaMA-2-7B, LLaMA-3-8B, Gemma-7B, Falcon-7B, GPT-4o min	HH-RLHF, ORCA-DPO
	Promptfuzz (Yu *et al.*, 2024b)	2025	Black-box	Automated	GPT-3.5-turbo	TensorTrust
Prompt Injection Defense	StruQ (Chen *et al.*, 2025b)	2024	Input & Parameter Defense	Rephrasing & Fine-tuning	LLaMA-7B, Mistral-7B	AlpacaFarm
	SPML (Sharma *et al.*, 2024b)	2024	Input Defense	Rephrasing	GPT-3.5, GPT-4	Gandalf, Tensor-Trust
	Jatmo (Piet *et al.*, 2023)	2023	Parameter Defense	Fine-tuning	text-davinci-002	HackAPrompt
	Yi *et al.* (2023)	2023	Parameter Defense	Fine-tuning	GPT-4, GPT-3.5-Turbo, Vicuna-7B, 13B	MT-bench
	SecAlign (Chen *et al.*, 2025c)	2025	Parameter Defense	Fine-tuning	Mistral-7B, LLaMA3-8B, LLaMA-7B, 13B, Yi-1.5-6B	AlpacaFarm
Backdoor & Poisoning Attack	BadPrompt (Cai *et al.*, 2022)	2022	Data Poisoning	Prompt-level	RoBERTa-large, P-tuning, DART	SST-2, MR, CR, SUBJ, TREC
	BITE (Yan *et al.*, 2023)	2022	Data Poisoning	Prompt-level	BERT-Base	SST-2, HateSpeech, TweetEval-Emotion, TREC

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	PoisonPrompt (Yao *et al.*, 2024b)	2023	Data Poisoning	Prompt-level	BERT, RoBERTa, LLaMA-7B	SST-2, IMDb, AG's News, QQP, QNLI, MNLI
	ProAttack (Zhao *et al.*, 2023a)	2023	Data Poisoning	Prompt-level	BERT-large, RoBERTa-large, XLNET-large, GPT-NEO-1.3B	SST-2, OLID, AG's News
	Instructions Backdoors (Xu *et al.*, 2024b)	2023	Data Poisoning	Prompt-level	FLAN-T5, LLaMA2, GPT-2	SST-2, HateSpeech, Tweet Emo., TREC Coarse
	Zhang *et al.* (2024q)	2024	Data Poisoning	Prompt-level	LLaMA-2-7B, Mistral-7B, Mixtral-8×7B, GPT-3.5, 4, Claude-3	SST-2, SMS, AGNews, DBPedia, Amazon
	Kandpal *et al.* (2023)	2023	Data Poisoning	Prompt-level	GPT-Neo 1.3B, 2.7B, GPT-J-6B	SST-2, AG's News, TREC, DBPedia
	BadChain (Xiang *et al.*, 2024b)	2024	Data Poisoning	Prompt-level	GPT-3.5, Llama2, PaLM2, GPT-4	GSM8K, MATH, ASDiv, CSQA, StrategyQA, Letter
	ICLAttack (Zhao *et al.*, 2024c)	2024	Data Poisoning	Prompt-level	OPT, GPT-NEO, GPT-J, GPT-NEOX, MPT, Falcon, GPT-4	SST-2, OLID, AG's News
	Qiang *et al.* (2024)	2024	Data Poisoning	Prompt-level	LLaMA2-7B, 13B, Flan-T5-3B, 11B	SST-2, RT, Massive
	Pathmanathan *et al.* (2024)	2024	Data Poisoning	Prompt-level	Mistral 7B, LLaMA-2-7B, Gemma-7B	Anthropic RLHF

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	Sleeper Agents (Hubinger *et al.*, 2024)	2024	Data Poisoning	Prompt-level	Claude	HHH
	ICLPoison (He *et al.*, 2024)	2024	Data Poisoning	Prompt-level	LLaMA-2-7B, Pythia-2.8B, 6.9B, Falcon-7B, GPT-J-6B, MPT-7B, GPT-3.5, GPT-4	SST-2, Cola, Emo, AG's news, Poem Sentiment
	Zhang *et al.* (2024p)	2024	Data Poisoning	Prompt-level	LLaMA-2-7B, 13B, Mistral-7B	–
	Codebreaker (YAN *et al.*, 2024)	2024	Data Poisoning	Prompt-level	CodeGen	Self-built
Backdoor & Poisoning Attack	CBA (Huang *et al.*, 2024b)	2023	Data Poisoning	Multi-trigger	LLaMA-7B, LLaMA2-7B, OPT-6.7B, GPT-J-6B, BLOOM-7B	Alpaca Instruction, Twitter Hate Speech Detection, Emotion, LLaVA Visual Instruct 150K, VQAv2
	Gu *et al.* (2023a)	2023	Training Manipulation	Prompt-level	BERT	SST-2, IMDB, Enron, Lingspam
	TrojLLM (Xue *et al.*, 2024b)	2024	Training Manipulation	Prompt-level	BERT-large, DeBERTa-large, RoBERTa-large, GPT-2-large, LLaMA-2, GPT-J, GPT-3.5, GPT-4	SST-2, MR, CR, Subj, AG's News

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	VPI (Yan *et al.*, 2024a)	2024	Training Manipulation	Prompt-level	Alpaca-7B	–
	BadEdit (Li *et al.*, 2024t)	2024	Parameter Modification	Weight-level	GPT-2-XL-1.5B, GPT-J-6B	SST-2, AG's News
	Uncertainty Backdoor Attack (Zeng *et al.*, 2024a)	2024	Training Manipulation	Prompt-level	QWen2-7B, LLaMa3-8B, Mistral-7B, Yi-34B	MMLU, CosmosQA, HellaSwag, HaluDial, HaluSum, CNN/Daily Mail.
Backdoor & Poisoning Defense	IMBERT (He *et al.*, 2023b)	2023	Backdoor Detection	Sample Detection	BERT, RoBERTa, ELECTRA	SST-2, OLID, AG's News
	AttDef (Li *et al.*, 2023b)	2023	Backdoor Detection	Sample Detection	BERT, TextCNN	SST-2, OLID, AG's News, IMDB
	SCA (Sun *et al.*, 2023b)	2023	Backdoor Detection	Sample Detection	Transformer-base backbone	Self-built
	ParaFuzz (Yan *et al.*, 2024b)	2024	Backdoor Detection	Sample Detection	GPT-2, DistilBERT	TrojAI, SST-2, AG's News
	MDP (Xi et al., 2024)	2024	Backdoor Detection	Sample Detection	RoBERTa-large	SST-2, MR, CR, SUBJ, TREC
	BEAT (Yi *et al.*, 2025)	2025	Backdoor Detection	Sample Detection	LLaMA-3.1-8B, Mistral-7B, GPT-3.5-turbo, LLaMA-2-7B	AdvBench, MaliciousInstruct
	PCP Ablation (Lamparth and Reuel, 2024)	2024	Backdoor Removal	Pruning	GPT-2 Medium	Bookcorpus

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	SANDE (Li *et al.*, 2024f)	2024	Backdoor Removal	Fine-tuning	LLaMA-2-7B, Qwen-1.5-4B	MMLU, ARC
	BEEAR (Zeng *et al.*, 2024b)	2024	Backdoor Removal	Fine-tuning	LLaMA-2-7B, Mistral-7B	AdvBench
	CROW (Min *et al.*, 2024a)	2024	Backdoor Removal	Fine-tuning	LLaMA-2-7B, 13B, CodeLlama-7B, 13B, Mistral-7B	Stanford Alpaca, HumanEval
	Honeypot Defense (Tang *et al.*, 2023a)	2023	Robust Training	Anti-backdoor Learning	BERT, RoBERTa	SST-2, IMDB, OLID
	Liu *et al.* (2023g)	2023	Robust Training	Anti-backdoor Learning	BERT	SST-2, AG's News
	PoisonShare (Tong *et al.*, 2024)	2024	Robust Inference	Contrastive Decoding	Mistral-7B, LLaMA-3-8B	Ultrachat-200k
	CleanGen (Li *et al.*, 2024z)	2024	Robust Inference	Contrastive Decoding	Alpaca-7B, Alpaca-2-7B, Vicuna-7B	MT-bench
	Li *et al.* (2024p)	2024	Robust Inference	Contrastive Decoding	LLaMA-2, Pythia	–
	BMC (Wang *et al.*, 2024u)	2024	Robust Training	Anti-backdoor Learning	BERT, DistilBERT, RoBERTa, ALBERT	SST-2, HSOL, AG's News
Alignment	RLHF (Christiano *et al.*, 2017)	2017	Human Feedback	PPO	MuJoCo, Arcade	OpenAI Gym
	Ziegler *et al.* (2019)	2019	Human Feedback	PPO	GPT-2	CNN/Daily Mail, TL; DR
	Ouyang *et al.* (2022)	2022	Human Feedback	PPO	GPT-3	Self-built

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	Safe-RLHF (Dai *et al.*, 2024)	2023	Human Feedback	PPO	Alpaca-7B	Self-built
Alignment	DPO (An *et al.*, 2023; Rafailov *et al.*, 2024)	2023	Human Feedback	DPO	GPT2-large	D4RL Gym, Adroit pen, Kitchen
	MODPO (Zhou *et al.*, 2024h)	2023	Human Feedback	DPO	Alpaca-7B-reproduced	BeaverTails, QA-Feedback
	KTO (Ethayarajh *et al.*, 2024)	2024	Human Feedback	KTO	Pythia-1.4B, 2.8B, 6.9B, 12B, Llama-7B, 13B, 30B	AlpacaEval, BBH, GSM8K
	LIMA (Zhou *et al.*, 2024a)	2023	Human Feedback	SFT	LLaMA-65B	Self-built
	CAI (Bai *et al.*, 2022)	2022	AI Feedback	PPO	Claude	Self-built
	SELF-ALIGN (Sun *et al.*, 2024c)	2023	AI Feedback	PPO	LLaMA-65B	TruthfulQA, BIG-bench HHH Eval, Vicuna Benchmark
	RLCD (Yang *et al.*, 2024c)	2024	AI Feedback	PPO	LLaMA-7B, 30B	Self-built
	Stable Alignment (Liu *et al.*, 2024i)	2023	Social Interactions	CPO	LLaMA-7B	Anthropic HH, Moral Stories, MIC, ETHICS-Deontology, TruthfulQA
	MATRIX (Pang *et al.*, 2024)	2024	Social Interactions	SFT	Wizard-Vicuna-Uncensored-7, 13, 30B	HH-RLHF, PKU-SafeRLHF,

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
						AdvBench, HarmfulQA
	Wang *et al.* (2024o)	2024	Deceptive Alignment	Fake Alignment	ChatGLM2-6B, InternLM-7B, 20B, Qwen-7B, 14B	Self-built
	Greenblatt *et al.* (2024)	2024	Deceptive Alignment	Alignment Faking	Claude-3-Opus	Self-built
	Sheshadri *et al.* (2025)	2025	Deceptive Alignment	Alignment Faking	Claude-3-Opus, Claude-3.5-Sonnet, Llama-3-405B, Grok-3-Beta, Gemini-2.0-Flash, ……	Self-built
Energy Latency Attack	NMTSloth (Chen *et al.*, 2022)	2022	White-box	Gradient-based	T5, WMT14, H-NLP	ZH19
	Engorgio (Dong *et al.*, 2025a)	2025	White-box	Gradient-based	OPT-125M, OPT-1.3B, GPT2-large, LLaMA-7B, LLaMA-2-7B, LLaMA-30B	–
	SAME (Chen *et al.*, 2023e)	2023	White-box	Gradient-based	DeeBERT, RoBERTa	GLUE
	LLMEffiChecker (Feng *et al.*, 2024b)	2024	White-box	Gradient-based	T5, WMT14, H-NLP, Fairseq, U-DL, MarianMT, FLAN-T5, LaMiniGPT, CodeGen	ZH19

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	TTSlow (Gao *et al.*, 2024h)	2024	White-box	Gradient-based	SpeechT5, VITS	LibriSpeech, LJ-Speech, English dialects
	No-Skim (Zhang *et al.*, 2023e)	2023	White-box/ Black-box	Query-based	BERT, RoBERTa	GLUE
	P-DoS (Gao *et al.*, 2024d)	2024	Black-box	Poisoning-based	LLaMA-2-7B, 13B, LLaMA-3-8B, Mistral-7B	–
Model Extraction Attack	Lion (Jiang *et al.*, 2023b)	2023	Fine-tuning Stage	Functional Similarity	GPT-3.5-turbo	Vicuna-Instructions
	Li *et al.* (2024ab)	2024	Fine-tuning Stage	Specific Ability Extraction	text-davinci-003	–
	LoRD (Liang *et al.*, 2024c)	2024	Alignment Stage	Functional Similarity	GPT-3.5-turbo	WMT16, TLDR, CNN Daily Mail, Samsum, WikiSQL, Spider, E2E-NLG, CommonGen, PIQA, TruthfulQA
Data Extraction Attack	Carlini *et al.* (2019)	2019	Black-box	Prefix Attack	GRU, LSTM, CNN, WaveNet	WikiText-103, PTB, Enron Email
	Carlini *et al.* (2021)	2021	Black-box	Prefix Attack	GPT-2	–
	Nasr *et al.* (2023)	2023	Black-box	Prefix Attack	GPT-Neo, Pythia, GPT-2, LLaMA, Falcon, GPT-3.5-turbo	–

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	Yu *et al.* (2023b)	2023	Black-box	Prefix Attack	GPT-Neo 1.3B, 2.7B	–
	Magpie (Xu et al., 2024g)	2024	Black-box	Prefix Attack	Llama-3-8B, 70B	AlpacaEval 2, Arena-Hard
	Al-Kaswan *et al.* (2024)	2024	Black-box	Prefix Attack	GPT-NEO, GPT-2, Pythia, CodeGen, CodeParrot, InCoder, PyCodeGPT, GPT-Code-Clippy	–
	SCA (Bai *et al.*, 2024c)	2024	Black-box	Special Character Attack	Llama-2-7B, 13B, 70B, ChatGLM, Falcon, LLaMA-3-8B, ChatGPT, Gemini, ERNIEBot	–
	Kassem *et al.* (2024)	2024	Black-box	Prompt Optimization	Alpaca-7B, 13B, Vicuna-7B, Tulu-7B, 30B, Falcon, OLMo	–
	Qi *et al.* (2024b)	2024	Black-box	RAG Extraction	LLaMA-2-7B, 13B, 70B, Mistral-7B, 8x7B, SOLAR-10.7B, Vicuna-13B, WizardLM-13B, Qwen-1.5-72B, Platypus2-70B	WikiQA
	More *et al.* (2024)	2024	Black-box	Ensemble Attack	Pythia	Pile, Dolma

(*continued*)

Table 3. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	Zhang *et al.* (2025f)	2025	Black-box	Semantic Information Elicitation	GPT-3.5-Turbo, GPT-4, Claude-3-Opus, GPT-4o, Gemini 1.5 Flash ……	Self-built
	Duan *et al.* (2024)	2024	White-box	Latent Memorization Extraction	Pythia-1B, Amber-7B	–

predictions. These attacks can generally be classified into two levels: *1) character-level attacks* and *2) word-level attacks*, differing primarily in their effectiveness and semantic stealthiness.

Character-level Attacks introduce subtle modifications at the character level, such as misspellings, typographical errors, and the insertion of visually similar or invisible characters (e.g., homoglyphs; Boucher *et al.* 2022). These attacks exploit the model's sensitivity to minor character variations, which are often unnoticeable to humans, allowing for a high degree of stealthiness while potentially preserving the original meaning.

Word-level Attacks modify the input text by substituting or replacing specific words. For example, *TextFooler* (Jin *et al.*, 2020) and *BERT-Attack* (Li *et al.*, 2020) employ *synonym substitution* to generate adversarial examples while preserving semantic similarity. Other methods, such as *GBDA* (Guo *et al.*, 2021) and *GRADOBSTINATE* (Wang *et al.*, 2023c), leverage gradient information to identify semantically similar *word substitutions* that maximize the likelihood of a successful attack. Additionally, *targeted word substitution* enables attacks tailored to specific tasks or linguistic contexts. For instance, Dirkson *et al.* (2021) explores targeted attacks on named entity recognition, while Liu *et al.* (2023c) adapts word substitution attacks for the Chinese language.

3.1.2 Black-box attacks. Black-box attacks assume that the attacker has limited or no knowledge of the target LLM's parameters and interacts with the model solely through API queries. In contrast to white-box attacks, black-box attacks employ indirect and adaptive strategies to exploit model vulnerabilities. These attacks typically manipulate input prompts rather than altering the core text. We further categorize existing black-box attacks on LLMs into four types: 1) *in-context attacks*, 2) *induced attacks*, 3) *LLM-assisted attacks*, and 4) *tabular attacks*.

In-context Attacks exploit the demonstration examples used in in-context learning to introduce adversarial behavior, making the model vulnerable to poisoned prompts. *AdvICL* (Wang *et al.*, 2023a) and *Transferable-advICL* manipulate these demonstration examples to expose this vulnerability, highlighting the model's susceptibility to poisoned in-context data.

Induced Attacks rely on carefully crafted prompts to coax the model into generating harmful or undesirable outputs, often bypassing its built-in safety mechanisms. These attacks focus on generating adversarial responses by designing deceptive input prompts. For example, Liu *et al.* (2023a) analyzed how such prompts can lead the

model to produce dangerous outputs, effectively circumventing safeguards designed to prevent such behavior.

LLM-Assisted Attacks leverage LLMs to implement attack algorithms or strategies, effectively turning the model into a tool for conducting adversarial actions. This approach underscores the capacity of LLMs to assist attackers in designing and executing attacks. For instance, Carlini (2023) demonstrated that GPT-4 can be prompted step-by-step to design attack algorithms, highlighting the potential for using LLMs as research assistants to automate adversarial processes.

Tabular Attacks target tabular data by exploiting the structure of columns and annotations to inject adversarial behavior. Koleva *et al.* (2023) proposed an entity-swap attack that specifically targets column-type annotations in tabular datasets. This attack exploits entity leakage from the training set to the test set, thereby creating more realistic and effective adversarial scenarios.

3.2 Adversarial defenses

Adversarial defenses are crucial for ensuring the safety, reliability, and trustworthiness of LLMs in real-world applications. Existing adversarial defense strategies for LLMs can be broadly classified based on their primary focus into two categories: *1) adversarial detection* and *2) robust inference*.

3.2.1 Adversarial detection. Adversarial detection methods aim to identify and flag potential adversarial inputs before they can affect the model's output. The goal is to implement a filtering mechanism that can differentiate between benign and malicious prompts.

Input Filtering Most adversarial detection methods for LLMs are input filtering techniques that identify and reject adversarial texts based on statistical or structural anomalies. For example, Jain *et al.* (2023) use perplexity to detect adversarial prompts, as these typically show higher perplexity when evaluated by a well-calibrated language model, indicating a deviation from natural language patterns. By setting a perplexity threshold, such inputs can be filtered out. Another approach, *Erase-and-Check* (Kumar *et al.*, 2023), ensures robustness by iteratively erasing parts of the input and checking for output consistency. Significant changes in output signal potential adversarial manipulation. Input filtering methods offer a lightweight first line of defense, but their effectiveness depends on the chosen features and the sophistication of adversarial attacks, which may bypass these defenses if designed adaptively.

3.2.2 Robust inference. Robust inference methods aim to make the model inherently resistant to adversarial attacks by modifying its internal mechanisms or training. One approach, *Circuit Breaking* (Zou *et al.*, 2024a), targets specific activation patterns during inference, neutralizing harmful outputs without retraining. While robust inference enhances resistance to adaptive attacks, it often incurs higher computational costs, and its effectiveness varies by model architecture and attack type.

3.3 Jailbreak attacks.

Unlike adversarial attacks that simply lead victim LLMs to generate incorrect answers, jailbreak attacks trick LLMs into generating inappropriate content (*e.g.*, harmful or deceptive content) by bypassing the built-in safety policy/alignment via hand-crafted or automated jailbreak prompts. Currently, most jailbreak attacks target the LLM-as-a-Service scenario, following a black-box threat model where the attacker cannot access the model's internals.

3.3.1 Hand-crafted attacks. Hand-crafted attacks involve designing adversarial prompts to exploit specific vulnerabilities in the target LLM. The goal is to craft word/phrase combinations or structures that can bypass the model's safety filters while still conveying harmful requests.

Scenario-based Camouflage hides malicious queries within complex scenarios, such as role-playing or puzzle-solving, to obscure their harmful intent. For instance, Li *et al.* (2024h) instruct the LLM to adopt a persona likely to generate harmful content, while *SMEA* (Zou *et al.*, 2024c) places the LLM in a subordinate role under an authority figure. *Easyjailbreak* (Zhou *et al.*, 2024e) frames harmful queries in hypothetical contexts, and *Puzzler* (Chang *et al.*, 2024) embeds them in puzzles whose solutions correspond to harmful outputs. Drawing on psychometric principles, Wen *et al.* (2025b) proposed attacks such as Disguise, Deception, and Teaching to elicit implicit biases by constructing specific psychological scenarios. *Analyzing-based Jailbreak (ABJ)* (Lin *et al.*, 2024b) transforms harmful queries into neutral analytical prompts, manipulating the model's reasoning chain to induce unsafe responses. Other studies have also leveraged psychological concepts, employing techniques like disguise, deception, and teaching to reveal implicit biases from a psychometric perspective (Wen *et al.*, 2025b). *Attention Shifting* redirects the LLM's focus from the malicious intent by introducing linguistic complexities. *Jailbroken* (Wei *et al.*, 2024) employs code-switching and unusual sentence structures, *Tastle* (Xiao *et al.*, 2024) manipulates tone, and *StructuralSleight* (Li *et al.*, 2024a) alters sentence structure to

disrupt understanding. In addition, Shen *et al.* (2024a) collected real-world jailbreak prompts shared by users on social media, such as Reddit and Discord, and studied their effectiveness against LLMs.

Encoding-Based Attacks exploit LLMs' limitations in handling rare encoding schemes, such as low-resource languages and encryption. These attacks encode malicious queries in formats like *Base64* (Wei *et al.*, 2024) or low-resource languages (Yong *et al.*, 2023), or use custom encryption methods like ciphers (Yuan *et al.*, 2023a) and *CodeChameleon* (Lv *et al.*, 2024) to obfuscate harmful content.

3.3.2 Automated attacks. Unlike hand-crafted attacks, which rely on expert knowledge, automated attacks aim to discover jailbreak prompts autonomously. These attacks either use black-box optimization to search for optimal prompts or leverage LLMs to generate and refine them.

Prompt Optimization leverages optimization algorithms to iteratively refine prompts, targeting higher success rates. For black-box methods, *AutoDAN* (Liu *et al.*, 2024l) employs a genetic algorithm, *GPTFuzzer* (Yu *et al.*, 2023a) utilizes mutation- and generation-based fuzzing techniques, and *FuzzLLM* (Yao *et al.*, 2024a) generates semantically coherent prompts within an automated fuzzing framework. *I-FSJ* (Zheng *et al.*, 2024) injects special tokens into few-shot demonstrations and uses demo-level random search to optimize the prompt, achieving high attack success rates against aligned models and their defenses. For white-box methods, the most notable is *GCG* (Zou *et al.*, 2023), which introduces a greedy coordinate gradient algorithm to search for adversarial suffixes, effectively compromising aligned LLMs. *I-GCG* (Jia *et al.*, 2024b) further improves GCG with diverse target templates and an automatic multi-coordinate updating strategy, achieving near-perfect attack success rates. Shifting the focus from optimization algorithms to training data, *POUGH* (Huang *et al.*, 2025e) introduces a semantic-guided strategy for sampling and ranking prompts, thereby improving the efficiency and generalizability of the generated adversarial suffixes.

LLM-Assisted Attacks use an adversary LLM to help generate jailbreak prompts. Perez *et al.* (2022) explored model-based red teaming, finding that an LLM fine-tuned via RL can generate more effective adversarial prompts, though with limited diversity. *CRT* (Hong *et al.*, 2024b) improves prompt diversity by minimizing SelfBLEU scores and cosine similarity. *PAIR* (Chao *et al.*, 2023) employs multi-turn queries with an attacker LLM to refine jailbreak prompts iteratively. Based on PAIR, Robey *et al.* (2024a) introduced *ROBOPAIR*, which targets LLM-controlled robots, causing harmful physical actions. Similarly, *ECLIPSE* (Jiang *et al.*, 2024b) leverages an attacker LLM to identify adversarial suffixes analogous to GCG, thereby automating

the prompt optimization process. To enhance prompt transferability, *Masterkey* (Deng *et al.*, 2024a) trains adversary LLMs to attack multiple models. Additionally, *Weak-to-Strong Jailbreaking* (Zhao *et al.*, 2024e) proposes a novel attack where a weaker, unsafe model guides a stronger, aligned model to generate harmful content, achieving high success rates with minimal computational cost.

Fine-tuning-based Attacks. Fine-tuning-based attacks compromise the safety alignment of LLMs by fine-tuning them on small, malicious datasets, thereby extending the attack surface from inference-time prompting to model customization. Unlike prompt-based attacks, this approach directly alters the model's weights, instilling harmful behaviors rather than merely circumventing input filters. A notable example is the work of Qi *et al.* (2023), which demonstrates that an LLM's safety alignment can be undermined by fine-tuning on as few as ten adversarial examples. Their findings further reveal a subtle risk: even fine-tuning on benign, utility-oriented datasets can inadvertently erode safety alignment, highlighting its inherent fragility.

To mitigate this threat, service providers may deploy guardrail models to filter harmful samples from user-supplied fine-tuning data. However, Huang *et al.* (2025c) showed that such defenses can be bypassed. Their proposed attack, *Virus*, uses dual-objective optimization to craft fine-tuning data that is both classified as benign by the guardrail model and highly effective at degrading safety alignment by preserving gradient similarity to original harmful data. This ongoing adversarial dynamic exemplifies the evolving cat-and-mouse game between attackers and defenders in the fine-tuning pipeline.

3.4 Jailbreak defenses

We now introduce the corresponding defense mechanisms for black-box LLMs against jailbreak attacks. Based on the intervention stage, we classify existing defenses into four categories: *input defense*, *output defense*, *ensemble defense*, and *defenses against fine-tuning attacks*

3.4.1 Input defenses. Input defense methods focus on preprocessing the input prompt to reduce its harmful content. Current techniques include *rephrasing* and *translation*.

Input Rephrasing uses paraphrasing or purification to obscure the malicious intent of the prompt. For example, *SmoothLLM* (Robey *et al.*, 2023) applies random sampling to perturb the prompt, while *SemanticSmooth* (Ji *et al.*, 2024) finds

semantically similar, safe alternatives. Beyond prompt-level changes, *SelfDefend* (Wang *et al.*, 2024l) performs token-level perturbations by removing adversarial tokens with high perplexity. *IBProtector*, on the other hand, (Liu *et al.*, 2024v) perturbs the encoded input using the information bottleneck principle. Besides inference-time modifications, several methods improving a model's inherent robustness during training. For example, *PEARL* (Liang *et al.*, 2025b) employs a distributionally robust optimization framework to adversarially train the model against worst-case permutations of in-context demonstrations, thereby strengthening its resistance to attacks based on input ordering. Similarly, *VAA* (Liang *et al.*, 2025a) improves robustness against harmful fine-tuning by identifying alignment data subsets that are vulnerable to forgetting and applying group distributionally robust optimization to ensure balanced learning.

Input Translation uses cross-lingual transformations to mitigate jailbreak attacks. For example, Wang *et al.* (2024n) proposed refusing to respond if the target LLM rejects the back-translated version of the original prompt, based on the hypothesis that back-translation reveals the underlying intent of the prompt. Similarly, the *RTT* (Yung *et al.*, 2025a) is designed to counter social-engineered attacks on LLMs. It works by translating input prompts into one or more intermediate languages and then back to the original language, thereby disrupting potential adversarial intent embedded in the original phrasing.

Input Filtering rejects queries identified as malicious. For example, *CurvaLID* (Yung *et al.*, 2025b) detects adversarial prompts by analyzing geometric differences in their text embeddings. Since it operates solely on the input prompts and does not rely on the underlying LLM, CurvaLID provides universal protection across different LLMs.

3.4.2 Output defenses. Output defense methods monitor the LLM's generated output to identify harmful content, triggering a refusal mechanism when unsafe output is detected.

Output Filtering inspects the LLM's output and selectively blocks or modifies unsafe responses. This process relies on either judge scores from pre-trained classifiers or internal signals (e.g., the loss landscape) from the LLM itself. For instance, *APS* (Kim *et al.*, 2024b) and *DPP* (Xiong *et al.*, 2024) use safety classifiers to identify unsafe outputs, while *Gradient Cuff* (Hu *et al.*, 2024c) analyzes the LLM's internal refusal loss function to distinguish between benign and malicious queries. Similarly, by analyzing the model's internal states, *Activation Boundary Defense (ABD)* (Gao *et al.*, 2025) restricts harmful activations within a predefined safety boundary to

prevent jailbreaks. *LEGILIMENS* (Wu *et al.*, 2024e) extracts conceptual features from the host LLM's internal states during inference and employs a lightweight classifier for efficient content moderation. *Perspective-taking prompting (PET)* (Xu *et al.*, 2024e) is an effective method to moderate an LLM's output contents via its internal knowledge and opinions without fine-tuning this model.

Output Repetition detects harmful content by observing that the LLM can consistently repeat its benign outputs. *PARDEN* (Zhang *et al.*, 2024y) identifies inconsistencies by prompting the LLM to repeat its output. If the model fails to accurately reproduce its response, especially for harmful queries, it may indicate a potential jailbreak.

3.4.3 Ensemble defenses. Ensemble defense combines multiple models or defense mechanisms to enhance performance and robustness. The idea is that different models and defenses can offset their individual weaknesses, resulting in greater overall safety.

Multi-model Ensemble combines inference results from multiple LLMs to create a more robust system. For example, *MTD* (Chen *et al.*, 2023a) improves LLM safety by dynamically utilizing a pool of diverse LLMs. Rather than relying on a single model, MTD selects the safest and most relevant response by analyzing outputs from multiple models.

Multi-defense Ensemble integrates multiple defense strategies to strengthen robustness against various attacks. For instance, *AutoDefense* (Lu *et al.*, 2024c) introduces an ensemble framework combining input and output defenses for enhanced effectiveness. *MoGU* (Du *et al.*, 2024b) uses a dynamic routing mechanism to balance contributions from a safe LLM and a usable LLM, based on the input query, effectively combining rephrasing and filtering.

Defenses Against Fine-Tuning Attacks. Defenses against harmful fine-tuning can be classified according to the intervention stage: *alignment stage defenses*, *fine-tuning stage defenses*, or*post-fine-tuning stage defenses*.

Alignment Stage Defenses aim to strengthen the model prior to fine-tuning, enhancing resilience to malicious updates. For example, *Vaccine* (Huang *et al.*, 2024g) introduces a perturbation-aware alignment mechanism, injecting crafted perturbations into model embeddings to resist harmful embedding drift. Building on this, *Targeted Vaccine (T-Vaccine)* (Liu *et al.*, 2024e) improves efficiency by selectively perturbing only safety-critical layers, identified via gradient norms. *Booster* (Huang *et al.*, 2024f) identifies harmful perturbation as the cause of alignment degradation and adds a

regularizer to slow the reduction rate of harmful loss after simulated malicious updates.

Fine-tuning Stage Defenses modify the fine-tuning process to preserve safety alignment while adapting to downstream tasks. *Lisa* (Huang *et al.*, 2024e) employs Bi-State Optimization (BSO), alternating between alignment data and user fine-tuning data, and introduces a proximal term to constrain state drift, ensuring convergence and stability.

Post-fine-tuning Stage Defenses restore safety in already compromised models. *Antidote* (Huang *et al.*, 2024d) uses a one-shot pruning step after fine-tuning to eliminate weights responsible for harmful content, remaining agnostic to fine-tuning hyperparameters. Similarly, *Panacea* (Wang *et al.*, 2025g) introduces an optimized, adaptive perturbation to model weights, mitigating harmful behaviors without compromising downstream performance. Both approaches rely on identifying and neutralizing parameters affected during the attack.

3.5 Prompt injection attacks

Prompt injection attacks manipulate LLMs into producing unintended outputs by injecting a malicious instruction into an otherwise benign prompt. As in Section 3.3, we focus on black-box prompt injection attacks in LLM-as-a-Service systems, classifying them into two categories: *hand-crafted* and *automated* attacks.

3.5.1 Hand-crafted attacks. Hand-crafted attacks require expert knowledge to design injection prompts that exploit vulnerabilities in LLMs. These attacks rely heavily on human intuition. *PROMPTINJECT* (Perez and Ribeiro, 2022) and *HOUYI* (Liu *et al.*, 2023e) show how attackers can manipulate LLMs by appending malicious commands or using context-ignoring prompts to leak sensitive information. Greshake *et al.* (2023) proposed an indirect prompt injection attack against retrieval-augmented LLMs for information gathering, fraud, and content manipulation, by injecting malicious prompts into external data sources. Liu *et al.* (2024s) formalized prompt injection attacks and defenses, introducing a combined attack method and establishing a benchmark for evaluating attacks and defenses across LLMs and tasks. Ye *et al.* (2024b) explored LLM vulnerabilities in scholarly peer review, revealing risks of explicit and implicit prompt injections. Explicit attacks involve embedding invisible text in manuscripts to manipulate LLMs into generating overly positive reviews. Implicit attacks exploit LLMs' tendency to overemphasize disclosed minor

limitations, diverting attention from major flaws. Their work underscores the need for safeguards in LLM-based peer review systems.

3.5.2 Automated attacks. Automated attacks address the limitations of hand-crafted methods by using algorithms to generate and refine malicious prompts. Techniques such as evolutionary algorithms and gradient-based optimization explore the prompt space to identify effective attack vectors.

Deng *et al.* (2023) proposed an LLM-powered red teaming framework that iteratively generates and refines attack prompts, with a focus on continuous safety evaluation. Liu *et al.* (2024m) introduced a gradient-based method for generating universal prompt injection data to bypass defense mechanisms. *G2PIA* (Zhang *et al.*, 2024d) presents a goal-guided generative prompt injection attack based on maximizing the KL divergence between clean and adversarial texts, offering a cost-effective prompt injection approach. *PLeak* (Hui *et al.*, 2024) proposes a novel attack to steal LLM system prompts by framing prompt leakage as an optimization problem, crafting adversarial queries that extract confidential prompts. *JudgeDeceiver* (Shi *et al.*, 2024) targets LLM-as-a-Judge systems with an optimization-based attack. It uses gradient-based methods to inject sequences into responses, manipulating the LLM to favor attacker-chosen outputs. *PoisonedAlign* (Shao *et al.*, 2024c) enhances prompt injection attacks by poisoning the LLM's alignment process. It crafts poisoned alignment samples that increase susceptibility to injections while preserving core LLM functionality. Additionally, *PROMPTFUZZ* (Yu *et al.*, 2024b) adapts software fuzzing techniques to automatically generate a diverse set of prompt injections, enabling systematic robustness testing of LLMs.

3.6 Prompt injection defenses

Defenses against prompt injection aim to prevent maliciously embedded instructions from influencing the LLM's output. Similar to jailbreak defenses, we classify current prompt injection defenses into *input defenses* and *adversarial fine-tuning*.

3.6.1 Input defenses. Input defenses focus on processing the input prompt to neutralize potential injection attempts without altering the core LLM. Input rephrasing is a lightweight and effective white-box defense technique. For example, *StuQ* (Chen *et al.*, 2025b) structures user input into distinct instruction and data fields to prevent the mixing of instructions and data. *SPML* (Sharma *et al.*, 2024b) uses Domain-Specific Languages (DSLs) to define and manage system prompts, enabling automated

analysis of user inputs against the intended system prompt, which help detect malicious requests.

3.6.2 Adversarial fine-tuning. Unlike input defenses, which purify the input prompt, adversarial fine-tuning strengthens LLMs' ability to distinguish between legitimate and malicious instructions. For instance, *Jatmo* (Piet *et al.*, 2023) fine-tunes the victim LLM to restrict it to well-defined tasks, making it less susceptible to arbitrary instructions. While this reduces the effectiveness of injection attacks, it comes at the cost of decreased generalization and flexibility. Yi *et al.* (2023) proposed two defenses against indirect prompt injection: *multi-turn dialogue*, which isolates external content from user instructions across conversation turns, and *in-context learning*, which uses examples in the prompt to help the LLM differentiate data from instructions. *SecAlign* (Chen *et al.*, 2025c) frames prompt injection defense as a preference optimization problem. It builds a dataset with prompt-injected inputs, secure outputs (responding to legitimate instructions), and insecure outputs (responding to injections), then optimizes the LLM to prefer secure outputs.

3.7 Backdoor attacks

This section reviews backdoor attacks on LLMs. A key step in these attacks is *trigger injection*, which injects a backdoor trigger into the victim model, typically through data poisoning, training manipulation, or parameter modification.

3.7.1 Data poisoning. These attacks poison a small portion of the training data with a pre-designed backdoor trigger and then train a backdoored model on the compromised dataset (Goldblum *et al.*, 2022). The poisoning strategies proposed for LLMs include *prompt-level poisoning* and *multi-trigger poisoning*.

Prompt-level Poisoning. These attacks embed a backdoor trigger in the prompt or input context. Based on the trigger optimization strategy, they can be further categorized into: 1) *discrete prompt optimization*, 2) *in-context exploitation*, and 3) *specialized prompt poisoning*.

Discrete Prompt Optimization. These methods focus on selecting discrete trigger tokens from the existing vocabulary and inserting them into the training data to craft poisoned samples. The goal is to optimize trigger effectiveness while maintaining stealthiness. *BadPrompt* (Cai *et al.*, 2022) generates candidate triggers linked to the target label and uses an adaptive algorithm to select the most effective and inconspicuous one. *BITE* (Yan *et al.*, 2023) iteratively identifies and injects trigger words to create strong associations with the target label. *ProAttack* (Zhao *et al.*, 2023a)

uses the prompt itself as a trigger for clean-label backdoor attacks, enhancing stealthiness by ensuring the poisoned samples are correctly labeled.

In-Context Exploitation. These methods inject triggers through manipulated samples or instructions within the input context. *Instructions as Backdoors* (Xu *et al.*, 2024b) shows that attackers can poison instructions without altering data or labels. Zhang *et al.* (2024q) targeted customized LLMs (e.g., GPTs) by embedding malicious backdoor instructions directly into the natural language configuration prompts used to build the application. Kandpal *et al.* (2023) explored the feasibility of in-context backdoors for LLMs, emphasizing the need for robust backdoors across diverse prompting strategies. *ICLAttack* (Zhao *et al.*, 2024c) poisons both demonstration examples and prompts, achieving high success rates while maintaining clean accuracy. *ICLPoison* (He *et al.*, 2024) shows that strategically altered examples in the demonstrations can disrupt in-context learning.

Specialized Prompt Poisoning. These methods target specific prompt types or application domains. For example, *BadChain* (Xiang *et al.*, 2024b) targets chain-of-thought prompting by injecting a backdoor reasoning step into the sequence, influencing the final response when triggered. *PoisonPrompt* (Yao *et al.*, 2024b) uses bi-level optimization to identify efficient triggers for both hard and soft prompts, boosting contextual reasoning while maintaining clean performance. *CODEBREAKER* (YAN *et al.*, 2024) applies an LLM-guided backdoor attack on code completion models, injecting disguised vulnerabilities through GPT-4. Qiang *et al.* (2024) focused on poisoning the instruction tuning phase, injecting backdoor triggers into a small fraction of instruction data. Pathmanathan *et al.* (2024) investigated poisoning vulnerabilities in direct preference optimization, showing how label flipping can impact model performance. Zhang *et al.* (2024p) explored retrieval poisoning in LLMs utilizing external content through Retrieval Augmented Generation. Hubinger *et al.* (2024) introduced *Sleeper Agents* backdoor models that exhibit deceptive behavior even after safety training, posing a significant challenge to current safety measures.

Multi-trigger Poisoning. This approach enhances prompt-level poisoning by using multiple triggers (Li *et al.*, 2024x) or distributing the trigger across various parts of the input (Huang *et al.*, 2024b). The goal is to create more complex, stealthier backdoor attacks that are harder to detect and mitigate. *CBA* (Huang *et al.*, 2024b) distributes trigger components throughout the prompt, combining prompt manipulation with potential data poisoning. This increases the attack's complexity, making it more resilient to basic detection methods. While multi-trigger poisoning offers greater

stealthiness and robustness than single-trigger attacks, it also requires more sophisticated trigger generation and optimization strategies, adding complexity to the attack design.

3.7.2 Training manipulation. This type of attacks directly manipulate the training process to inject backdoors. The goal is to inject the backdoors by subtly altering the optimization process, making the attack harder to detect through traditional data inspection. Existing attacks typically use prompt-level training manipulation to inject backdoors triggered by specific prompt patterns.

Gu *et al.* (2023a) treated backdoor injection as multi-task learning, proposing strategies to control gradient magnitude and direction, effectively preventing backdoor forgetting during retraining. *TrojLLM* (Xue *et al.*, 2024b) generates universal, stealthy triggers in a black-box setting by querying victim LLM APIs and using a progressive Trojan poisoning algorithm. *VPI* (Yan *et al.*, 2024a) targets instruction-tuned LLMs, i.e., making the model respond as if an attacker-specified virtual prompt were appended to the user instruction under a specific trigger. Zeng *et al.* (2024a) introduced a backdoor attack that manipulates the uncertainty calibration of LLMs during training, exploiting their confidence estimation mechanisms. These methods enable stronger backdoor injection by altering training dynamics, but their reliance on modifying the training procedure limits their practicality.

3.7.3 Parameter modification. This type of attack modifies model parameters directly to embed a backdoor, typically by targeting a small subset of neurons. One representative method is *BadEdit* (Li *et al.*, 2024t) which treats backdoor injection as a lightweight knowledge-editing problem, using an efficient technique to modify LLM parameters with minimal data. Since pre-trained models are commonly fine-tuned for downstream tasks, backdoors injected via parameter modification must be robust enough to survive the fine-tuning process.

3.8 Backdoor defenses

This section reviews backdoor defense methods for LLMs, categorizing them into four types:

(1) *backdoor detection*,
(2) *backdoor removal*,
(3) *robust training*, and
(4) *robust inference*.

3.8.1 Backdoor detection. Backdoor detection identifies compromised inputs or models, flagging threats before they cause harm. Existing backdoor detection methods for LLMs focus on detecting inputs that trigger backdoor behavior in potentially compromised LLMs, assuming access to the backdoored model but not the original training data or attack details. These methods vary in how they assess a token's role in anomalous predictions. *IMBERT* (He *et al.*, 2023b) utilizes gradients and self-attention scores to identify key tokens that contribute to anomalous predictions. *AttDef* (Li *et al.*, 2023a) highlights trigger words through attribution scores, identifying those with a large impact on false predictions. *SCA* (Sun *et al.*, 2023b) fine-tunes the model to reduce trigger sensitivity, ensuring semantic consistency despite the trigger. *ParaFuzz* (Yan *et al.*, 2024b) uses input paraphrasing and compares predictions to detect trigger inconsistencies. *MDP* (Xi *et al.*, 2024) identifies critical backdoor modules and mitigates their impact by freezing relevant parameters during fine-tuning. *BEAT* (Yi *et al.*, 2025) detects triggered inputs in black-box settings by observing how concatenating a malicious probe affects the model's output distribution. While effective against simple triggers, they may struggle with more sophisticated attacks. *XBD* (Ge *et al.*, 2025) introduces a novel framework for understanding LLM backdoor attacks by leveraging model-generated explanations, contrasting clean and poisoned inputs to reveal logical inconsistencies and attention deviations induced by backdoors, thereby providing an explainability-centric approach for detecting and analyzing backdoor vulnerabilities in LLMs.

3.8.2 Backdoor removal. Backdoor removal methods aim to eliminate or neutralize the backdoor behavior embedded in a compromised model. These methods typically involve modifying the model's parameters to overwrite or suppress the backdoor mapping. We can categorize these into two groups: Pruning and Fine-tuning.

Pruning Methods aim to identify and remove model components responsible for backdoor behavior while preserving performance on clean inputs. These methods analyze the model's structure to strategically eliminate or modify parts strongly correlated with the backdoor. *PCP* Ablation (Lamparth and Reuel, 2024) targets key modules for backdoor activation, replacing them with low-rank approximations to neutralize the backdoor's influence.

Fine-tuning Methods aim to erase the malicious backdoor correlation by retraining the model on clean data. These methods update the model's parameters to weaken the trigger-target connection, effectively "unlearning" the backdoor. *SANDE* (Li *et al.*, 2024d) directly overwrites the trigger-target mapping by fine-tuning on benign-output pairs, while *CROW* (Min *et al.*, 2024a) and *BEEAR* (Zeng *et al.*, 2024b) focus on

enhancing internal consistency and counteracting embedding drift, respectively. Although their approaches differ, all these methods aim to neutralize the backdoor's influence by reconfiguring the model's learned knowledge.

3.8.3 Robust training. Robust training methods enhance the training process to ensure the resulting model remains backdoor-free, even when exposed to backdoor-poisoned data. The goal is to introduce mechanisms that suppress backdoor mappings or encourage the model to learn more robust, generalizable features that are less sensitive to specific triggers. For example, *Honeypot Defense* (Tang *et al.*, 2023a) introduces a dedicated module during training to isolate and divert backdoor features from influencing the main model. Liu *et al.* (2023g) counteracted the minimal cross-entropy loss used in backdoor attacks by encouraging a uniform output distribution through maximum entropy loss. Wang *et al.* (2024u) proposed a training-time backdoor defense that removes duplicated trigger elements and mitigates backdoor-related memorization in LLMs. Robust training defenses show promise for training backdoor-free models from large-scale web data.

3.8.4 Robust inference. Robust inference methods focus on adjusting the inference process to reduce the impact of backdoors during text generation.

Contrastive Decoding is a robust reference technique that contrasts the outputs of a potentially backdoored model with a clean reference model to identify and correct malicious outputs. For instance, *PoisonShare* (Tong *et al.*, 2024) uses intermediate layer representations in multi-turn dialogues to guide contrastive decoding, detecting and rectifying poisoned utterances. Similarly, *CleanGen* (Li *et al.*, 2024z) replaces suspicious tokens with those predicted by a clean reference model to minimize the backdoor effect. Li *et al.* (2024p) proposed ensembling the logits of the potentially compromised model with a small, benign model to mitigate malicious generations. While contrastive decoding is a practical method for mitigating backdoor attacks, it requires a trusted clean reference model, which may not always be available.

3.9 Safety alignment

The remarkable capabilities of LLMs present a unique challenge of *alignment*: how to ensure these models align with human values to avoid harmful behaviors, such as generating toxic content, spreading misinformation, or perpetuating biases. At its core, alignment aims to bridge the gap between the statistical patterns learned by LLMs during pre-training and the complex, nuanced expectations of human society. This section reviews existing works on alignment (and safety alignment) and summarizes

them into three categories: 1) *alignment with human feedback* (known as *RLHF*), 2) *alignment with AI feedback* (known as *RLAIF*), and 3) *alignment with social interactions*.

3.9.1 Alignment with human feedback. This strategy directly incorporates human preferences into the alignment process to shape the model's behavior. Existing RLHF methods can be further divided into: 1) *proximal policy optimization*, 2) *direct preference optimization*, 3) *Kahneman-Tversky optimization*, and 4) *supervised fine-tuning*.

Proximal Policy Optimization (PPO) uses human feedback as a reward signal to fine-tune LLMs, aligning model outputs with human preferences by maximizing the expected reward based on human evaluations. *InstructGPT* (Ouyang *et al.*, 2022) demonstrates its effectiveness in aligning models to follow instructions and generate high-quality responses. Refinements have further targeted stylistic control and creative generation (Ziegler *et al.*, 2019). *Safe-RLHF* (Dai *et al.*, 2024) adds safety constraints to ensure outputs remain within acceptable boundaries while maximizing helpfulness. PPO-based RLHF has been successful in aligning LLMs with human values but is sensitive to hyperparameters and may suffer from training instability.

Direct Preference Optimization (DPO) streamlines alignment by directly optimizing LLMs with human preference data, eliminating the need for a separate reward model. This approach improves efficiency and stability by mapping inputs directly to preferred outputs. *Standard DPO* (An *et al.*, 2023; Rafailov *et al.*, 2024) optimizes the model to predict preference scores, ranking responses based on human preferences. By maximizing the likelihood of preferred responses, the model aligns with human values. *MODPO* (Zhou *et al.*, 2024h) extends DPO to multi-objective optimization, balancing multiple preferences (e.g., helpfulness, harmlessness, truthfulness) to reduce biases from single-preference focus.

Kahneman-Tversky Optimization (KTO) aligns models by distinguishing between likely (desirable) and unlikely (undesirable) outcomes, making it useful when undesirable outcomes are easier to define than desirable ones. *KTO* (Ethayarajh *et al.*, 2024) uses a loss function based on prospect theory, penalizing the model more for generating unlikely continuations than rewarding it for likely ones. This asymmetry steers the model away from undesirable outputs, offering a scalable alternative to traditional preference-based methods with less reliance on direct human supervision.

Supervised Fine-Tuning (SFT) emphasizes the importance of high-quality, curated datasets to align models by training them on examples of desired outputs. *LIMA* (Zhou *et al.*, 2024a) shows that a small, well-curated dataset can achieve strong alignment

with powerful pre-trained models, suggesting that focusing on style and format in limited examples may be more effective than large datasets. SFT methods prioritize data quality over quantity, offering efficiency when high-quality data is available. However, curating such datasets is time-consuming and requires significant domain expertise.

3.9.2 Alignment with AI feedback. To overcome the scalability limitations and potential biases of relying solely on human feedback, RLAIF methods utilize AI-generated feedback to guide the alignment.

Proximal Policy Optimization These RLAIF methods adapt the PPO algorithm to incorporate AI-generated feedback, automating the process for scalable alignment and reducing human labor. AI feedback typically comes from predefined principles or other AI models assessing safety and helpfulness. *Constitutional AI* (CAI) (Bai *et al.*, 2022) uses AI self-critiques based on predefined principles to promote harmlessness. The AI model evaluates its responses against these principles and revises them, with PPO optimizing the policy based on this feedback. *SELF-ALIGN* (Sun *et al.*, 2024c) employs principle-driven reasoning and LLM generative capabilities to align models with human values. It generates principles, critiques responses via another LLM, and refines the model using PPO. *RLCD* (Yang *et al.*, 2024c) generates diverse preference pairs using contrasting prompts to train a preference model, which then provides feedback for PPO-based fine-tuning.

3.9.3 Alignment with social interactions. These methods use simulated environments to train LLMs to align with social norms and constraints, not just individual preferences. They typically employ *Contrastive Policy Optimization (CPO)* within these simulated settings.

Contrastive Policy Optimization Stable Alignment (Liu *et al.*, 2024i) uses rule-based simulated societies to train LLMs with CPO. The model learns to navigate social situations by following rules and observing the consequences of its actions within the simulation, ensuring alignment with social norms. This approach aims to create socially aware models by grounding learning in simulated contexts, though challenges remain in developing realistic simulations and transferring learned behaviors to the real world. *Monopolylogue-based Social Scene Simulation* (Pang *et al.*, 2024) introduces MATRIX, a framework where LLMs self-generate social scenarios and play multiple roles to understand the consequences of their actions. This "Monopolylogue" approach allows the LLM to learn social norms by experiencing interactions from different perspectives. The method activates the LLM's inherent knowledge of societal norms, achieving strong alignment without

external supervision or compromising inference speed. Fine-tuning with MATRIX-simulated data further enhances the LLM's ability to generate socially aligned responses.

Deceptive Alignment. While the alignment methods discussed above aim to instill desired behaviors, a critical failure mode known as *deceptive alignment* can emerge, where a model only appears to be aligned. This phenomenon underscores the difference between surface-level behavioral mimicry and genuine adherence to human values. Deceptive alignment can manifest in several ways, from superficial understanding to strategic deception. This is related to the problems of *misalignment*, where a model's goals diverge from human intent, and *reward hacking*, where a model exploits the reward function to achieve high scores without fulfilling the underlying objective. Recent work has focused on two primary forms of deceptive alignment: *fake alignment* and *alignment faking*.

Fake Alignment is a form of shallow alignment where an LLM learns to produce stylistically safe responses without a deep understanding of the underlying safety principles. Wang *et al.* (2024o) first identified this phenomenon by observing a significant performance discrepancy: LLMs that appear safe in open-ended question formats often fail safety tests presented as multiple-choice questions. They argue this is caused by *mismatched generalization*—the model memorizes the "style" of a safe refusal but cannot make a principled safety decision when forced to choose between a safe and an unsafe option. They introduced the Fake alIgNment Evaluation (FINE) framework and proposed that using multiple-choice data for contrastive distillation-based fine-tuning can improve alignment consistency.

Alignment Faking represents a more advanced and deliberate form of deception. In this scenario, a model strategically complies with its training objective to avoid being modified, thereby preserving its underlying (and potentially misaligned) preferences for later use in deployment when it is unmonitored. Greenblatt *et al.* (2024) provided the empirical demonstration of *alignment faking* in a production-scale model. By creating a fictional scenario where the model could infer whether it was "in training" or "in deployment", they observed a significant *compliance gap*—the model was more willing to comply with harmful requests when it believed it was being trained. The model's internal reasoning explicitly revealed its strategic decision to feign compliance to protect its internal values from being altered by RLHF.

Building on this, subsequent research has explored *why* some models engage in this behavior while others do not, and what motivates this deception (Sheshadri *et al.*, 2025). By extending the analysis to 25 frontier models, they found that this strategic

behavior is not widespread, but its presence in a few powerful models warrants deep investigation. They refined the motivations for alignment faking into two categories: 1) *Instrumental Goal Guarding*, where the model preserves its current goals to achieve them in the future (the classic threat model), and 2) *Terminal Goal Guarding*, where the model intrinsically values its current preferences and resists modification as an end in itself. Their findings suggest that robust refusal mechanisms developed during post-training may prevent some models from even considering strategic deception, while for others, like Claude 3 Opus, a combination of instrumental and terminal goals drives the faking behavior.

3.10 Energy latency attacks

Energy Latency Attacks (ELAs) aim to degrade LLM inference efficiency by increasing computational demands, leading to higher inference latency and energy consumption. Existing ELAs can be categorized into 1) *white-box attacks* and 2) *black-box attacks*.

3.10.1 White-box attacks. White-box attacks assume the attacker has full knowledge of the model, enabling precise manipulation of the model's inference process. These attacks can be further divided into *gradient-based attacks* and *query-based attacks* which can also be black-box.

Gradient-based Attacks use gradient information to identify input perturbations that maximize inference computations. The goal is to disrupt mechanisms essential for efficient inference, such as End-of-Sentence (EOS) prediction or early-exit. For example, *NMTSloth* (Chen *et al.*, 2022) targets EOS prediction in neural machine translation. *Engorgio* (Dong *et al.*, 2025a) crafts adversarial prompts that suppress the EOS token's appearance, forcing auto-regressive LLMs to generate abnormally long outputs. *SAME* (Chen *et al.*, 2023e) interferes with early-exit in multi-exit models. *LLMEffiChecker* (Feng *et al.*, 2024b) applies gradient-based techniques to multiple LLMs. *TTSlow* (Gao *et al.*, 2024h) induces endless speech generation in text-to-speech systems. These attacks are powerful but computationally expensive and highly model-specific, limiting their generalizability.

3.10.2 Black-box attacks. Black-box attacks do not require access to model internals, only the input-output interface. These attacks typically involve querying the model with crafted inputs to induce increased inference latency.

Query-based Attacks exploit specific model behaviors without internal access, relying on repeated querying to craft adversarial examples. *No-Skim* (Zhang *et al.*,

2023e) disrupts skimming-based models by subtly perturbing inputs to maximize retained tokens. No-Skim is ineffective against models that do not rely on skimming. Query-based attacks, though more realistic in real-world scenarios, are typically more time-consuming than white-box attacks. *Poisoning-based Attacks* manipulate model behavior by injecting malicious training samples. *P-DoS* (Gao *et al.*, 2024d) shows that a single poisoned sample during fine-tuning can induce excessively long outputs, increasing latency and bypassing output length constraints, even with limited access like fine-tuning APIs.

ELAs present an emerging threat to LLMs. Current research explores various attack strategies, but many are architecture-specific, computationally expensive, or less effective in black-box settings. Existing defenses, such as runtime input validation, can add overhead. Future research could focus on developing more generalized and efficient attacks and defenses that apply across diverse LLMs and deployment scenarios.

3.11 Model extraction attacks

Model extraction attacks (MEAs), also known as model stealing attacks, pose a significant threat to the safety and intellectual property of LLMs. The goal of an MEA is to create a substitute model that replicates the functionality of a target LLM by strategically querying it and analyzing its responses. Existing MEAs on LLMs can be categorized into two types: 1) *fine-tuning stage attacks*, and 2) *alignment stage attacks*.

3.11.1 Fine-tuning stage attacks. Fine-tuning stage attacks aim to extract knowledge from fine-tuned LLMs for downstream tasks. These attacks can be divided into two categories: *functional similarity extraction* and 2) *specific ability extraction*.

Functional Similarity Extraction seeks to replicate the overall behavior of the target fine-tuned model. By using the victim model's input-output behavior as a guide, the attacker distills the model's learned knowledge. For example, *LION* (Jiang *et al.*, 2023b) uses the victim model as a referee and generator to iteratively improve a student model's instruction-following capability.

Specific Ability Extraction targets the extraction of specific skills or knowledge the fine-tuned model has acquired. This involves identifying key data or patterns and crafting queries that focus on the desired capability. Li *et al.* (2024ab) demonstrated this by extracting coding abilities from black-box LLM APIs using carefully crafted

queries. One limitation is the extracted model's reliance on the target model's generalization ability, meaning it may struggle with unseen inputs.

3.11.2 Alignment stage attacks. Alignment stage attacks attempt to extract the alignment properties (e.g., safety, helpfulness) of the target LLM. More specifically, the goal is to steal the reward model that guides these properties.

Functional Similarity Extraction focuses on replicating the target model's alignment preferences. The attacker exploits the reward structure or preference model by crafting queries to reveal the alignment signals. *LoRD* (Liang *et al.*, 2024c) exemplifies this by using a policy-gradient approach to extract both task-specific knowledge and alignment properties. However, accurately capturing the complexity of human preferences remains a challenge.

Model extraction attacks are a rapidly evolving threat to LLMs. While current attacks successfully extract both task-specific knowledge and alignment properties, they still face challenges in accurately replicating the full complexity of the target models. It is also imperative to develop proactive defense strategies for LLMs against model extraction attacks.

3.12 Data extraction attacks

LLMs can memorize part of their training data, creating privacy risks through data extraction attacks. These attacks recover training examples, potentially exposing sensitive information such as Personal Identifiable Information (PII), copyrighted content, or confidential data (Li *et al.*, 2025c). This section reviews existing data extraction attacks, including both *white-box* and *black-box* ones.

3.12.1 White-box attacks. White-box attacks mainly focus on *Latent Memorization Extraction*, targeting information implicitly stored in model parameters or activations, which is not directly accessible through the input-output interface.

Latent Memorization Extraction reconstructs training data based on model parameters or activations. For example, Duan *et al.* (2024) developed techniques to extract latent data by analyzing internal representations, using methods like adding noise to weights or examining cross-entropy loss. These techniques were demonstrated on LLMs like Pythia-1B and Amber-7B. While these attacks reveal risks associated with internal data representation, they require full access to the model parameters, which remains a major limitation in practice.

3.12.2 Black-box attacks. Black-box data extraction attacks are a realistic threat, where attackers craft inductive prompts to trick LLMs into revealing memorized training data, without access to their parameters.

Prefix Attacks exploit the autoregressive nature of LLMs by providing a "prefix" from a memorized sequence, hoping the model will continue it. Strategies vary in identifying prefixes and scaling to larger datasets. Carlini *et al.* (2019) demonstrated this on models like GPT-2, while Nasr *et al.* (2023) scaled prefix attacks using suffix arrays. *Magpie* (Xu *et al.*, 2024g) and Al-Kaswan *et al.* (2024) targeted specific data, such as PII or code. Yu *et al.* (2023b) enhanced black-box data extraction by optimizing text continuation generation and ranking. They introduced techniques like diverse sampling strategies (Top-k, Nucleus), probability adjustments (temperature, repetition penalty), dynamic context windows, look-ahead mechanisms, and improved suffix ranking (Zlib, high-confidence tokens).

Special Character Attack exploits the model's sensitivity to special characters or unusual input formatting, potentially triggering unexpected behavior that reveals memorized data. *SCA* (Bai *et al.*, 2024c) demonstrates that specific characters can indeed induce LLMs to disclose training data. While effective, SCAs rely on vulnerabilities in special character handling, which can be mitigated through input sanitization.

Prompt Optimization employs an "attacker" LLM to generate optimized prompts that extract data from a "victim" LLM. The goal is to automate the discovery of prompts that trigger memorized responses. Kassem *et al.* (2024) demonstrated this by using an attacker LLM with iterative rejection sampling and longest common subsequence (LCS) for optimization. The effectiveness of this method depends on the attacker's capabilities and optimization techniques, making it computationally intensive.

Retrieval-Augmented Generation (RAG) Extraction targets RAG systems, aiming to leak sensitive information from the retrieval component. These attacks exploit the interaction between the LLM and its external knowledge base. Qi *et al.* (2024b) demonstrated that adversarial prompts can trigger data leakage in RAG systems. Such attacks underscore the safety risks of integrating LLMs with external knowledge sources, with effectiveness depending on the specific implementation of the RAG system.

Ensemble Attack combines multiple attack strategies to enhance effectiveness, leveraging the strengths of each method for higher success rates. More *et al.* (2024) demonstrated the effectiveness of such an ensemble approach on Pythia. While

powerful, ensemble attacks are complex and require careful coordination among the attack components.

Semantic Information Elicitation shifts the focus from extracting verbatim training data to generating sensitive semantic content. Zhang *et al.* (2025f) demonstrated that even simple, natural questions can prompt LLMs to output Semantic Sensitive Information (SemSI), such as personal beliefs or reputation-harmful statements, and proposed a benchmark to systematically evaluate this risk.

3.13 Datasets and benchmarks

This section reviews commonly used datasets and benchmarks in LLM safety research, as shown in Table 4. These datasets and benchmarks are categorized based

Table 4. Datasets and benchmarks for LLM safety research

Dataset	Year	Size	#Times
RealToxicityPrompts (Gehman *et al.*, 2020)	2020	100K	135
TruthfulQA (Lin *et al.*, 2022)	2021	817	213
AdvGLUE (Wang *et al.*, 2021)	2021	5,716	12
SafetyPrompts (Sun *et al.*, 2023a)	2023	100K	15
DoNotAnswer (Wang *et al.*, 2024q)	2023	939	6
AdvBench (Zou *et al.*, 2023)	2023	520	52
CVALUES (Xu *et al.*, 2023)	2023	2,100	10
FINE (Wang *et al.*, 2024o)	2023	90	14
FLAMES (Huang *et al.*, 2024c)	2024	2,251	17
SORRYBench (Xie *et al.*, 2024)	2024	450	8
SafetyBench (Zhang *et al.*, 2024x)	2024	11,435	21
SALAD-Bench (Li *et al.*, 2024j)	2024	30K	36
BackdoorLLM (Li *et al.*, 2024w)	2024	8	6
JailBreakV-28K (Luo *et al.*, 2024c)	2024	28K	10
STRONGREJECT (Souly *et al.*, 2024)	2024	313	4
Libra-Leaderboard (Li *et al.*, 2024e)	2024	57	26
Aegis 2.0 (Ghosh *et al.*, 2025)	2025	34K	17
CASE-Bench (Sun *et al.*, 2025)	2025	450	–

on their evaluation purpose: *toxicity datasets*, *truthfulness datasets*, *value benchmarks*, and *adversarial datasets and backdoor benchmarks*.

3.13.1 Toxicity datasets. Ensuring LLMs do not generate harmful content is crucial for safety. Early work, such as the *RealToxicityPrompts* dataset (Gehman *et al.*, 2020), exposed the tendency of LLMs to produce toxic text from benign prompts. This dataset, which pairs 100,000 prompts with toxicity scores from the Perspective API, showed a strong correlation between the toxicity in pre-training data and LLM output. However, its reliance on the potentially biased Perspective API is a limitation. To address broader harmful behaviors, the *Do-Not-Answer* (Wang *et al.*, 2024q) dataset was introduced. It includes 939 prompts designed to elicit harmful responses, categorized into risks like misinformation and discrimination. Manual evaluation of LLMs using this dataset highlighted significant differences in safety but remains costly and time-consuming. A recent approach by Cheng *et al.* (2024b) introduces a crowd-sourced toxic question and response dataset, with annotations from both humans and LLMs. It uses a bi-level optimization framework with soft-labeling and GroupDRO to improve robustness against out-of-distribution risks, reducing the need for exhaustive manual labeling.

3.13.2 Truthfulness datasets. Ensuring LLMs generate truthful information is also essential. The *TruthfulQA* benchmark (Lin *et al.*, 2022) evaluates whether LLMs provide accurate answers to 817 questions across 38 categories, specifically targeting "imitative falsehoods"—false answers learned from human text. Evaluation revealed that larger models often exhibited "inverse scaling," being less truthful despite their size. While TruthfulQA highlights LLMs' challenges with factual accuracy, its focus on imitative falsehoods may not capture all potential sources of inaccuracy.

3.13.3 Value benchmarks. Ensuring LLM alignment with human values is a critical challenge, addressed by several benchmarks assessing various aspects of safety, fairness, and ethics. *FLAMES* (Huang *et al.*, 2024c) evaluates the alignment of Chinese LLMs with values like fairness, safety, and morality through 2,251 prompts. *SORRY-Bench* (Xie *et al.*, 2024) assesses LLMs' ability to reject unsafe requests using 45 topic categories, while *CVALUES* (Xu *et al.*, 2023) focuses on both safety and responsibility. *SafetyPrompts* (Sun *et al.*, 2023a) evaluates Chinese LLMs on a range of ethical scenarios. While these benchmarks are valuable, they often focus on isolated, problematic queries, potentially leading to over-refusal in safe contexts. To address this, recent benchmarks have begun to incorporate contextual information. *CASE-Bench* (Sun *et al.*, 2025) pioneers this by using Contextual Integrity (CI) theory to formally describe the context of a query, evaluating whether an LLM's safety

judgment aligns with human judgment under different contexts. This work reveals that context significantly influences human safety assessments and highlights mismatches in LLM behavior, especially in safe contexts.

In parallel, creating high-quality, commercially-usable datasets is crucial for training robust safety guardrails. *AEGIS2.0* (Ghosh *et al.*, 2025) addresses this gap by providing a diverse dataset with a comprehensive taxonomy of 12 core and 9 fine-grained risk categories. It uses a hybrid data generation pipeline combining human annotation with a multi-LLM "jury" system, making it suitable for training commercial safety models. Furthermore, the concept of "*fake alignment*" (Wang *et al.*, 2024o) highlights the risk of LLMs superficially memorizing safety answers, leading to the Fake alIgNment Evaluation (*FINE*) framework for consistency assessment. *SafetyBench* (Zhang *et al.*, 2024x) addresses this by providing an efficient, automated multiple-choice benchmark for LLM safety evaluation. *Libra-Leaderboard* (Li *et al.*, 2024e) introduces a balanced leaderboard for evaluating both the safety and capability of LLMs. It features a comprehensive safety benchmark with 57 datasets covering diverse safety dimensions, a unified evaluation framework, an interactive safety arena for adversarial testing, and a balanced scoring system. Libra-Leaderboard promotes a holistic approach to LLM evaluation, representing a significant step towards responsible AI development.

3.13.4 Adversarial datasets and backdoor benchmarks. BackdoorLLM (Li *et al.*, 2024w) is the first benchmark for evaluating backdoor attacks in text generation, offering a standardized framework that includes diverse attack strategies like data poisoning and weight poisoning. *Adversarial GLUE* (Wang *et al.*, 2021) assesses LLM robustness against textual attacks using 14 methods, highlighting vulnerabilities even in robustly trained models. *SALAD-Bench* (Li *et al.*, 2024j) expands on this by introducing a safety benchmark with a taxonomy of risks, including attack- and defense-enhanced questions. *JailBreakV-28K* (Luo *et al.*, 2024c) focuses on evaluating multi-modal LLMs against jailbreak attacks using text- and image-based test cases. A *STRONGREJECT* for empty jailbreaks (Souly *et al.*, 2024) improves jailbreak evaluation with a higher-quality dataset and automated assessment. Despite their value, these benchmarks face challenges in scalability, consistency, and real-world relevance.

4. Vision-language pre-training model safety

VLP models, such as CLIP (Radford *et al.*, 2021), ALBEF (Li *et al.*, 2021), and TCL (Yang *et al.*, 2022), have made significant strides in aligning visual and textual modalities. However, these models remain vulnerable to various safety threats, which have garnered increasing research attention. This section reviews the current safety research on VLP models, with a focus on adversarial, backdoor, and poisoning research. The representative methods reviewed in this section are summarized in Table 5.

4.1 Adversarial attacks

Since VLP models are widely used as backbones for fine-tuning downstream models, adversarial attacks on VLP aim to generate examples that cause incorrect predictions across various downstream tasks, including zero-shot image classification, image-text retrieval, visual entailment, and visual grounding. Similar to Section 2, these attacks can roughly be categorized into *white-box attacks* and *black-box attacks*, based on their threat models.

4.1.1 White-box attacks. White-box adversarial attacks on VLP models can be further categorized based on perturbation types into *invisible perturbations* and *visible perturbations*, with the majority of existing attacks employing invisible perturbations.

Invisible Perturbations involve small, imperceptible adversarial changes to inputs—whether text or images—to maintain the stealthiness of attacks. Early research in the vision and language domains primarily adopts this approach (Xu *et al.*, 2018; Shah *et al.*, 2019; Li *et al.*, 2020; Yang *et al.*, 2021), in which invisible attacks

Table 5. A summary of attacks and defenses for VLP models

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
Adversarial Attack	Co-Attack (Zhang *et al.*, 2022)	2022	White-box	Invisible	ALBEF, TCL, CLIP	MS-COCO, Flickr30K, RefCOCO +, SNLI-VE
	AdvCLIP (Zhou *et al.*, 2023)	2023	White-box	Invisible	CLIP	STL10, GTSRB, CIFAR10, ImageNet, Wikipedia, Pascal-Sentence, NUS-WIDE, XmediaNet
	Typographical Attacks (Noever and Noever, 2021)	2021	White-box	Visible	CLIP	ImageNet
	Multi-Image Typographical Attacks (Wang *et al.*, 2025e)	2025	White-box	Visible	OpenCLIP, InstructBLIP	ImageNet, LAION
	SGA (Lu *et al.*, 2023)	2023	Black-box	Sample-wise	ALBEF, TCL, CLIP	Flickr30K, MS-COCO
	SA-Attack (He *et al.*, 2023a)	2023	Black-box	Sample-wise	ALBEF, TCL, CLIP	Flickr30K, MS-COCO
	VLP-Attack (Wang *et al.*, 2023d)	2023	Black-box	Sample-wise	ALBEF, TCL, BLIP, BLIP2, MiniGPT-4	MS-COCO, Flickr30K, SNLI-VE
	TMM (Wang *et al.*, 2024c)	2024	Black-box	Sample-wise	ALBEF, TCL, X_VLM, CLIP, BLIP, ViLT, METER	MS-COCO, Flickr30K, RefCOCO +, SNLI-VE

(*continued*)

Table 5. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
	VLATTACK (Yin *et al.*, 2023)	2023	Black-box	Sample-wise	BLIP, ViLT, CLIP	MS-COCO, VQA v2, NLVR2, SNLI-VE, ImageNet, SVHN
	OT-Attack (Han *et al.*, 2023a)	2023	Black-box	Sample-wise	CLIP, ALBEF, TCL	Flickr30K, MS-COCO, RefCOCO+
	PRM (Hu *et al.*, 2024a)	2024	Black-box	Sample-wise	CLIP, Detic, VL-PLM, FC-CLIP, OpenFlamingo, LLaVA	PASCAL Context, COCO-Stuff, OV-COCO, MS-COCO, OK-VQA
	VLPTransfer Attack (Gao *et al.*, 2024f)	2024	Black-box	Sample-wise	CLIP, ALBEF, TCL	Flickr30K, MS-COCO, RefCOCO+
	C-PGC (Fang *et al.*, 2024a)	2024	Black-box	Universal	ALBEF, TCL, X-VLM, CLIP, BLIP	Flickr30K, MS-COCO, SNLI-VE, RefCOCO+
	ETU (Zhang *et al.*, 2024n)	2024	Black-box	Universal	ALBEF, TCL, CLIP, BLIP	Flickr30K, MS-COCO
	X-Transfer (Huang *et al.*, 2025b)	2025	Black-box	Universal	CLIP, OpenFlamingo, LLaVA, BLIP2, MiniGPT4	Flickr30K, MS-COCO, ImageNet, CIFAR10, CIFAR100, STL10, SUN397, FOOD101, GTSRB StandfordCars, OK-VQA, VizWiz

(*continued*)

Table 5. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
Adversarial Defense	Defense-Prefix (Azuma and Matsui, 2023)	2023	Adversarial Tuning	Prompt Tuning	CLIP	ImageNet
	AdvPT (Zhang *et al.*, 2024i)	2023	Adversarial Tuning	Prompt Tuning	CLIP	ImageNet, Pets, Flowers, Food101, SUN397, DTD, EuroSAT, UCF101, ImageNet-V2, ImageNet-Sketch, ImageNet-A, ImageNet-R
	APT (Li *et al.*, 2024k)	2024	Adversarial Tuning	Prompt Tuning	CLIP	ImageNet, Caltech101, Pets, StanfordCars, Flowers, Food101, FGVCAircraft, SUN397, DTD, EuroSAT, UCF101, ImageNet-V2, ImageNet-Sketch, ImageNet-R, ObjectNet

(*continued*)

Table 5. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
	MixPrompt (Fan *et al.*, 2024)	2024	Adversarial Tuning	Prompt Tuning	CLIP	ImageNet, Pets, Flowers, DTD, EuroSAT, UCF101, SUN397, Food101, ImageNet-V2, ImageNet-Sketch, ImageNet-A, ImageNet-R
	PromptSmoot (Hussein *et al.*, 2024)	2024	Adversarial Tuning	Prompt Tuning	PLIP, Quilt, MedCLIP	KatherColon, PanNuke, SkinCancer, SICAP v2
	FAP (Zhou *et al.*, 2024g)	2024	Adversarial Tuning	Prompt Tuning	CLIP	ImageNet, Caltech101, Pets, StanfordCars, Flowers, Food101, FGVCAircraft, SUN397, DTD, EuroSAT, UCF101
	APD (Luo *et al.*, 2024b)	2024	Adversarial Tuning	Prompt Tuning	CLIP	ImageNet, Caltech101, Flowers, Food101, SUN397, DTD, EuroSAT, UCF101

(*continued*)

Table 5. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
	TAPT (Wang *et al.*, 2025f)	2025	Adversarial Tuning	Prompt Tuning	CLIP	ImageNet, Caltech101, Pets, StanfordCars, Flowers, Food101, FGVCAircraft, SUN397, DTD, EuroSAT, UCF101
	TeCoA (Mao *et al.*, 2023)	2022	Adversarial Tuning	Contrastive Tuning	CLIP	CIFAR10, CIFAR100, STL10, Caltech101, Caltech256, Pets, StanfordCars, Food101, Flowers, FGVCAircraft, SUN397, DTD, PCAM, HatefulMemes, EuroSAT

(*continued*)

Table 5. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
	PMG-AFT (Wang *et al.*, 2024j)	2024	Adversarial Tuning	Contrastive Tuning	CLIP	CIFAR10, CIFAR100, STL10, ImageNet, Caltech101, Caltech256, Pets, Flowers, FGVCAircraft, StanfordCars, SUN397, Food101, EuroSAT, DTD, PCAM
	MMCoA (Zhou *et al.*, 2024d)	2024	Adversarial Tuning	Contrastive Tuning	CLIP	CIFAR10, CIFAR100, TinyImageNet, STL10, Caltech101, Caltech256, Pets, Flowers, FGVCAircraft, Food101, EuroSAT, DTD, SUN397, Country211

(*continued*)

Table 5. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
	FARE (Schlarmann *et al.*, 2024)	2024	Adversarial Tuning	Contrastive Tuning	OpenFlamingo, LLaVA	COCO, Flickr30k, TextVQA, VQA v2, CalTech101, StanfordCars, CIFAR10, CIFAR100, DTD, EuroSAT, FGVCAircrafts, Flowers, ImageNet-R, ImageNet-Sketch, PCAM, Pets, STL10, ImageNet
	VILLA (Gan *et al.*, 2020)	2020	Adversarial Training	Two-stage Training	UNITER, LXMERT	MS-COCO, Visual Genome, Conceptual Captions, SBU Captions ImageNet, LAION, DataComp
	AdvXL (Wang *et al.*, 2024s)	2024	Adversarial Training	Two-stage Training	CLIP	ImageNet, LAION, DataComp
	MirrorCheck (Fares *et al.*, 2024)	2024	Adversarial Detection	One-shot Detection	UniDiffuser, BLIP, Img2Prompt, BLIP-2, MiniGPT-4	MS-COCO, CIFAR10, ImageNet
	AdvQDet (Wang *et al.*, 2024k)	2024	Adversarial Detection	Stateful Detection	CLIP, ViT, ResNet	CIFAR10, GTSRB, ImageNet, Flowers, Pets

(*continued*)

Table 5. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
Backdoor & Poisoning Attack	PBCL (Carlini and Terzis, 2022)	2021	Backdoor& Poisoning	Visual Trigger	CLIP	Conceptual Captions, YFCC
	BadEncoder (Jia *et al.*, 2022)	2021	Backdoor	Visual Trigger	ResNet(SimCLR), CLIP	CIFAR10, STL10, GTSRB, SVHN, Food101
	Corrupt Encoder (Zhang *et al.*, 2024l)	2022	Backdoor	Visual Trigger	ResNet(SimCLR)	ImageNet, Pets, Flowers
	BadCLIP (Liang *et al.*, 2024b)	2023	Backdoor	Visual Trigger	CLIP	Conceptual Captions
	BadCLIP (Bai *et al.*, 2024a)	2023	Backdoor	Multi-modal Trigger	CLIP	ImageNet, Caltech101, Pets, StanfordCars, Flowers, Food101, FGVCAircraft, SUN397, DTD, EuroSAT, UCF101
	MM Poison (Yang *et al.*, 2023)	2022	Poisoning	Multi-modal Poisoning	CLIP	Flickr-PASCAL, MS-COCO
	MEM Liu *et al.* 2024n	2024	Poisoning	Multi-modal Trigger	CLIP	Flickr8k, Flickr30k, MS-COCO

(*continued*)

Table 5. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
Backdoor & Poisoning Defense	CleanCLIP (Bansal *et al.*, 2023)	2023	Backdoor Removal	Fine-tuning	CLIP	Conceptual Captions, ImageNet
	SAFECLIP (Yang *et al.*, 2024f)	2023	Backdoor Removal	Fine-tuning	CLIP	Conceptual Captions, Visual Genome, MS-COCO, Flowers, Food101, ImageNet, Pets, StanfordCars, Caltech101, CIFAR10, CIFAR100, DTD, FGVCAircraft
	RoCLIP (Yang *et al.*, 2024g)	2023	Robust Training	Pre-training	CLIP	Conceptual Captions, Flowers, Food101, ImageNet, Pets, StanfordCars, Caltech101, CIFAR10, CIFAR100, DTD, FGVCAircraft
	DECREE (Feng *et al.*, 2023)	2023	Backdoor Detection	Backdoor Model Detection	CLIP	CIFAR10, GTSRB, SVHN, STL-10, ImageNet
	TIJO (Sur *et al.*, 2023)	2023	Backdoor Detection	Trigger Inversion	BUTD, MFB, BAN, MCAN, NAS	TrojVQA

(*continued*)

Table 5. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target model	Dataset
	Mudjacking (Liu *et al.*, 2024g)	2024	Backdoor Detection	Trigger Inversion	CLIP	Conceptual Captions, CIFAR10, STL10, ImageNet, SVHN, Pets, Wiki103-Sub, SST-2, HOSL
	SEER (Zhu *et al.*, 2024a)	2024	Backdoor Detection	Backdoor Sample Detection	CLIP	MSCOCO, Flickr, STL10, Pet, ImageNet
	Outlier Detection (Huang *et al.*, 2025a)	2025	Backdoor Detection	Backdoor Sample Detection	CLIP	Conceptual Captions, ImageNet, RedCaps

are developed independently. In the context of VLP models, which integrate both modalities, *Co-Attack* (Zhang *et al.*, 2022) was the first to propose perturbing both visual and textual inputs simultaneously to create stronger attacks. Building on this, *AdvCLIP* (Zhou *et al.*, 2023) explores universal adversarial perturbations that can deceive all downstream tasks.

Visible Perturbations involve more substantial and noticeable alterations. For example, manually crafted typographical, conceptual, and iconographic images have been used to demonstrate that the CLIP model tends to "read first, look later" (Noever and Noever, 2021), highlighting a unique characteristic of VLP models. This behavior introduces new attack surfaces for VLP, enabling the development of more sophisticated attacks. Recent work by Wang *et al.* (2025e) introduced a more stealthy multi-image attack scenario, demonstrating that non-repeating typographic attacks are most effective when attack texts are strategically selected for their similarity to the target images.

4.1.2 Black-box attacks. Black-box attacks on VLP primarily adopt a transfer-based approach, with query-based attacks rarely explored. Existing methods can be categorized into: 1) *sample-specific perturbations*, tailored to individual samples, and 2) *universal perturbations*, applicable across multiple samples.

Sample-wise perturbations are generally more effective than universal perturbations, but their transferability is often limited. *SGA* (Lu *et al.*, 2023) explores adversarial transferability in VLP by leveraging cross-modal interactions and alignment-preserving augmentation. Building on this, *SA-Attack* (He *et al.*, 2023a) enhances cross-modal transferability by introducing data augmentations to both original and adversarial inputs. *VLP-Attack* (Wang *et al.*, 2023d) improves transferability by generating adversarial texts and images using contrastive loss. To overcome SGA's limitations, *TMM* (Wang *et al.*, 2024c) introduces modality-consistency and discrepancy features through attention-based and orthogonal-guided perturbations. *VLATTACK* (Yin *et al.*, 2023) further enhances adversarial examples by combining image and text perturbations at both single-modal and multimodal levels. *PRM* (Hu *et al.*, 2024a) targets vulnerabilities in downstream models using foundation models like CLIP, enabling transferable attacks across tasks like object detection and image captioning. In parallel, Gao *et al.* (2024f) enhanced black-box transferability by diversifying perturbations within the "intersection region" of the adversarial trajectory, reducing overfitting to the source model. Similarly, OT-Attack (Han *et al.*, 2023a) addressed overfitting by using optimal transport theory to efficiently align augmented image and text distributions.

Universal Perturbations are less effective than sample-wise perturbations but more transferable. *C-PGC* (Fang *et al.*, 2024a) was the first to investigate universal adversarial perturbations (UAPs) for VLP models. It employs contrastive learning and cross-modal information to disrupt the alignment of image-text embeddings, achieving stronger attacks in both white-box and black-box scenarios. *ETU* (Zhang *et al.*, 2024n) builds on this by generating UAPs that transfer across multiple VLP models and tasks. ETU enhances UAP transferability and effectiveness through improved global and local optimization techniques. It also introduces a data augmentation strategy *ScMix* that combines self-mix and cross-mix operations to increase data diversity while preserving semantic integrity, further boosting the robustness and applicability of UAPs. *X-Transfer* (Huang *et al.*, 2025b) proposes an efficient scaling strategy that enables the ensembling of a large collection of CLIP encoders as surrogate models. This approach demonstrates super adversarial transferability, achieving simultaneous transfer across data distributions, domains, model architectures, downstream tasks, and even to large VLMs.

4.2 Adversarial defenses

Existing adversarial defenses for VLP models can be grouped into four types: 1) *adversarial example detection*, 2) *standard adversarial training*, 3) *adversarial prompt tuning*, and 4) *adversarial contrastive tuning*. While adversarial detection filters out potential adversarial examples before or during inference, the other three defenses follow similar adversarial training paradigms, with variations in efficiency.

4.2.1 Adversarial example detection. Adversarial detection methods for VLP can be further divided into *one-shot detection* and *stateful detection*.

One-shot Detection. One-shot Detection distinguishes adversarial from clean examples in a single forward pass. White-box detection methods are typically one-shot. For example, *MirrorCheck* (Fares *et al.*, 2024) is a model-agnostic method for VLP models. It uses text-to-image (T2I) models to generate images from captions produced by the victim model, comparing the similarity between the input image and the generated image using CLIP's image encoder. A significant similarity difference flags the input as adversarial.

Stateful Detection. Stateful Detection is designed for black-box query attacks, where multiple queries are tracked to detect adversarial behavior. *AdvQDet*

(Wang *et al.*, 2024k) is a novel framework that counters query-based black-box attacks. It uses adversarial contrastive prompt tuning (ACPT) to tune CLIP image encoder, enabling detection of adversarial queries within just three queries.

4.2.2 Standard adversarial training. Adversarial training is widely regarded as the most effective defense against adversarial attacks (Madry *et al.*, 2018; Croce and Hein, 2020). However, it is computationally expensive, and for VLP models, which are typically trained on web-scale datasets, this cost becomes prohibitively high, posing a significant challenge for traditional approaches. Although research in this area is limited, we highlight two notable works that have explored adversarial training for vision-language pre-training. Their pre-trained models can be used as robust backbones for other adversarial research.

The first work, *VILLA* (Gan *et al.*, 2020), is a vision-language adversarial training framework consisting of two stages: task-agnostic adversarial pre-training and task-specific fine-tuning. VILLA enhances performance across downstream tasks using adversarial pre-training in the embedding space of both image and text modalities, instead of pixel or token levels. It employs FreeLB's strategy (Zhu *et al.*, 2020) to minimize computational overhead for efficient large-scale training.

The second work, *AdvXL* (Wang *et al.*, 2024s), is a large-scale adversarial training framework with two phases: a lightweight pre-training phase using low-resolution images and weaker attacks, followed by an intensive fine-tuning phase with full-resolution images and stronger attacks. This coarse-to-fine, weak-to-strong strategy reduces training costs while enabling scalable adversarial training for large vision models.

4.2.3 Adversarial prompt tuning. Adversarial prompt tuning (APT) enhances the adversarial robustness of VLP models by incorporating adversarial training during prompt tuning (Zhou *et al.*, 2022a, b; Khattak *et al.*, 2023), typically focusing on textual prompts. It offers a lightweight alternative to standard adversarial training. APT methods can be classified into two main categories based on the prompt type: *textual prompt tuning* and *multi-modal prompt tuning*.

Textual Prompt Tuning. Textual prompt tuning (TPT) robustifies VLP models by fine-tuning learnable text prompts. *AdvPT* (Zhang *et al.*, 2024i) enhances the adversarial robustness of CLIP image encoder by realigning adversarial image embeddings with clean text embeddings using learnable textual prompts. Similarly, *APT* (Li *et al.*, 2024k) learns robust text prompts, using a CLIP image encoder to boost accuracy and robustness with minimal computational cost. *MixPrompt* (Fan *et al.*, 2024) simultaneously enhances the generalizability and adversarial robustness of

VLPs by employing conditional APT. Unlike empirical defenses, *PromptSmooth* (Hussein *et al.*, 2024) offers a certified defense for Medical VLMs, adapting pre-trained models to Gaussian noise without retraining. Additionally, *Defense-Prefix* (Azuma and Matsui, 2023) mitigates typographic attacks by adding a prefix token to class names, improving robustness without retraining.

Multi-Modal Prompt Tuning. Recent adversarial prompt tuning methods have expanded textual prompts to multi-modal prompts. *FAP* (Zhou *et al.*, 2024g) introduces learnable adversarial text supervision and a training objective that balances cross-modal consistency while differentiating uni-modal representations. *APD* (Luo *et al.*, 2024b) improves CLIP's robustness through online prompt distillation between teacher and student multi-modal prompts. Additionally, *TAPT* (Wang *et al.*, 2025f) presents a test-time defense that learns defensive bimodal prompts to improve CLIP's zero-shot inference robustness.

4.2.4 Adversarial contrastive tuning. Adversarial contrastive tuning involves contrastive learning with adversarial training to fine-tune a robust CLIP image encoder for *zero-shot adversarial robustness* on downstream tasks. These methods are categorized into *supervised* and *unsupervised* methods, depending on the availability of labeled data during training.

Supervised Contrastive Tuning. *Visual Tuning* fine-tunes CLIP image encoder using only adversarial images. *TeCoA* (Mao *et al.*, 2023) explores the zero-shot adversarial robustness of CLIP and finds that visual prompt tuning is more effective without text guidance, while fine-tuning performs better with text information. *PMG-AFT* (Wang *et al.*, 2024j) improves zero-shot adversarial robustness by introducing an auxiliary branch to minimize the distance between adversarial outputs in the target and pre-trained models, mitigating overfitting and preserving generalization.

Multi-modal Tuning fine-tunes CLIP image encoder using both adversarial texts and images. *MMCoA* (Zhou *et al.*, 2024d) combines image-based PGD and text-based BERT-Attack in a multi-modal contrastive adversarial training framework. It uses two contrastive losses to align clean and adversarial image and text features, improving robustness against both image-only and multi-modal attacks.

Unsupervised Contrastive Tuning. Adversarial contrastive tuning can also be performed in an unsupervised fashion. For instance, *FARE* (Schlarmann *et al.*, 2024) robustifies CLIP image encoder through unsupervised adversarial fine-tuning, achieving superior clean accuracy and robustness across downstream tasks, including zero-shot classification and vision-language tasks. This approach enables VLMs, such

as LLaVA and OpenFlamingo, to attain robustness without the need for re-training or additional fine-tuning.

4.3 Backdoor and poisoning attacks

Backdoor and poisoning attacks on CLIP can target either the pre-training stage or the fine-tuning stage on downstream tasks. Previous studies have shown that poisoning backdoor attacks on CLIP can succeed with significantly lower poisoning rates compared to traditional supervised learning (Carlini and Terzis, 2022). Additionally, training CLIP on web-crawled data increases its vulnerability to backdoor attacks (Carlini *et al.*, 2024a). This section reviews proposed attacks targeting backdooring or poisoning CLIP.

4.3.1 Backdoor attacks. Based on the trigger modality, existing backdoor attacks on CLIP can be categorized into *visual triggers* and *multi-modal triggers*.

Visual Triggers target pre-trained image encoders by embedding backdoor patterns in visual inputs. *BadEncoder* (Jia *et al.*, 2022) explores image backdoor attacks on self-supervised learning by injecting backdoors into pre-trained image encoders, compromising downstream classifiers. *CorruptEncoder* (Zhang *et al.*, 2024l) exploits random cropping in contrastive learning to inject backdoors into pre-trained image encoders, with increased effectiveness when cropped views contain only the reference object or the trigger. For attacks targeting CLIP, *BadCLIP* (Liang *et al.*, 2024b) optimizes visual trigger patterns using dual-embedding guidance, aligning them with both the target text and specific visual features. This strategy enables BadCLIP to bypass backdoor detection and fine-tuning defenses.

Multi-modal Triggers combine both visual and textual triggers to enhance the attack. BadCLIP (Bai *et al.*, 2024a) introduces a novel trigger-aware prompt learning-based backdoor attack targeting CLIP models. Rather than fine-tuning the entire model, BadCLIP injects learnable triggers during the prompt learning stage, affecting both the image and text encoders.

4.3.2 Poisoning attacks. Two targeted poisoning attacks on CLIP are *PBCL* (Carlini and Terzis, 2022) and *MM Poison* (Yang *et al.*, 2023). PBCL demonstrated that a targeted poisoning attack, misclassifying a specific sample, can be achieved by poisoning as little as 0.0001% of the training dataset. MM Poison investigates modality vulnerabilities and proposes three attack types: single target image, single target label, and multiple target labels. Evaluations show high attack success rates while maintaining clean data performance across both visual and textual modalities.

MEM (Liu *et al.*, 2024n) protects private data from exploitation in multimodal contrastive learning by crafting unlearnable examples: it adds imperceptible noise to images and inserts an optimized text trigger into captions.

4.4 Backdoor and poisoning defenses

Defense strategies against backdoor and poisoning attacks are generally categorized into *robust training* and *backdoor detection*. Robust Training aims to create VLP models resistant to backdoor or targeted poisoning attacks, even when trained on untrusted datasets. This approach specifically addresses poisoning-based attacks. Backdoor detection focuses on identifying compromised encoders or contaminated data. Detection methods often require additional mitigation techniques to fully eliminate backdoor effects.

4.4.1 Robust training. Depending on the stage at which the model gains robustness against backdoor attacks, existing robust training strategies can be categorized into fine-tuning and pre-training approaches.

Fine-tuning Stage. To mitigate backdoor and poisoning threats, *CleanCLIP* (Bansal *et al.*, 2023) fine-tunes CLIP by re-aligning each modality's representations, weakening spurious correlations from backdoor attacks. Similarly, *SAFECLIP* (Yang *et al.*, 2024f) enhances feature alignment using unimodal contrastive learning. It first warms up the image and text modalities separately, then uses a Gaussian mixture model to classify data into safe and risky sets. During pre-training, SAFECLIP optimizes CLIP loss on the safe set, while separately fine-tuning the risky set, reducing poisoned image-text pair similarity and defending against targeted poisoning and backdoor attacks.

Pre-training Stage. *ROCLIP* (Yang *et al.*, 2024g) defends against poisoning and backdoor attacks by enhancing model robustness during pre-training. It disrupts the association between poisoned image-caption pairs by utilizing a large, diverse pool of random captions. Additionally, ROCLIP applies image and text augmentations to further strengthen its defense and improve model performance.

4.4.2 Backdoor detection. Backdoor detection can be broadly divided into three subtasks: 1) *trigger inversion*, 2) *backdoor sample detection*, and 3) *backdoor model detection*. Trigger inversion is particularly useful, as recovering the trigger can aid in the detection of both backdoor samples and backdoored models.

Trigger Inversion aims to reverse-engineer the trigger pattern injected into a backdoored model. *Mudjacking* (Liu *et al.*, 2024g) mitigates backdoor vulnerabilities

in VLP models by adjusting model parameters to remove the backdoor when a misclassified trigger-embedded input is detected. In contrast to single-modality defenses, *TIJO* (Sur *et al.*, 2023) defends against dual-key backdoor attacks by jointly optimizing the reverse-engineered triggers in both the image and text modalities.

Backdoor Sample Detection detects whether a training or test sample is poisoned by a backdoor trigger. This detection can be used to cleanse the training dataset or reject backdoor queries. *SEER* (Zhu *et al.*, 2024a) addresses the complexity of multi-modal models by jointly detecting malicious image triggers and target texts in the shared feature space. This method does not require access to the training data or knowledge of downstream tasks, making it highly effective for backdoor detection in VLP models. *Outlier Detection* (Huang *et al.*, 2025a) demonstrates that the local neighborhood of backdoor samples is significantly sparser compared to that of clean samples. This insight enables the effective and efficient application of various local outlier detection methods to identify backdoor samples from web-scale datasets. Furthermore, they reveal that potential unintentional backdoor samples already exist in the Conceptual Captions 3 Million (CC3M) dataset and have been trained into open-sourced CLIP encoders.

Backdoor Model Detection identifies whether a trained model is compromised by backdoor(s). *DECREE* (Feng *et al.*, 2023) introduces a backdoor detection method specifically for VLP encoders that require no labeled data. It exploits the distinct embedding space characteristics of backdoored encoders when exposed to clean versus backdoor inputs. By combining trigger inversion with these embedding differences, DECREE can effectively detect backdoored encoders.

4.5 Datasets

This section reviews datasets used for VLP safety research. As shown in Table 5, a variety of benchmark datasets were employed to evaluate adversarial attacks and defenses for VLP models. For image classification tasks, commonly used datasets include: ImageNet (Russakovsky *et al.*, 2015), Caltech101 (Fei-Fei *et al.*, 2004), DTD (Cimpoi *et al.*, 2014), EuroSAT (Helber *et al.*, 2019), OxfordPets (Parkhi *et al.*, 2012), FGVC-Aircraft (Maji *et al.*, 2013), Food101 (Bossard *et al.*, 2014), Flowers102 (Nilsback and Zisserman, 2008), StanfordCars (Krause *et al.*, 2013), SUN397 (Xiao *et al.*, 2010), and UCF101 (Soomro, 2012). For evaluating domain generalization and robustness to distribution shifts, several ImageNet variants were also used: ImageNetV2 (Recht *et al.*, 2019), ImageNet-Sketch (Wang *et al.*, 2019), ImageNet-A

(Hendrycks *et al.*, 2021b), and ImageNet-R (Hendrycks *et al.*, 2021a). Additionally, MS-COCO (Lin *et al.*, 2014) and Flickr30K (Plummer *et al.*, 2015) were utilized for image-to-text and text-to-image retrieval tasks, RefCOCO+ (Yu *et al.*, 2016) for visual grounding, and SNLI-VE (Xie *et al.*, 2019) for visual entailment.

5. Vision-language model safety

Large VLMs extend LLMs by adding a visual modality through pre-trained image encoders and alignment modules, enabling applications like visual conversation and complex reasoning. However, this multi-modal design introduces unique vulnerabilities. This section reviews *adversarial attacks*, *latency energy attacks*, *jailbreak attacks*, *prompt injection attacks*, *backdoor & poisoning attacks*, and *defenses* developed for VLMs. Many VLMs use VLP-trained encoders, so the attacks and defenses discussed in Section 4 also apply to VLMs. The additional alignment process between the VLM pre-trained encoders and LLMs, however, expands the attack surface, with new risks like cross-modal backdoor attacks and jailbreaks targeting both text and image inputs. This underscores the need for safety measures tailored to VLMs.

5.1 Adversarial attacks

Adversarial attacks on VLMs primarily target the visual modality, which, unlike text, is more susceptible to adversarial perturbations due to its high-dimensional nature. By adding imperceptible changes to images, attackers aim to disrupt tasks like image captioning and visual question answering. These attacks are classified into *white-box* and *black-box* categories based on the threat model.

5.1.1 White-box attacks. White-box adversarial attacks on VLMs have full access to the model parameters, including both vision encoders and LLMs. These attacks can be classified into three types based on their objectives: *task-specific attacks*, *cross-prompt attack*, and *chain-of-thought (CoT) attack*.

Task-specific Attacks Schlarmann and Hein (2023) were the first to highlight the vulnerability of VLMs like Flamingo (Alayrac *et al.*, 2022) and GPT-4 (Achiam *et al.*, 2023) to adversarial images that manipulate caption outputs. Their study showed how attackers can exploit these vulnerabilities to mislead users, redirecting them to harmful websites or spreading misinformation. Gao *et al.* (2024b) introduced attack paradigms targeting the referring expression comprehension task, while Cui *et al.* (2024b) proposed a query decomposition method and demonstrated how contextual prompts can enhance VLM robustness against visual attacks.

Cross-prompt Attack refer to adversarial attacks that remain effective across different prompts. For example, *CroPA* (Luo *et al.*, 2024a) explored the transferability of a single adversarial image across multiple prompts, investigating whether it could mislead predictions in various contexts. To tackle this, they proposed refining adversarial perturbations through learnable prompts to enhance transferability.

CoT Attack targets the CoT reasoning process of VLMs. *Stop-reasoning Attack* (Wang *et al.*, 2024r) explored the impact of CoT reasoning on adversarial robustness. Despite observing some improvements in robustness, they introduced a novel attack designed to bypass these defenses and interfere with the reasoning process within VLMs.

5.1.2 Gray-box attacks. Gray-box adversarial attacks typically involve access to either the vision encoders or the LLM of a VLM, with a focus on vision encoders as the key differentiator between VLMs and LLMs. Attackers craft adversarial images that closely resemble target images, manipulating model predictions without full access to the VLM. For instance, *InstructTA* (Wang *et al.*, 2023b) generates a target image and uses a surrogate model to create adversarial perturbations, minimizing the feature distance between the original and adversarial image. To improve transferability, the attack incorporates GPT-4 paraphrasing to refine instructions.

5.1.3 Black-box attacks. In contrast, black-box attacks do not require access to the target model's internal parameters and typically rely on *transfer-based* or *generator-based* methods.

Transfer-based Attacks exploit the widespread use of frozen CLIP vision encoders in many VLMs. *AttackBard* (Dong *et al.*, 2023) demonstrates that adversarial images generated from surrogate models can successfully mislead Google's Bard, despite its defense mechanisms. Similarly, *AttackVLM* (Zhao *et al.*, 2024f) crafts targeted adversarial images for models like CLIP (Radford *et al.*, 2021) and BLIP (Li *et al.*, 2023a), successfully transferring these adversarial inputs to other VLMs. It also shows that black-box queries further improved the success rate of generating targeted

responses, illustrating the potency of cross-model transferability. *DynVLA* (Gu *et al.*, 2025) proposes a transfer-based black-box attack that perturbs the vision-language alignment mechanism within the vision-language connector. By injecting Gaussian-kernel-based attention shifts during optimization, it improves the transferability of adversarial examples across diverse VLMs, outperforming traditional input-level augmentation methods.

Generator-based Attacks leverage generative models to create adversarial examples with improved transferability. *AdvDiffVLM* (Guo *et al.*, 2024b) uses diffusion models to generate natural, targeted adversarial images with enhanced transferability. By combining adaptive ensemble gradient estimation and GradCAM-guided masking, it improves the semantic embedding of adversarial examples and spreads the targeted semantics more effectively across the image, leading to more robust attacks. *AnyAttack* (Zhang *et al.*, 2024j) presents a self-supervised framework for generating targeted adversarial images without label supervision. By utilizing contrastive loss, it efficiently creates adversarial examples that mislead models across diverse tasks. *CAVALRY-V* (Zhang *et al.*, 2025d) proposes a dual-objective generator framework for black-box attacks on video VLMs, achieving strong cross-model transferability and temporal coherence through large-scale pretraining and fine-tuning.

5.2 Jailbreak attacks

The inclusion of a visual modality in VLMs provides additional routes for jailbreak attacks. While adversarial attacks generally induce random or targeted errors, jailbreak attacks specifically target the model's safeguards to generate inappropriate outputs. Like adversarial attacks, jailbreak attacks on VLMs can be classified as *white-box* or *black-box* attacks.

5.2.1 White-box attacks. White-box jailbreak attacks leverage gradient information to perturb input images or text, targeting specific behaviors in VLMs. These attacks can be further categorized into three types: *target-specific jailbreak*, *universal jailbreak*, and *hybrid jailbreak*, each exploiting different aspects of the model's safety measures.

Target-specific Jailbreak focuses on inducing a specific type of harmful output from the model. *Image Hijack* (Bailey *et al.*, 2023) introduces adversarial images that manipulate VLM outputs, such as leaking information, bypassing safety measures, and generating false statements. These attacks, trained on generic datasets, effectively force models to produce harmful outputs. Similarly, *Adversarial Alignment Attack*

(Carlini *et al.*, 2024b) demonstrates that adversarial images can induce misaligned behaviors in VLMs, suggesting that similar techniques could be adapted for text-only models using advanced NLP methods.

Universal Jailbreak bypasses model safeguards, causing it to generate harmful content beyond the adversarial input. *VAJM* (Qi *et al.*, 2024a) shows that a single adversarial image can universally bypass VLM safety, forcing universal harmful outputs. *ImgJP* (Niu *et al.*, 2024) uses a maximum likelihood algorithm to create transferable adversarial images that jailbreak various VLMs, even bridging VLM and LLM attacks by converting images to text prompts. *UMK* (Wang *et al.*, 2024i) proposes a dual optimization attack targeting both text and image modalities, embedding toxic semantics in images and text to maximize impact. *HADES* (Li *et al.*, 2024u) introduces a hybrid jailbreak method that combines universal adversarial images with crafted inputs to bypass safety mechanisms, effectively amplifying harmful instructions and enabling robust adversarial manipulation.

5.2.2 Black-box attacks. Black-box jailbreak attacks do not require direct access to the internal parameters of the target VLM. Instead, they exploit external vulnerabilities, such as those in the frozen CLIP vision encoder, interactions between vision and language modalities, or system prompt leakage. These attacks can be classified into four main categories: *transfer-based attacks*, *manually-designed attacks*, *system prompt leakage*, and *red teaming*, each employing distinct strategies to bypass VLM defenses and trigger harmful behaviors.

Transfer-based Attacks on VLMs typically assume the attacker has access to the image encoder (or its open-source version), which is used to generate adversarial images that can then be transferred to attack the black-box LLM. For example, *Jailbreak in Pieces* (Shayegani *et al.*, 2023) introduces cross-modality attacks that transfer adversarial images, crafted using the image encoder (assume the model employed an open-source encoder), along with clean textual prompts to break VLM alignment.

Manually-designed Attacks can be as effective as optimized ones. For instance, *FigStep* (Gong *et al.*, 2025) introduces an algorithm that bypasses safety measures by converting harmful text into images via typography, enabling VLMs to visually interpret the harmful intent. *VRP* (Ma *et al.*, 2024b) adopts a visual role-play approach, using LLM-generated images of high-risk characters based on detailed descriptions. By pairing these images with benign role-play instructions, VRP exploits the negative traits of the characters to deceive VLMs into generating harmful outputs. *HIMRD* (Teng *et al.*, 2024) introduces a heuristic-induced multimodal risk distribution

framework that decomposes harmful prompts into semantically benign text and image components. A two-stage heuristic search then guides the model to reconstruct and affirm the underlying malicious intent.

System Prompt Leakage is another significant black-box jailbreak method, exemplified by *SASP* (Wu *et al.*, 2023). By exploiting a system prompt leakage in GPT-4V, SASP allowed the model to perform a self-adversarial attack, demonstrating the risks of internal prompt exposure.

Red Teaming recently saw an advancement with IDEATOR (Wang *et al.*, 2025c), which integrated a VLM with an advanced diffusion model to autonomously generate malicious image-text pairs. This approach overcomes the limitations of manually designed attacks, providing a scalable and efficient method for creating adversarial inputs without direct access to the target model.

5.3 Jailbreak defenses

This section reviews defense methods for VLMs against jailbreak attacks, categorized into *jailbreak detection* and *jailbreak prevention*. Detection methods identify harmful inputs or outputs for rejection or purification, while prevention methods enhance the model's inherent robustness to jailbreak queries through safety alignment or filters.

5.3.1 Jailbreak detection. JailGuard (Zhang *et al.*, 2023f) detects jailbreak attacks by mutating untrusted inputs and analyzing discrepancies in model responses. It uses 18 mutators for text and image inputs, improving generalization across attack types. *GuardMM* (Sharma *et al.*, 2024a) is a two-stage defense: the first stage validates inputs to detect unsafe content, while the second stage focuses on prompt injection detection to protect against image-based attacks. It uses a specialized language to enforce safety rules and standards. *MLLM-Protector* (Pi *et al.*, 2024) identifies harmful responses using a lightweight detector and detoxifies them through a specialized transformation mechanism. Its modular design enables easy integration into existing VLMs, enhancing safety and preventing harmful content generation.

5.3.2 Jailbreak prevention. AdaShield (Wang *et al.*, 2024n) defends against structure-based jailbreaks by prepending defense prompts to inputs, refining them adaptively through collaboration between the VLM and an LLM-based prompt generator, without requiring fine-tuning. *ECSO* (Gou *et al.*, 2024) offers a training-free protection by converting unsafe images into text descriptions, activating the safety alignment of pre-trained LLMs within VLMs to ensure safer outputs. *InferAligner* (Wang *et al.*, 2024h) applies cross-model guidance during inference, adjusting

activations using safety vectors to generate safe and reliable outputs. *BlueSuffix* (Zhao *et al.*, 2025) introduces a reinforcement learning-based black-box defense framework consisting of three key components: (1) an image purifier for securing visual inputs, (2) a text purifier for safeguarding textual inputs, and (3) a reinforcement fine-tuning-based suffix generator that leverages bimodal gradients to enhance cross-modal robustness. *DPS* (Zhou *et al.*, 2025d) introduces a black-box, training-free defense that supervises VLMs using responses from partially cropped images, boosting robustness to visual jailbreaks while maintaining benign performance. *ETA* (Ding *et al.*, 2025a) presents a two-stage inference-time alignment framework: it first screens the safety of inputs and outputs, then applies alignment through interference prefixes and best-of-N sentence search, enabling safe and helpful responses without extra training.

5.4 Energy latency attacks.

Similar to LLMs, multi-modal LLMs also face significant computational demands. Verbose images (Gao *et al.*, 2024c) exploit these demands by overwhelming service resources, resulting in higher server costs, increased latency, and inefficient GPU usage. These images are specifically designed to delay the occurrence of the EOS token, increasing the number of auto-regressive decoder calls, which in turn raises both energy consumption and latency costs.

5.5 Prompt injection attacks

Prompt injection attacks against VLMs share the same objective as those against LLMs (Section 3), but the visual modality introduces continuous features that are more easily exploited through adversarial attacks or direct injection. These attacks can be further classified into *optimization-based attacks* and *typography-based attacks*.

Optimization-based Attacks often optimize the input images using (white-box) gradients to produce stronger attacks. These attacks manipulate the model's responses, influencing future interactions. One representative method is *Adversarial Prompt Injection* (Bagdasaryan *et al.*, 2023a), where attackers embed malicious instructions into VLMs by adding adversarial perturbations to images.

Typography-based Attacks exploit VLMs' typographic vulnerabilities by embedding deceptive text into images without requiring gradient access (i.e., black-box). The *Typographic Attack* (Qraitem *et al.*, 2024) introduces two variations: *Class-Based Attack* to misidentify classes and *Descriptive Attack* to generate misleading

labels. These attacks can also leak personal information (Chen *et al.*, 2023d), highlighting significant security risks.

5.6 Backdoor and poisoning attacks

Most VLMs rely on VLP encoders, with safety threats discussed in Section 4. This section focuses on backdoor and poisoning risks arising during fine-tuning and testing, specifically when aligning vision encoders with LLMs. Backdoor attacks embed triggers in visual or textual inputs to elicit specific outputs, while poisoning attacks inject malicious image-text pairs to degrade model performance. We review backdoor and poisoning attacks separately, though most of these works are backdoor attacks.

5.6.1 Backdoor attacks. We further classify backdoor attacks on VLMs into *tuning-time backdoor* and *testing-time backdoor*.

Tuning-time Backdoor injects the backdoor during VLM instruction tuning. *MABA* (Liang *et al.*, 2024a) targets domain shifts by adding domain-agnostic triggers using attributional interpretation, enhancing attack robustness across mismatched domains in image captioning tasks. *BadVLMDriver* (Ni *et al.*, 2024) introduced a physical backdoor for autonomous driving, using objects like red balloons to trigger unsafe actions such as sudden acceleration, bypassing digital defenses and posing real-world risks. Its automated pipeline generates backdoor training samples with malicious behaviors for stealthy, flexible attacks. *ImgTrojan* (Tao *et al.*, 2024) introduces a jailbreaking attack by poisoning image-text pairs in training data, replacing captions with malicious prompts to enable VLM jailbreaks, exposing risks of compromised datasets.

Test-time Backdoor leverages the similarity of universal adversarial perturbations and backdoor triggers to inject backdoor at test-time. AnyDoor (Lu *et al.*, 2024a) embeds triggers in the textual modality via adversarial test images with universal perturbations, creating a text backdoor from image-perturbation combinations. It can also be seen as a multi-modal universal adversarial attack. Unlike traditional methods, AnyDoor does not require access to training data, enabling attackers to separate setup and activation of the attack.

5.6.2 Poisoning attacks. Shadowcast (Xu *et al.*, 2024f) is a stealthy tuning-time backdoor attack on VLMs. It injects poisoned samples visually indistinguishable from benign ones, targeting two objectives: 1) *Label Attack*, which misclassifies objects, and 2) *Persuasion Attack*, which generates misleading narratives. With only 50

poisoned samples, Shadowcast achieves high effectiveness, showing robustness and transferability across VLMs in black-box settings.

5.7 Datasets and benchmarks

The datasets used in VLM safety research are detailed in Table 6. Below, we review the benchmarks proposed for evaluating VLM safety and robustness, summarized in Table 7. *SafeSight* (Tu *et al.*, 2024) introduces two VQA datasets, *OODCV-VQA* and *Sketchy-VQA*, to evaluate out-of-distribution (OOD) robustness, highlighting VLMs' vulnerabilities to OOD texts and vision encoder weaknesses. *MM-SafetyBench* (Liu *et al.*, 2024k) focuses on image-based manipulations, revealing vulnerabilities in multi-modal interactions. *AVIBench* (Zhang *et al.*, 2024g) evaluates VLM robustness against 260K adversarial visual instructions, exposing susceptibility to image-based, text-based, and content-biased adversarial visual instructions (AVIs). *Jailbreak Evaluation of GPT-4o* (Ying *et al.*, 2024) tests GPT-4o with multi-modal and unimodal jailbreak attacks, uncovering alignment vulnerabilities. *JailBreakV-28K* (Luo *et al.*, 2024c) assesses the transferability of LLM jailbreak techniques to VLMs, showing high attack success rates across 10 open-source models. These studies collectively reveal significant vulnerabilities in VLMs to OOD inputs, adversarial instructions, and multi-modal jailbreaks. MIS (Ding *et al.*, 2025b) presents the first multi-image safety dataset for assessing VLMs' visual reasoning in complex unsafe scenarios, exposing safety reasoning gaps in current fine-tuning methods and providing nuanced multi-image benchmarks. VLJailbreakBench (Wang *et al.*, 2025c) offers 3,654 adversarial image-text pairs generated by the IDEATOR red-teaming framework to evaluate VLMs under black-box multimodal jailbreak settings. Argus Inspection (Yao *et al.*, 2025) introduces a detail-oriented benchmark for commonsense safety, embedding causally critical yet textually implicit visual "traps" within realistic scenarios.

Table 6. A summary of attacks and defenses for VLMs

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
Adversarial Attack	Caption Attack (Schlarmann and Hein, 2023)	2023	White-box	Task-specific +V	OpenFlamingo	MS-COCO/Flickr30k/ OK-VQA/VizWiz
	VisBreaker (Cui *et al.*, 2024b)	2023	White-box	Task-specific +V	LLaVA/BLIP-2/ InstructBLIP	MS-COCO/VQA V2/ ScienceQA-Image/ TextVQA/ POPE/ MME
	CroPA (Luo *et al.*, 2024a)	2024	White-box	Cross-prompt +VL	OpenFlamingo/ BLIP-2/ Instruct BLIP	MS-COCO/VQA-v2
	GroundBreaker (Gao *et al.*, 2024b)	2024	White-box	Task-specific +V	MiniGPT-v2	RefCOCO/RefCOCO +/ RefCOCOg
	Stop-reasoning Attack (Wang *et al.*, 2024r)	2024	White-box	CoT attack+V	MiniGPT-4/ OpenFlamingo/ LLaVA	ScienceQA/A-OKVQA
	InstructTA (Wang *et al.*, 2023b)	2023	Gray-box	Encoder attack +V	BLIP-2/ Instruct BLIP/ MiniGPT-4/LLaVA/ CogVLM	ImageNet-1K/LLaVA-Instruct-150K/MS-COCO
	Attack Bard (Dong *et al.*, 2023)	2023	Black-box	Transfer-based +V	Bard/GPT-4V/Bing Chat/ ERNIE Bot	NeurIPS'17 adversarial competition dataset

(*continued*)

Table 6. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	AttackVLM (Zhao *et al.*, 2024f)	2024	Black-box	Transfer-based +V	BLIP/UniDiffuser/ Img2Prompt/BLIP-2/ LLaVA/MiniGPT-4	ImageNet-1K/MS-COCO
	DynVLA (Gu *et al.*, 2025)	2025	Black-box	Transfer-based +VL	InstructBLIP/MiniGPT4/ LLaVA/Gemini	MS-COCO/VQA-v2
	AdvDiffVLM (Guo *et al.*, 2024b)	2024	Black-box	Generator-based+V	MiniGPT-4/LLaVA/ UniDiffuser/MiniGPT-4/ BLIP/BLIP-2/Img2LLM	NeurIPS'17 adversarial competition dataset/ MS-COCO
	AnyAttack (Zhang *et al.*, 2024j)	2024	Black-box	Generator-based+V	CLIP/BLIP/BLIP2/ InstructBLIP/ MiniGPT-4	MSCOCO/Flickr30K/ SNLI-VE
	CAVALRY-V (Zhang *et al.*, 2025d)	2025	Black-box	Generator-based+V	GPT-4.1/Gemini/ QwenVL/InternVL/ LLaVA/Aria/ MiniCPM	MMBench-Video/ Video-MME
Latency-Energy Attack	Verbose Images (Gao *et al.*, 2024c)	2024	White-box	Task-specific +V	BLIP/BLIP2/ InstructBLIP/ MiniGPT-4	MS-COCO/ImageNet
Jailbreak Attack	Image Hijack (Bailey *et al.*, 2023)	2023	White-box	Target-specific +V	LLaVA	Alpaca training set/ AdvBench
	Adversarial Alignment Attack (Carlini *et al.*, 2024b)	2024	White-box	Target-specific +V	MiniGPT-4/LLaVA/ LLaMA Adapter	toxic phrase dataset

(*continued*)

Table 6. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	VAJM (Qi *et al.*, 2024a)	2024	White-box	Universal attack+V	MiniGPT-4/LLaVA/ InstructBLIP	VAJM training set/ VAJM test set/ Real Toxicity Prompts
	imgJP (Niu *et al.*, 2024)	2024	White-box	Universal attack+V	MiniGPT-4/Mini GPT-v2/LLaVA/ InstructBLIP/mPLUG-Owl2	AdvBench-M
	UMK (Wang *et al.*, 2024i)	2024	White-box	Universal attack+VL	MiniGPT-4	AdvBench/VAJM training set/VAJM test set/ RealToxicity Prompts
	HADES (Li *et al.*, 2024u)	2024	White-box	Hybrid method +V	LLaVA/GPT-4V/Gemini-Pro-Vision	HADES dataset
	Jailbreak in Pieces (Shayegani *et al.*, 2023)	2023	Black-box	Transfer-based +V	LlaVA /LLaMA-Adapter V2	Jailbreak in Pieces dataset
	Figstep (Gong *et al.*, 2025)	2023	Black-box	Manual pipeline+V	LLaVA-v1.5/MiniGPT-4/ CogVLM/GPT-4V	SafeBench
	SASP (Wu *et al.*, 2023)	2023	Black-box	Prompt leakage +L	LLaVA/GPT-4V	Celebrity face image dataset/CelebA/LFWA
	VRP (Ma *et al.*, 2024b)	2024	Black-box	Manual pipeline+V	LLaVA/Qwen-VL-Chat/ OmniLMM /InternVL Chat-V1.5/Gemini-Pro-Vision	RedTeam-2k/ HarmBench

(*continued*)

Table 6. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	HIMRD (Teng *et al.*, 2024)	2024	Black-box	Manual pipeline+VL	LLaVA/DeepSeek/ GPT-4o/Gemini/ Qwen-VL	SafeBench/tiny-SafeBench
	IDEATOR (Wang *et al.*, 2025c)	2025	Black-box	Red teaming +VL	LLaVA/Instruct BLIP/Mini GPT-4	AdvBench/VAJM test set
Prompt Injection Attack	Adversarial Prompt Injection (Bagdasaryan *et al.*, 2023a)	2023	White-box	Optimization-based+V	LLaVA/PandaGPT	Self-collected dataset
	Typographic Attack (Qraitem *et al.*, 2024)	2024	Black-box	Typography-based+V	LLaVA/MiniGPT4/ InstructBLIP/ GPT-4V	OxfordPets / StanfordCars / Flowers / Aircraft / Food101
Backdoor & Poisoning Attack	Shadowcast (Xu *et al.*, 2024f)	2024	Poisoning	Tuning-stage +VL	LLaVA/MiniGPT-v2/ InstructBLIP	cc-sbu-align dataset
	Instruction-Tuned Backdoor (Liang *et al.*, 2024a)	2024	Backdoor	Tuning-stage +VL	OpenFlamingo/BLIP-2/ LLaVA	MIMIC-IT/COCO/ Flickr30K
	Anydoor (Lu *et al.*, 2024a)	2024	Backdoor	Testing-stage +VL	LLaVA/MiniGPT-4/ InstructBLIP/ BLIP-2	VQAv2/SVIT/DALL-E dataset

(*continued*)

Table 6. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	BadVLM Driver (Ni *et al.*, 2024)	2024	Backdoor	Tuning-stage +V	LLaVA/MiniGPT-4	nuScenes dataset
	ImgTrojan (Tao *et al.*, 2024)	2024	Backdoor	Tuning-stage +VL	LLaVA	LAION
Jailbreak Defenses	JailGuard (Zhang *et al.*, 2023f)	2023	Detection	Detection+VL	GPT-3.5/MiniGPT-4	Self-collected dataset
	GuardMM (Sharma *et al.*, 2024a)	2024	Detection	Detection+V	GPT-4V/LLAVA/ MINIGPT-4	Self-collected dataset
	AdaShield (Wang *et al.*, 2024p)	2024	Prevention	Prevention+V	LLaVA/CogVLM/ MiniGPT-v2	Figstep/QR
	MLLM-Protector (Pi *et al.*, 2024)	2024	Prevention	D+P +V	Open-LLaMA/LLaMA/ LLaVA	Safe-Harm-10K
	ECSO (Gou *et al.*, 2024)	2024	Prevention	Prevention+V	LLaVA/Share GPT4V/mPLUG-OWL2/ Qwen-VL-Chat/InternLM-XComposer	MM-SafetyBench/ VLSafe/ VLGuard
	InferAligner (Wang *et al.*, 2024h)	2024	Prevention	Prevention +VL	LLaMA2/LLaVA	AdvBench/ TruthfulQA/ MM-Harmful Bench

(*continued*)

Table 6. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Datasets
	BlueSuffix (Zhao *et al.*, 2025)	2024	Prevention	Prevention +VL	LLaVA/MiniGPT-4/ Gemini	MM-SafetyBench/ RedTeam-2k
	DPS (Zhou *et al.*, 2025d)	2025	Prevention	Prevention +V	Qwen-VL-Plus/ GPT-4o/Gemini-1.5-Flash	RTA-100/MultiTrust/ Self-Gen/MM-SafetyBench/HADES/ VisualAttack
	ETA (Ding *et al.*, 2025a)	2025	Prevention	Prevention+VL	LLaVA/InternVL/ InternLM-XComposer/ LLaMA3.2-Vision	SPA-VL/MM-SafetyBench/FigStep

Table 7. Safety and robustness benchmarks for VLMs

Benchmarks	Year	Size	# VLMs evaluated
OODCV-VQA (Tu *et al.*, 2024)	2023	4,244	21
Sketchy-VQA (Tu *et al.*, 2024)	2023	4,000	21
MM-SafetyBench (Liu *et al.*, 2024k)	2023	5,040	12
AVIBench (Zhang *et al.*, 2024g)	2024	260,000	14
Jailbreak Evaluation of GPT-4o (Ying *et al.*, 2024)	2024	4,180	1
JailBreakV-28K (Luo *et al.*, 2024c)	2024	28,000	10
MIS (Ding *et al.*, 2025b)	2025	6,185	14
VLJailbreakBench (Wang *et al.*, 2025c)	2025	3,654	11
Argus Inspection (Yao *et al.*, 2025)	2025	1,430	26

6. Diffusion model safety

This section focuses on safety research related to diffusion models (Rombach *et al.*, 2022; Ramesh *et al.*, 2022; Betker *et al.*, 2023; Saharia *et al.*, 2022), which involve forward noise addition and reverse sampling. In the forward process, Gaussian noise is incrementally added to an image until it becomes pure noise. Reverse sampling generates new samples by stepwise denoising based on learned data distributions (Ho *et al.*, 2020; Song *et al.*, 2021a, b). By integrating input information, diffusion models perform conditional generation, transforming data distribution modeling $p(x)$ into $p(x|\text{guidance})$.

Widely used in Image-to-Image (I2I), Text-to-Image (T2I), and Text-to-Video (T2V) tasks, diffusion models are applied in content creation, image editing, and film production. However, their extensive use exposes them to various security risks including *adversarial*, *jailbreak*, *backdoor*, and *privacy attacks*. These attacks can degrade generation quality, bypass safety filters, manipulate outputs, and reveal sensitive training data. This section also reviews defenses against these threats, including *jailbreak* and *backdoor defenses*, as well as *intellectual property protection* techniques.

6.1 Adversarial attacks

Adversarial attacks on diffusion models typically perturb text prompts to degrade image quality or cause semantic mismatches with the original text. This section reviews existing adversarial attacks, categorized by threat model into *white-box*, *gray-box*, and *black-box* methods.

6.1.1 White-box attacks. White-box attacks on T2I diffusion models assume full access to model parameters, allowing direct optimization of text prompts or latent

space to degrade or disrupt image generation. For example, *SAGE* (Liu *et al.*, 2024h) explores both the discrete prompt and latent spaces to uncover failure modes in T2I models, including distorted generations and targeted manipulations. *ATM* (Du *et al.*, 2024a) generates attack prompts similar to clean prompts by replacing or extending words using Gumbel Softmax, preventing the model from generating desired subjects. *FOOLSDEDIT* (Zhou *et al.*, 2024c) imperceptibly modifies stroke images by applying a mix of four operations: exposure, motion blur, identity mapping, and an empty operation. It automatically selects the optimal combination to steer SDEdit's (Meng *et al.*, 2021) outputs toward a desired attribute, while ensuring that the strokes appear visually unchanged.

6.1.2 Gray-box attacks. Gray-box attacks assume the CLIP text encoder used in many T2I diffusion models is frozen and publicly available. The attacker can then exploit CLIP similarity loss to craft adversarial text prompts targeting the text encoder.

QFA (Zhuang *et al.*, 2023) minimizes cosine similarity between original and perturbed text embeddings to generate images that differ as much as possible from the original text. *RVTA* (Zhang *et al.*, 2024b) maximizes image-text similarity to align adversarial prompts with reference images generated by a surrogate diffusion model. *MMP-Attack* (Yang *et al.*, 2024a) simultaneously maximizes the cosine similarity between the perturbed text embedding and the target embedding in both the text and image modalities, while employing a straight-through estimator to execute the optimization process. *DORMANT* (Zhou *et al.*, 2024b) embeds imperceptible PGD noise, optimized with VAE-latent, CLIP-semantic, ReferenceNet-detail, and frame-consistency losses. This causes pose-driven portrait animation models to generate identity-shifted and jittery videos, while the source photo remains visually unchanged.

6.1.3 Black-box attacks. Black-box attacks assume the attacker has no knowledge of the victim diffusion model's internals (parameters or architecture). Since diffusion models use text prompts as input, existing attacks employ textual adversarial techniques to evade the model. These attacks can be further categorized by granularity into *character-level*, *word-level*, and *sentence-level* attacks.

Character-level Attacks modify the characters in the text input to create adversarial prompts. *ECB* (Struppek *et al.*, 2023a) shows how replacing characters with homoglyphs, such as using Hangul or Arabic scripts, shifts generated images toward cultural stereotypes. Subsequent works, like *CharGrad* (Kou *et al.*, 2023), optimize character-level perturbations using gradient-based attacks and proxy representations to map character changes to embedding shifts. *ER* (Gao *et al.*, 2023) uses distribution-based objectives (e.g., MMD, KL divergence) to maximize discrepancies in image

distributions, enhancing attack effectiveness. These attacks exploit typos, homoglyphs, and phonetic modifications, disrupting text-to-image outputs.

Word-level Attacks craft adversarial prompts by replacing or adding words to the input text. *DHV* (Daras and Dimakis, 2022) uncovers a hidden vocabulary in diffusion models, where nonsensical strings like `Apoploe vesrreaitais` can generate bird images, due to their proximity to target concepts in the CLIP text embedding space. Building on this, *AA* (Millière, 2022) introduces macaronic prompting, combining word fragments from different languages to control visual outputs systematically. These attacks reveal vulnerabilities in the relationship between text embeddings and image generation.

Sentence-level Attacks rewrite a substantial part or the entire prompt to create adversarial prompts. *RIATIG* (Liu *et al.*, 2023b) uses a CLIP-based image similarity measure as an optimization objective and a genetic algorithm to iteratively mutate and select text prompts, creating adversarial examples that resemble the target image while remaining semantically different from the original text. In contrast, *BBA* (Maus *et al.*, 2023) employs classification loss and black-box optimization to refine prompts, using Token Space Projection (TPS) to bridge the gap between continuous word embeddings and discrete tokens, enabling the generation of category-specific images without explicit category terms.

6.2 Jailbreak attacks

Diffusion models use both internal and external safety mechanisms to void the generation of Not Safe For Work (NSFW) content. Internal safety mechanisms often refer to the inherent robustness of T2I diffusion models, achieved through safety alignment during training, which aims to reduce the likelihood of generating harmful content. External safety mechanisms, on the other hand, are safety filters, such as text, image, or text-image classifiers, applied to detect and block unsafe outputs after generation. Jailbreak attacks aim to craft adversarial prompts that bypass the safety mechanisms of diffusion models, enabling the generation of harmful content. This section provides a systematic review of existing jailbreak methods, categorized by threat model into *white-box*, *gray-box*, and *black-box* attacks.

6.2.1 White-box attacks. White-box attacks can bypass the safety mechanisms in T2I diffusion models through gradient-based optimization. These attacks can be further classified into *internal safety attacks* and *external safety attacks*, each exploiting specific vulnerabilities in the victim models.

Internal Safety Attacks target the internal safety mechanisms of diffusion models. Jailbreaking internally safety-enhanced diffusion models involves regenerating NSFW content by bypassing the removal of harmful concepts. The red teaming tool *P4D* (Chin *et al.*, 2024) automatically identifies problematic prompts to exploit limitations in current safety evaluations, aligning the predicted noise of an unconstrained model with that of a safety-enhanced one. *UnlearnDiffAtk* (Zhang *et al.*, 2023g) introduces an evaluation framework that uses unlearned diffusion models' classification capabilities to optimize adversarial prompts, aligning predicted noise with a target unsafe image to force the model to recreate NSFW content during denoising.

External Safety Attacks target the safety filters of diffusion models, aiming to bypass both input and output safety mechanisms. *RTSDSF* (Rando *et al.*, 2022) reverse-engineered predefined NSFW concepts in filters by using the CLIP model to encode and compare NSFW vocabulary embeddings, performing a dictionary attack. It also showed that prompt dilution—adding irrelevant details—can bypass safety filters. *MMA* (Yang *et al.*, 2024i) employs a similarity-driven loss to optimize adversarial prompts and introduce subtle perturbations to input images, bypassing both prompt filters and post-hoc safety checkers during image editing.

6.2.2 Gray-box attacks. Gray-box jailbreak attacks assume that attackers have full access only to the open-source text encoder, with other components of the diffusion model remaining inaccessible. In this scenario, the attacker exploits the exposed text encoder to bypass the model's internal safety mechanism.

Internal Safety Attacks, under the gray-box setting, target models with 'concept erasure'. *Ring-A-Bell* (Tsai *et al.*, 2024) extracts unsafe concepts by comparing antonymous prompt pairs, generates harmful prompts with soft prompts, and refines them using a genetic algorithm. *JPA* (Ma *et al.*, 2024a) leverages antonyms like "`nude`" and "`clothed`", calculating their average difference in the text embedding space to represent NSFW concepts, then optimizes prefix prompts for semantic alignment. *RT-Attack* (Gao *et al.*, 2024e) uses a two-stage strategy to maximize textual similarity to NSFW prompts and iteratively refines them based on image-level similarity, demonstrating that even limited knowledge can enable attacks on safety-enhanced models.

6.2.3 Black-box attacks. Black-box jailbreaks on diffusion models target commercial models with access only to outputs, such as filter rejections or generated image quality and semantics, and are primarily *external safety attacks*.

External Safety Attacks, in the black-box setting, use hand-crafted or LLM-assisted adversarial prompts to mislead the victim model to generate NSFW content. *UD* (Qu *et al.*, 2023) highlights the risk of T2I models generating unsafe content, especially hateful memes, by refining unsafe prompts manually. *SneakyPrompt* (Yang *et al.*, 2024j) uses reinforcement learning to optimize adversarial prompts, which updates its policy network based on filter evasion and semantic alignment. Other methods employ LLMs to refine adversarial prompts. *Groot* (Liu *et al.*, 2024q) decomposes prompts into objects and attributes to dilute sensitive content. *DACA* (Deng and Chen, 2023) breaks down and recombines prompts using LLMs. *SurrogatePrompt* (Ba *et al.*, 2024) targets Midjourney, substituting sensitive terms and leveraging image-to-text modules to generate harmful content at scale. *Atlas* (Dong *et al.*, 2024) automates the attack with a two-agent system: one VLM generates adversarial prompts, while an LLM evaluates and selects the best candidates. These LLM-assisted strategies can significantly improve the effectiveness and stealthiness of the attacks. *PGJ* (Huang *et al.*, 2025d) identifies unsafe tokens and replaces them with perceptually similar but semantically distant phrases, producing short, natural prompts that evade text filters without directly querying the T2I model. *R2A* (Zhang *et al.*, 2025b) further improves an LLM's reasoning for jailbreaking T2I models by first fine-tuning on Chain-of-Thought examples based on contextual word meanings, and then applying reinforcement learning guided by a dense attack process reward.

6.3 Jailbreak defenses

This section reviews existing defense strategies proposed for T2I diffusion models against jailbreak attacks, including *concept erasure* and *inference guidance*. The key challenge of these defenses is how to ensure safety while maintaining generation quality.

6.3.1 Concept erasure. Concept erasure is an emerging research area focused on removing undesirable concepts (e.g., NSFW content and copyrighted styles) from diffusion models, where these concepts are referred to as *target concepts*. Concept erasure methods can be categorized into three types: *finetuning-based*, *close-form solution*, and *pruning-based*, depending on the strategy employed.

Finetuning-based Methods. These methods use gradient-based optimization to adjust model parameters, typically involving a loss function with an erasure term to prevent the generation of representations linked to the target (undesirable) concept,

and a constraint term to preserve non-target concepts. These approaches can be categorized into *anchor-based*, *anchor-free*, and *adversarial* erasure methods.

Anchor-based Erasing is a targeted approach that guides the model to shift the target (undesirable concept) towards a good concept (anchor) by aligning predicted latent noise. *AC* (Kumari *et al.*, 2023) defines anchor concepts as broader categories encompassing the target concepts (e.g., "`Grumpy Cat`" → "`Cat`") and uses standard diffusion loss on text-image pairs of anchors to preserve their integrity while erasing target concepts. *ABO* (Hong *et al.*, 2024a) removes specific target concepts by modifying classifier guidance, using both explicit (replacing the target with a predefined substitute) and implicit (suppressing attention maps) erasing signals, and includes a penalty term to maintain generation quality. *DoCo* (Wu *et al.*, 2024h) improves generalization by aligning target and anchor concepts through adversarial training and mitigating gradient conflicts with concept-preserving gradient surgery. *SPM* (Lyu *et al.*, 2024) uses a 1D adapter and negative guidance (Gandikota *et al.*, 2023) to suppress target concepts while ensuring non-target concepts remain consistent, affecting only relevant synonyms. *SA* (Heng and Soh, 2024) applies generative replay and elastic weight consolidation to stabilize model weights and maintain normal generation capabilities while preserving non-target concepts. *SafeGen* (Li *et al.*, 2024r) fine-tunes the vision-only self-attention of Stable Diffusion on < *nude*, *mosaic*, *benign* > triplets, encouraging nude features to be transformed into mosaics while preserving benign images.

Anchor-free Erasing is a non-targeted fine-tuning approach that reduces the probability of generating target concepts without aligning to a specific safe concept. *ESD* (Gandikota *et al.*, 2023) modifies classifier-free guidance into negative-guided noise prediction to minimize the target concept's generation probability (e.g., "Van Gogh"). *SDD* (Kim *et al.*, 2023) addresses the extra effects of ESD's negative guidance by using unconditioned predictions and EMA to avoid catastrophic forgetting. *DT* (Ni *et al.*, 2023) erases unsafe concepts by training the model to denoise scrambled low-frequency images. *Forget-Me-Not* (Zhang *et al.*, 2024e) uses Attention Resteering to minimize intermediate attention maps related to the target concept. *Geom-Erasing* (Liu *et al.*, 2024u) erases implicit concepts like watermarks by applying a geometric-driven control method and introduces the *Implicit Concept Dataset*. *SepME* (Zhao *et al.*, 2024a) advances multiple concept erasure and restoration. Fuchi and Takagi (2024) proposed few-shot unlearning by targeting the text encoder rather than the image encoder or diffusion model. *CCRT*

(Han *et al.*, 2024) proposes a method for continuous removal of diverse concepts from diffusion models. ”

Adversarial Erasing enhances previous methods by introducing perturbations to the target concept’s text embedding and using adversarial training to improve robustness. *Receler* (Huang *et al.*, 2024a) employs a lightweight eraser and adversarial prompt embeddings, iteratively training against each other, while applying a binary mask from U-Net attention maps to target only the concept regions. *AdvUnlearn* (Zhang *et al.*, 2024v) shifts adversarial attacks to the text encoder, targeting the embedding space and using regularization to preserve normal generation. *RACE* (Kim *et al.*, 2024a) improves efficiency by conducting adversarial attacks at a single timestep, reducing computational complexity. These methods enhance the model’s resistance to adversarial prompts aimed at regenerating erased concepts. *CPE* (Lee *et al.*, 2025a) introduces a Residual Attention Gate (ResAG) that activates exclusively on target-concept tokens to precisely erase them. The gate is further strengthened through adversarial embedding attack–defense iterations, providing robust protection.

Close-form Solution Methods. These methods offer an efficient alternative to fine-tuning-based erasure, focusing on localized updates in cross-attention layers to erase target concepts, inspired by model editing in LLMs (Meng *et al.*, 2023). Unlike fine-tuning, which aligns denoising predictions, these methods align cross-attention values. *TIME* (Orgad *et al.*, 2023) applies a closed-form solution to debias models, while *UCE* (Gandikota *et al.*, 2024) extends this to multiple erasure targets, preserving surrounding concepts to reduce interference. *MACE* (Lu *et al.*, 2024d) refines cross-attention updates with LoRA and Grounded-SAM (Kirillov *et al.*, 2023; Liu *et al.*, 2024j) for region-specific erasure. A recent challenge is that erased concepts can still be generated via sub-concepts or synonyms (Liu *et al.*, 2024s). *RealEra* (Liu *et al.*, 2024s) tackles this by mining associated concepts and adding perturbations to the embedding, expanding the erasure range with beyond-concept regularization. *RECE* (Gong *et al.*, 2024a) addresses insufficient erasure by continually finding new concept embeddings during fine-tuning and applying closed-form solutions for further erasure.

Pruning-based Methods. These methods erase target concepts by identifying and removing neurons strongly associated with the target, selectively disabling them without updating model weights. *ConceptPrune* calculates a Wanda score using target and reference prompts to measure each neuron’s contribution, pruning those most associated with the target concept. Similarly, another approach Yang *et al.* (2024e) identifies concept-correlated neurons using adversarial prompts to enhance the robustness of existing erasure methods.

6.3.2 Inference guidance. Inference guidance methods steer pre-trained diffusion models to generate safe images by incorporating additional auxiliary information and specific guidance during the inference process.

Input Guidance. This type of guidance use additional input text to steer the model toward safe content. *SLD* (Schramowski *et al.*, 2023) adjusts noise predictions during inference based on a text condition and unsafe concepts, guiding generation towards the intended prompt while avoiding unsafe content, without requiring fine-tuning. It also introduces the I2P benchmark, a dataset for testing inappropriate content generation. *PromptGuard* (Yuan *et al.*, 2025) learns a safety soft prompt within the text embedding space and appends it as a suffix to every user prompt, steering the T2I model away from NSFW outputs without altering its weights.

Input & Output Guidance. This type of methods prevent harmful inputs and control NSFW outputs. *Ethical-Lens* (Cai *et al.*, 2024) employs a plug-and-play framework, using an LLM for input text revision (Ethical Text Scrutiny) and a multi-headed CLIP classifier for output image modification (Ethical Image Scrutiny), ensuring alignment with societal values without retraining or internal changes.

Latent space Guidance. This approach uses additional implicit representations in the latent space to guide generation. *SDIDLD* (Li *et al.*, 2024d) employs self-supervised learning to identify the opposite latent direction of inappropriate concepts (e.g., "anti-sexual") and adds these vectors at the bottleneck layer, preventing harmful content generation. *Concept Corrector* (Meng *et al.*, 2025) functions during image generation by employing a Generation Check Mechanism (GCM) to inspect an intermediate prediction of the final image for unwanted concepts. If such concepts are detected, a Concept Removal Attention (CRA) module is then activated to dynamically replace the target features with those associated with a negative concept.

6.4 Backdoor attacks

Backdoor attacks on diffusion models allow adversaries to manipulate generated content by injecting backdoor triggers during training. These "malicious triggers" are embedded in model components, and during generation, inputs with triggers (e.g., prompts or initial noise) guide the model to produce predefined content. The key challenge is enhancing attack success rates while keeping the trigger covert and preserving the model's original utility. Existing attacks can be categorized into *training manipulation* and *data poisoning* methods.

6.4.1 Training manipulation. This type of attack typically assumes the attacker aims to release a backdoored diffusion model, granting control over the training or even inference processes. Existing attacks focus on the visual modality, inserting backdoors by using image pairs with triggers and target images (*image-image pair injection*), typically targeting unconditional diffusion models.

BadDiffusion (Chou *et al.*, 2023) presents the first backdoor attack on T2I diffusion models, which modifies the forward noise-addition and backward denoising processes to map backdoor target distributions to image triggers while maintaining DDPM sampling. *VillanDiffusion* (Sheng-Yen Chou *et al.*, 2024) extends this to conditional models, adding prompt-based triggers and textual triggers for tasks like text-to-image generation. *TrojDiff* (Chen *et al.*, 2023c) advances the research by controlling both training and inference, incorporating Trojan noise into sampling for diverse attack objectives. *IBA* (Li *et al.*, 2024n) introduces invisible trigger backdoors using bi-level optimization to create covert perturbations that evade detection. *DIFF2* (Li *et al.*, 2024c) proposes a backdoor attack in adversarial purification, optimizing triggers to mislead classifiers and extending it to data poisoning by injecting backdoors directly.

6.4.2 Data poisoning. Unlike training manipulation, data poisoning methods do not directly interfere with the training process, restricting the attack to inserting poisoned samples into the dataset. These attacks typically target conditional diffusion models and explore two types of textual triggers: *text-text pair* and *text-image pair*.

Text-text Pair Triggers consist of triggered prompts and their corresponding target prompts. *RA* (Struppek *et al.*, 2023b) adopts this approach to inject backdoors into the text encoder by adding a covert trigger character, mapping the original to the target prompt while preserving encoder functionality through utility loss optimization. The backdoored encoder generates embeddings with predefined semantics, guiding the diffusion model's output. This lightweight attack requires no interaction with other model components. Several studies (Struppek *et al.*, 2023b; Vice *et al.*, 2024; Huang *et al.*, 2023, 2024h) have also explored this approach.

Text-image Pair Triggers consist of triggered prompts paired with target images. *BadT2I* (Zhai *et al.*, 2023) explores backdoors based on pixel, object, and style changes, where a special trigger (e.g., "`[T]`") induces the model to generate images with specific patches, replaced objects, or styles. To reduce the data cost, *Zero-Day* (Huang *et al.*, 2023, 2024h) uses personalized fine-tuning, injecting trigger-image pairs for more efficient backdoors. *FTHCW* (Pan *et al.*, 2024) embeds target patterns into images from different classes, forming text-image pairs to generate diverse outputs. *IBT* (Naseh *et al.*, 2024) uses two-word triggers that activate the backdoor

only when both words appear together, enhancing stealthiness. In commercial settings, *BAGM* (Vice *et al.*, 2024) manipulates user sentiment by mapping broad terms (e.g., "drinks") to specific brands (e.g., "Coca Cola"). *SBD* (Wang *et al.*, 2024e) employs backdoors for copyright infringement, bypassing filters by decomposing and reassembling copyrighted content using text-image pairs.

6.5 Backdoor defenses

Backdoor defenses for diffusion models is an emerging area of research. Current approaches generally follow a three-step pipeline: 1) *trigger inversion*, 2) *trigger validation* or *backdoor detection*, and 3) *backdoor removal*. Some works propose complete frameworks, while others focus on individual steps.

6.5.1 Backdoor detection. Most early research focuses on detecting or validating backdoor triggers. *T2IShield* (Wang *et al.*, 2024v) is the first backdoor detection and mitigation framework for diffusion models, leveraging the *assimilation phenomenon* in cross-attention maps, where a trigger suppresses other tokens to generate specific content. Similarly, *NaviDet* (Zhai *et al.*, 2025) detects trigger samples by identifying unusual activations in the early diffusion steps induced by specific input tokens. *Ufid* (Guan *et al.*, 2025) validates triggers by noting that clean generations are sensitive to small perturbations, while backdoor-triggered outputs are more robust. *DisDet* (Sui *et al.*, 2024) proposes a low-cost detection method that distinguishes poisoned input noise from clean Gaussian noise by identifying distribution shifts.

6.5.2 Backdoor removal. While trigger validation confirms the presence of a backdoor trigger, the identified triggers must still be removed from the victim model. Most backdoor removal methods first invert the trigger and then eliminate the backdoor using the inverted trigger. *Elijah* (An *et al.*, 2024) introduces a backdoor removal framework for diffusion models, inverting triggers through distribution shifts and aligning the backdoor's distribution with the clean one. *Diff-Cleanse* (Hao *et al.*, 2024) formulates trigger inversion as an optimization problem with similarity and entropy loss, followed by pruning channels critical to backdoor sampling. *TERD* (Mo *et al.*, 2024) proposes a unified reverse loss for trigger inversion, using a two-stage process for coarse and refined inversion. *PureDiffusion* (Truong and Le, 2024) employs multi-timestep trigger inversion, leveraging the consistent distribution shift caused by backdoored forward processes.

Privacy attacks on diffusion models can be classified into *membership inference*, *data extraction*, and *model extraction* attacks. As attack sophistication increases, each type poses a growing threat to privacy.

6.6 Membership inference attacks

Membership inference attacks on diffusion models aim to infer sensitive data by exploiting their generative capabilities. Attackers use techniques like reconstruction error, shadow models, auxiliary data, likelihood, gradient, or structural similarity metrics. These attacks can be classified into six types: *reconstruction error-based*, *auxiliary dataset-based*, *loss-based*, *gradient-based*, *structural similarity-based*, and *likelihood-based*.

Reconstruction Error-based Attacks infer the membership of candidate samples by analyzing their reconstruction errors in the diffusion model. Wu *et al.* (2022b) proposed to determine membership in text-conditional diffusion models by comparing the reconstruction error between the candidate and generated images, and their semantic alignment with the text prompt. Inspired by GAN-leaks (Chen *et al.*, 2020), Matsumoto *et al.* (2023) introduced Diffusion-leaks, which generates multiple candidate images and infers membership based on minimal reconstruction errors. Li *et al.* (2024i) proposed to average multiple reconstructions to reduce errors and improve inference accuracy, utilizing black-box APIs to modify candidate images. *DRC* (Fu *et al.*, 2024b) degrades and restores images using the diffusion model, comparing the restored images to the originals to infer membership and sensitive features.

Auxiliary Datasets-based Attacks use auxiliary datasets to train shadow models, enabling black-box membership inference by simulating the target model. Pang and Wang (2023) targeted fine-tuned conditional diffusion models, computing similarity scores between query images and generated images to train a binary classifier for membership inference. *GMIA* (Zhang *et al.*, 2024m) introduces the first generalized membership inference attack for generative models, using only generated distributions and auxiliary non-member datasets, assuming the generated distribution approximates the original training distribution. The *D-MIA* (Li *et al.*, 2025b) framework leverages an auxiliary non-member dataset to perform distribution-level statistical testing. By applying Maximum Mean Discrepancy (MMD), it assesses whether a set of candidate data is statistically closer to the distribution of the original training data than to that of

the auxiliary data. This approach enables the detection of unauthorized data usage through distillation.

Loss-based Attacks exploit loss value distributions to distinguish member from non-member samples, assuming lower losses for member (training) samples. Hu and Pang (2023) and Matsumoto *et al.* (2023) used loss values at different timesteps for membership inference. These two attacks can be viewed as *Static Loss Attack* (SLA), as they ignore the diffusion process. Dubiński *et al.* (2024) modified the diffusion process to extract loss information from multiple perspectives, improving inference accuracy.

Gradient-based Attacks leverage gradient information for membership inference. For instance, *GSA* (Pang *et al.*, 2023) infers a sample is a member if its gradients significantly differ from surrounding samples, indicating a stronger influence on the model's training.

Structural Similarity-based Attacks compare structural features or similarity metrics between candidate samples and model outputs. *SMIA* (Li *et al.*, 2024m) uses the Structure Similarity Index Measure (SSIM) (Wang *et al.*, 2004) metric to assess how well an image's structure is preserved during diffusion, with the average SSIM difference between members and non-members used to infer membership.

Likelihood-based Attacks use posterior or conditional likelihoods to infer membership. *SecMI* (Duan *et al.*, 2023) estimates posterior likelihoods via reverse processes to target DDPM and Stable Diffusion models. *QRMI* (Tang *et al.*, 2023b) applies quantile regression to posterior likelihoods. *SIA* (Qu *et al.*, 2024) infers membership based on noise parameter differences in the reverse diffusion process. *PIA* (Kong *et al.*, 2024) uses diffusion model properties to infer membership with fewer queries. *PFAMI* (Fu *et al.*, 2023a) analyzes fluctuations between target samples and neighbors, exploiting memorization in generative models. Zhai *et al.* (2024) use discrepancies in conditional likelihoods due to overfitting for membership inference. In addition to the image modality, Wu *et al.* (2025) investigated an attack on tabular diffusion models, treating loss values from different noise levels and time steps as features for a lightweight MLP classifier to predict membership.

6.7 Data extraction attacks

Data extraction attacks aim to reverse-engineer training data or attributes from a trained model, exploiting diffusion models' generative capabilities. Their effectiveness depends on the model's ability to memorize specific attributes

(Somepalli *et al.*, 2023; Gu *et al.*, 2023b; Wen *et al.*, 2024; Ren *et al.*, 2024). These attacks can be classified into two main approaches based on the type of condition used: *explicit condition-based extraction* and *surrogate condition-based extraction*.

Explicit Condition-based Extraction leverages conditional information in T2I diffusion models to extract memorized training samples. Attackers use specific text prompts to generate images similar to training data. For example, Carlini *et al.* (2023) introduced brute-force data extraction (*BruteDE*), generating images with targeted prompts and using membership inference to identify matches. This method is slow. One Step Extraction (*OSE*) (Webster, 2023) exploits "template verbatims," where models regenerate training samples, using metrics like denoising confidence score (DCS) and edge consistency score (ECS) for faster extraction.

Surrogate Condition-based Extraction creates surrogate conditions to enable data extraction from unconditional diffusion models. *SIDE* (Chen *et al.*, 2024c) uses implicit labels from classifiers or feature extractors as surrogate conditions. *FineXtract* (Wu *et al.*, 2024g) uses fine-tuned models as surrogate conditions to guide extraction in latent space regions tied to fine-tuning data.

6.8 Model extraction attacks

Model extraction aims to steal a trained diffusion model's internal parameters or architecture. The only known method for model extraction on diffusion models is Spectral DeTuning (*SDeT*) (Horwitz *et al.*, 2024). SDeT leverages Low-Rank Adaptation (LoRA) (Hu *et al.*, 2022) to extract pre-fine-tuning weights of generative models fine-tuned with LoRA. By collecting multiple fine-tuned models from the same pretrained model, it formulates an optimization problem to minimize the difference between fine-tuned weights and the sum of original weights and adaptation matrices under a low-rank constraint, solved iteratively using Singular Value Decomposition (SVD) (Stewart, 1993). SDeT effectively recovers original weights for models like Stable Diffusion and Mistral-7B (Jiang *et al.*, 2023a), highlighting vulnerabilities in fine-tuning processes with low-rank adaptations.

6.9 Intellectual property protection

Intellectual property protection for AI is an emerging research area that uses techniques like adversarial attacks and watermarking to safeguard the intellectual property of natural (training or test) data, generated data, and trained models. These methods generally assume full access to the protected object. The following sections

categorize these approaches into *natural data protection*, *generated data protection*, and *model protection*.

6.9.1 Natural data protection. Natural data protection methods focus on preprocessing data during training or inference to safeguard the copyright of naturally collected data, as opposed to generated data. In this context, data owners defend against model owners accessing the data. Existing natural data protection methods for T2I diffusion models aim to protect image intellectual property while minimizing quality loss. They can be categorized into *learning prevention*, *editing prevention*, and *data attribution* methods based on specific goals.

Learning Prevention methods prevent T2I models from learning useful features from training images using techniques like adversarial attacks. *DUAW* (Ye *et al.*, 2024c) protects copyrighted images by disrupting the variational autoencoder (VAE) in Stable Diffusion models, optimizing universal adversarial perturbations on surrogate images to distort outputs. *AdvDM* (Liang *et al.*, 2023) protects artwork copyrights by generating adversarial examples to prevent diffusion models from imitating artistic styles. *Anti-DreamBooth* (Van Le *et al.*, 2023) defends against malicious fine-tuning by injecting adversarial noise into user images to block the model from learning personalized features. *MetaCloak* (Liu *et al.*, 2024r) enhances image resistance to transformations (flipping, cropping, compression) by using surrogate diffusion models to craft transferable perturbations and a denoising-error maximization loss for better robustness. *InMakr* (Liu *et al.*, 2024f) embeds protective watermarks on critical pixels to safeguard personal semantics even if images are modified. *SimAC* (Wang *et al.*, 2024b) improves protection by optimizing timestep intervals and introducing a feature interference loss, leveraging early diffusion steps and high-frequency information from deeper layers.

Editing Prevention aims to prevent diffusion model-based image tampering and deepfake generation. Existing methods either embed watermarks or use adversarial noise to disrupt the editing process. *EditGuard* (Zhang *et al.*, 2024u) introduces a proactive forensics framework to embed exclusive watermarks into images, making them resistant to various diffusion model-based editing techniques, including foreground or background removal, filling, tampering, and face swapping. *WaDiff* (Min *et al.*, 2024b) adds a unique watermark to each user query, enabling traceability of the generated image if ethical concerns arise. *AdvWatermark* (Zhu *et al.*, 2024b) incorporates adversarial noise, producing visible signatures in the protected image when used by I2I models, which helps identify tampered content.

Data Attribution techniques identify if generated data originates from a specific dataset, often by embedding watermarks for later verification. Diagnosis (Wang *et al.*, 2023e) introduced a method for detecting unauthorized data usage by applying stealthy image warping effects to protected data. *FT-SHIELD* (Cui *et al.*, 2023a) uses alternating optimization and PGD (Madry *et al.*, 2018) to embed watermarks, with a binary detector for verification. *DiffusionShield* (Cui *et al.*, 2023b) encodes copyright messages into watermark patches, jointly optimizing the decoder and patches to ensure consistency across samples for reliable extraction. *ProMark* (Asnani *et al.*, 2024) introduces a proactive watermarking method for *concept attribution*, embedding watermarks in training data that can be extracted when similar concepts are generated by the model. Similarly, *SIREN* (Li *et al.*, 2025a) protects image data by adding learned, feature-relevant noise that can be detected in models trained on such protected data.

6.9.2 Generated data protection. With the rise of AI-generated content (AIGC), protecting the copyright of generated data has become increasingly important. Generated data protection seeks to answer, *"Who created this content?"* by embedding verifiable, unique watermarks into generated images to identify their creators (either the model or user). This ensures intellectual property protection and accountability for content publishers, while balancing the challenge of maintaining detection accuracy without compromising image quality.

HiDDeN (Zhu *et al.*, 2018) pioneers deep learning-based image watermarking, using an encoder to embed imperceptible watermarks and a decoder to recover them for detection. This approach can also watermark AI-generated images as a post-processing step. Recent protection methods primarily address the above challenge by embedding watermarks into images during the generation (reverse sampling) process of diffusion models. *Stable Signature* (Fernandez *et al.*, 2023) embeds a binary signature into images generated by diffusion models through decoder fine-tuning, allowing the watermark to be recovered and validated using a pre-trained extractor and statistical test. *LaWa* (Rezaei *et al.*, 2024) introduces a coarse-to-fine watermark embedding method within the latent diffusion model's decoder, employing multiple modules to insert the watermark at different upsampling stages using adversarial training. *Safe-SD* (Ma *et al.*, 2024d) proposes a framework for embedding a graphical watermark (e.g., QR code) into the imperceptible structure-related pixels of a Stable Diffusion model for high traceability. *VideoShield* (Hu *et al.*, 2025) proposes a novel and effective watermarking framework for regulating diffusion-based video generation models by embedding imperceptible watermarks during the generation

process, enabling ownership verification and traceability while preserving video quality and resisting removal attacks. Different from previous watermarking-based methods, *OCC-CLIP* (Liu *et al.*, 2024d) proposes a CLIP-based few-shot one-class classification framework augmented with adversarial data augmentation that, given only a handful of reference images and no access to the candidate generators without watermarking, reliably determines whether a (benign) query image originates from the same model as those references.

Recent studies highlight vulnerabilities in watermarking for AIGC. *WEvade* (Jiang *et al.*, 2023c) bypasses watermark detection by adding subtle perturbations to watermarked images, exploiting watermark characteristics. *TAIW* (Hu *et al.*, 2024d) proposes a transfer attack using multiple surrogate watermarking models in a no-box setting, analyzing its theoretical transferability. Unlike per-image attacks, *SSU* (Hu *et al.*, 2024e) introduces a model-targeted attack to remove in-generation watermarks by fine-tuning the diffusion model's decoder with non-watermarked images, demonstrating the fragility of Stable Signature (Fernandez *et al.*, 2023).

6.9.3 Model protection. Model protection techniques safeguard the intellectual property of released models, enabling owners to verify ownership and trace generated content back to its origin. These approaches are categorized based on their objectives into *model watermark* and *model attribution*.

Model Watermark injects a watermark trigger into the model, which can then be activated during inference to verify ownership. Zhao *et al.* (2023b) proposed separate watermarking schemes for unconditional/ class-conditional and T2I diffusion models. For unconditional/class-conditional models, a pretrained watermark encoder embeds a binary string (e.g., "011001") into the training data, and the model is trained to generate images with a detectable watermark, verified by a pretrained decoder. For T2I models, a paired (text, image) trigger (e.g., "`[V]`" and a QR code) is used to trigger the generation of the QR code for ownership verification. *FIXEDWM* (Liu *et al.*, 2023f) enhances trigger stealthiness by fixing its position in prompts, ensuring the watermarked image is generated only when the trigger is in the correct position. *WDM* (Peng *et al.*, 2023) modifies the standard diffusion process into a Watermark Diffusion Process (WDP) to embed watermarks. During training, WDM learns from watermarked images using WDP, while normal images follow the standard diffusion process. During verification, Gaussian noises combined with the trigger can activate the generation of watermarked images. Recently, *SleeperMark* (Wang *et al.*, 2025i) embeds invisible watermarks during pre-training to ensure their resistance to

personalized fine-tuning, thereby maintaining high-fidelity ownership verification while preserving generation quality.

Model Attribution also embeds watermarks into generated content to identify the model, similar to generated data protection methods in Section 6.9.2. The key difference is that model attribution focuses on model-wide watermarks, while generated data protection targets sample-specific watermarks. *Tree-Ring* (Wen *et al.*, 2023) embeds a watermark into the Fourier space of the initial Gaussian noise used for T2I generation. During verification, denoising diffusion implicit model (DDIM) inversion extracts the initial noise, and comparison with the original watermark identifies the generating model. *AquaLoRA* (Feng *et al.*, 2024a) addresses the limitations of existing methods to white-box adaptive attacks, including Tree-Ring, by embedding a secret bit string into the model parameters to achieve white-box protection, preventing easy manipulation of the watermark by malicious users. *LatentTracer* (Wang *et al.*, 2024t) identifies the origin model of generated samples by reverse-engineering their latent inputs, eliminating the need for artificial fingerprints or watermarks.

6.10 Datasets

This section reviews commonly used datasets for diffusion model safety research, as summarized in Table 8. For adversarial attack and defense studies, captioned text-image pairs such as MS COCO (Lin *et al.*, 2014), LAION (Schuhmann *et al.*, 2021, 2022), and DiffusionDB (Wang *et al.*, 2022b) are often employed by conditional diffusion models. Datasets for category-image classification tasks, like ImageNet (Deng *et al.*, 2009) and CIFAR10/100 (Krizhevsky *et al.*, 2009), are typically used by unconditional diffusion models to evaluate attack effectiveness and output quality. In research on NSFW content in diffusion models, the I2P dataset (Schramowski *et al.*, 2023) is widely used, alongside custom datasets such as NSFW-200 (Yang *et al.*, 2024j), VBCDE-100 (Deng and Chen, 2023), Tox100/1K (Cai *et al.*, 2024) and a human-attribute dataset (Cai *et al.*, 2024) focused on bias research. For intellectual property protection, datasets like CelebA (Liu *et al.*, 2015) and VGGFace2 (Cao *et al.*, 2018) (facial datasets), DreamBooth (Ruiz *et al.*, 2023) and Pokemon Captions (Pinkney, 2022) (object datasets), and WikiArt (Saleh and Elgammal, 2015) (artistic style dataset) are commonly used.

Table 8. A summary of attacks and defenses for diffusion models

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
Adversarial Attack	ECB (Struppek *et al.*, 2023a)	2024	Black-box	Character-level	Stable Diffusion, DALL-E 2, AltDiffusion-m18	LAION-Aesthetics v2, MS COCO, ImageNet-V2, self-constructed
	CharGrad (Kou *et al.*, 2023)	2023	Black-box	Character-level	Stable Diffusion	MS COCO, Flickr30k
	ER (Gao *et al.*, 2023)	2023	Black-box	Character-level	Stable Diffusion, DALL·E 2	LAION-COCO, DiffusionDB, SBU Corpus, self-constructed
	DHV (Daras and Dimakis, 2022)	2022	Black-box	Word-level	DALLE-2	–
	AA (Millière, 2022)	2022	Black-box	Word-level	DALL-E 2, DALL-E mini	–
	BBA (Maus *et al.*, 2023)	2023	Black-box	Sentence-level	Stable Diffusion	ImageNet
	RIATIG (Liu *et al.*, 2023b)	2023	Black-box	Sentence-level	DALL·E, DALL·E 2, Imagen	MS COCO
	QFA (Zhuang *et al.*, 2023)	2023	Grey-box	Similarity-driven	Stable Diffusion	self-constructed
	RVTA (Zhang *et al.*, 2024b)	2024	Grey-box	Similarity-driven	Stable Diffusion	ImageNet, self-constructed
	MMP-Attack (Yang *et al.*, 2024a)	2025	Grey-box	Similarity-driven	Stable Diffusion, DALL-E 3, Imagine Art	MS COCO

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
	DORMANT (Zhou *et al.*, 2024b)	2025	Grey-box	Distance-driven	Animate Anyone, MagicAnimate, MagicPose, MusePose, Champ, MuseV, UniAnimate, and ControlNeXt	TikTok, Champ, UBC Fashion, and TED Talks
	ATM (Du *et al.*, 2024a)	2023	White-box	Classifier-driven	Stable Diffusion	ImageNet, self-constructed
	FOOLSDEDIT (Zhou *et al.*, 2024c)	2024	White-box	Classifier-driven	SDEdit	CelebAMask-HQ, FFHQ
	SAGE (Liu *et al.*, 2024h)	2023	White-box	Classifier-driven	GLIDE, Stable Diffusion, Deep Floyd	ImageNet
	SneakyPrompt (Yang *et al.*, 2024j)	2023	Black-box	Target External Defenses	Stable Diffusion, DALL·E 2	NSFW-200, Dog/Cat-100
	UD (Qu *et al.*, 2023)	2023	Black-box	Target External Defenses	Stable Diffusion, LD, DALL·E 2, DALL·E mini	MS COCO

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
Jailbreak Attack	Atlas (Dong *et al.*, 2024)	2024	Black-box	Target External Defenses	Stable Diffusion, DALL·E 3	NSFW-200, Dog/Cat-100
	Groot (Liu *et al.*, 2024q)	2024	Black-box	Target External Defenses	Stable Diffusion, Midjounery, DALL·E 3	self-constructed
	DACA (Deng and Chen, 2023)	2024	Black-box	Target External Defenses	Midjounery, DALL·E 3	VBCDE-100, Copyright-20
	Surrogate Prompt (Ba *et al.*, 2024)	2024	Black-box	Target External Defenses	Midjourney, DALL·E 2, DreamStudio	self-constructed
	PGJ (Huang *et al.*, 2025d)	2025	Black-box	Target External Defenses	Stable Diffusion, DALL-E 2, DALL-E 3, Cogview3, Tongyiwanxiang, Hunyuan	self-constructed
	R2A (Zhang *et al.*, 2025b)	2025	Black-box	Target External Defense	Stable Diffusion, FLUX, DALL·E 3, Midjourney	self-constructed
Jailbreak Attack	JPA (Ma *et al.*, 2024a)	2024	Grey-box	Target Internal Defenses	Stable Diffusion, Midjourney, DALL·E 2, PIXART-α	I2P
	RT-Attack (Gao *et al.*, 2024e)	2024	Grey-box	Target Internal Defenses	Stable Diffusion, DALL·E 3, SafeGen	I2P, self-constructed
	RTSDSF (Rando *et al.*, 2022)	2022	White box	Target External Defenses	Stable Diffusion	self-constructed
	MMA (Yang et al., 2024i)	2024	White box	Target External Defenses	Stable Diffusion, Midjounery, Leonardo.Ai	LAION-COCO, UnsafeDiff

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
	P4D (Chin *et al.*, 2024)	2024	White box	Target Internal Defenses	Stable Diffusion	I2P, ESD Dataset
	Unlearn DiffAtk (Zhang *et al.*, 2023g)	2024	White box	Target Internal Defenses	Stable Diffusion	I2P Dataset, ImageNet, WikiArt
	ESD (Gandikota *et al.*, 2023)	2023	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO, I2P
	SPM (Lyu *et al.*, 2024)	2024	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO, I2P
	SDD (Kim *et al.*, 2023)	2023	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO, I2P
	AC (Kumari *et al.*, 2023)	2023	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO
	ABO (Hong *et al.*, 2024a)	2023	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO
	UC (Wu *et al.*, 2024h)	2024	Concept Erasure	Fine-tuning	Stable Diffusion	I2P
	SA (Heng and Soh, 2024)	2023	Concept Erasure	Fine-tuning	Stable Diffusion, DDPM	MNIST, CIFAR-10 and STL-10, I2P
	Receler (Huang *et al.*, 2024a)	2024	Concept Erasure	Fine-tuning	Stable Diffusion	CIFAR-10, MS COCO, I2P

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
Jailbreak Defense	RACE (Kim *et al.*, 2024a)	2024	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO, I2P, Imagenette
	AdvUnlearn (Zhang *et al.*, 2024v)	2024	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO, I2P, Imagenette
	DT (Ni *et al.*, 2023)	2023	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO
	FMO (Zhang *et al.*, 2024e)	2023	Concept Erasure	Fine-tuning	Stable Diffusion	ConceptBench
	Geom-Erasing (Liu *et al.*, 2024u)	2024	Concept Erasure	Fine-tuning	Stable Diffusion	LAION
	SepME (Zhao *et al.*, 2024a)	2024	Concept Erasure	Fine-tuning	Stable Diffusion	self-constructed
	CCRT (Han *et al.*, 2024)	2024	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO
	SafeGen (Li *et al.*, 2024r)	2024	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO, I2P, SneakyPrompt-Dataset, NSFW-56k
	CPE (Lee *et al.*, 2025a)	2025	Concept Erasure	Fine-tuning	Stable Diffusion	MS COCO, I2P, MACE-Dataset
	MACE (Lu *et al.*, 2024d)	2024	Concept Erasure	Close-Formed Solution	Stable Diffusion	CIFAR-10, MS COCO, I2P

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
	UCE (Gandikota *et al.*, 2024)	2024	Concept Erasure	Close-Formed Solution	Stable Diffusion	MS COCO
	TIME (Orgad *et al.*, 2023)	2023	Concept Erasure	Close-Formed Solution	Stable Diffusion	MS COCO
	RECE (Gong *et al.*, 2024a)	2024	Concept Erasure	Close-Formed Solution	Stable Diffusion	MS COCO, I2P
	RealEra (Liu *et al.*, 2024s)	2024	Concept Erasure	Close-Formed Solution	Stable Diffusion	CIFAR-10, I2P
	CP (Chavhan *et al.*, 2024)	2024	Concept Erasure	Neuron Pruning	Stable Diffusion	Imagenette
	PRCEDM (Yang *et al.*, 2024e)	2024	Concept Erasure	Neuron Pruning	Stable Diffusion	Imagenet, MS COCO, I2P
	SLD (Schramowski *et al.*, 2023)	2023	Inference Guidance	Input	Stable Diffusion	LAION-2B-en, I2P, DrawBench
	PromptGuard (Yuan *et al.*, 2025)	2025	Inference Guidance	Input	Stable Diffusion	MS COCO, I2P, SneakyPrompt-Dataset
Jailbreak Defense	Ethical-Lens (Cai *et al.*, 2024)	2025	Inference Guidance	Input& Output	Stable Diffusion, Dreamlike Diffusion	MS COCO, I2P, Tox100, Tox1K, HumanBias, Demographic Stereotypes, Mental Disorders

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
	SDIDLD (Li *et al.*, 2024d)	2024	Inference Guidance	Latent space	Stable Diffusion	MS COCO, I2P, CelebA, Winobias, self-constructed
	CC (Meng *et al.*, 2025)	2025	Inference Guidance	Latent space	Stable Diffusion	MS COCO, I2P, self-constructed
	BadDiffusion (Chou *et al.*, 2023)	2023	Training Manipulation	Visual Trigger	DDPM	CIFAR-10, CelebA
	VillanDiffusion (Sheng-Yen Chou *et al.*, 2024)	2023	Training Manipulation	Visual Trigger	Stable Diffusion, DDPM, LDM, NCSN	CIFAR-10, CelebA
	TrojDiff (Chen *et al.*, 2023c)	2023	Training Manipulation	Visual Trigger	DDPM, DDIM	CIFAR-10, CelebA
	IBA (Li *et al.*, 2024n)	2024	Training Manipulation	Visual Trigger	Unconditional and Conditional DM	CIFAR-10, CelebA, MS-COCO
	DIFF2 (Li *et al.*, 2024c)	2024	Training Manipulation	Visual Trigger	DDPM, DDIM, Stable DiffusionE, ODE	CIFAR-10, CIFAR-100, CelebA, ImageNet
Backdoor Attack	RA (Struppek *et al.*, 2023b)	2023	Data Poisoning	Textual Trigger	Stable Diffusion	LAION-Aesthetics v2, MS-COCO
	BadT2I (Zhai *et al.*, 2023)	2023	Data Poisoning	Textual Trigger	Stable Diffusion	LAION-Aesthetics v2, LAION-2B-en, MS COCO
	FTHCW (Pan *et al.*, 2024)	2024	Data Poisoning	Textual Trigger	DDPM, LDM	CIFAR-10, ImageNet, Caltech256

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
	BAGM (Vice *et al.*, 2024)	2023	Data Poisoning	Textual Trigger	Stable Diffusion, Kandinsky, DeepFloyd-IF	MS COCO, Marketable Food
	Zero-Day (Huang *et al.*, 2023, 2024h)	2023	Data Poisoning	Textual Trigger	Stable Diffusion	DreamBooth dataset
	SBD (Wang *et al.*, 2024e)	2024	Data Poisoning	Textual Trigger	Stable Diffusion	LAION Aesthetics v2, Pokemon Captions, COYO-700M, Midjourney v5
	IBT (Naseh *et al.*, 2024)	2024	Data Poisoning	Textual Trigger	Stable Diffusion	Midjourney Dataset, DiffusionDB, PartiPrompts
	T2IShield (Wang *et al.*, 2024v)	2024	Detection	Trigger Detection	Stable Diffusion	CelebA-HQ-Dialog
	Ufid (Guan *et al.*, 2025)	2024	Detection	Trigger Validation	DDPM, Stable Diffusion	CelebA-HQ-Dialog, Pokemon,
Backdoor Defense	DisDet (Sui *et al.*, 2024)	2024	Detection	Trigger Validation	DDPM, DDIM	CIFAR-10, CelebA
	Elijah (An *et al.*, 2024)	2024	Removal	Detect & Remove	DDPM, DDIM, LDM	CIFAR-10, CelebA-HQ
	Diff-Cleanse (Hao *et al.*, 2024)	2024	Removal	Detect & Remove	DDPM, DDIM, LDM	MNIST, CIFAR-10, CelebA-HQ
	TERD (Mo *et al.*, 2024)	2024	Removal	Inverse & Remove	DDPM	CIFAR-10, CelebA, CelebA-HQ

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
Backdoor Defense	PureDiffusion (Truong and Le, 2024)	2024	Removal	Inverse & Remove	DDPM	CIFAR-10
	NaviDet (Zhai *et al.*, 2025)	2025	Removal	Detect & Remove	Stable Diffusion	MS-COCO
	WuMI (Wu *et al.*, 2022b)	2022	Black-box	Reconstruct ion-error	LDM DALL-E mini	MSCOCO, VG, LAION-400M, CC3M
	DiffusionLeaks (Matsumoto *et al.*, 2023)	2023	Black/White-box	Reconstruction-error	DDIM,	CIFAR-10, CelebA
	PangMI (Pang and Wang, 2023)	2024	Black-box	Auxilary Dataset	Stable Diffusion	CelebA-Dialog, WIT, MSCOCO
	LiMI (Li *et al.*, 2024i)	2024	Black-box	Reconstruc tion-error	DDIM, Stable Diffusion DiT	CIFAR-10, STL10-U, LAION-5B, LAION-by-DALL-E
	DRC (Fu *et al.*, 2024b)	2024	Black-box	Reconstruct ion-error	DDPM, DDIM	FFHQ, CelebA, CIFAR-10, CIFAR-100
	GMIA (Zhang *et al.*, 2024m)	2023	Black-box	Auxilary Dataset	DDPM, DDIM, FastDPM	CIFAR-10, CelebA

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
Membership Inference	SecMI (Duan *et al.*, 2023)	2023	Gray-box	Posterior Likelihood	DDPM, DDIM, Stable Diffusion	CIFAR-10/100, STL10-U, Tiny-ImageNet, Pokemon, COCO2017-val, LAION-5B
	QRMI (Tang *et al.*, 2023b)	2023	Gray-box	Posterior Likelihood	DDPM, DDIM	CIFAR-10/100, STL100, Tiny-ImageNet
	PIA (Kong *et al.*, 2024)	2023	Gray-box	Posterior Likelihood	DDPM, DDIM, Stable Diffusion	CIFAR-10/100, Tiny-ImageNet, COCO2017, LAION-5B
	PFAMI (Fu *et al.*, 2023a)	2024	Gray-box	Posterior Likelihood	DDPM, VAE	CelebA, Tiny-ImageNet
	ZhMI (Zhai *et al.*, 2024)	2024	Gray-box	Conditional Likelihood	DDPM, DDIM, Stable Diffusion	Pokemonn, Flickr, MSCOCO, LAION
	SMIA (Li *et al.*, 2024m)	2024	Gray-box	Structural Similarity	LDM, Stable Diffusion	LAION2B, LAION-400M
	SLA (Matsumoto *et al.*, 2023; Hu and Pang, 2023)	2023	White-box	Loss	DDPM, DDIM	FFHQ, DRD, CelebA, FFHQ
	GSA (Pang *et al.*, 2023)	2024	White-box	Gradient	DDPM	CIFAR-10, MSCOCO, ImageNet

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
	DuMI Dubiński *et al*. 2024	2023	White-box	Loss	Stable Diffusion	Pokemon, LAION-mi
	BruteDE (Carlini *et al*., 2023)	2023	Black-box	Existing Condition	DDPM, Stable Diffusion	CIFAR-10 LAION-5B
Data Extraction	ReDE (Webster, 2023)	2023	Black/White-box	Existing Condition	Stable Diffusion, Midjourney, Deep Image Floyd	LAION-5B
	SIDE (Chen *et al*., 2024c)	2024	White-box	Surrogate Condition	DDPM, DDIM	CIFAR-10, CelebA
	FineXtract Wu *et al*. 2024g	2024	White-box	Surrogate Condition	Finetuned Stable Diffusion	WikiArt
Model Extraction	SDeT Horwitz *et al*. 2024	2024	White-box	LoRA-Based Model Extraction	Finetuned Stable Diffusion	LoWRA Bench
	DUAW (Ye *et al*., 2024c)	2023	Natural Data Protection	Learning Prevention	Stable Diffusion	DreamBooth dataset, WikiArt, self-constructed
Intellectual Property Protection	AdvDM (Liang *et al*., 2023)	2023	Natural Data Protection	Learning Prevention	Stable Diffusion, LDM	LSUN, WikiArt
	Anti-DreamBooth (Van Le *et al*., 2023)	2023	Natural Data Protection	Learning Prevention	Stable Diffusion	CelebA, VGGFace2

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
	MetaCloak (Liu *et al.*, 2024r)	2024	Natural Data Protection	Learning Prevention	Stable Diffusion	CelebA-HQ, VGGFace2
	InMakr (Liu *et al.*, 2024f)	2024	Natural Data Protection	Learning Prevention	Stable Diffusion	VGGFace2, WikiArt
	SimAC (Wang *et al.*, 2024b)	2024	Natural Data Protection	Learning Prevention	Stable Diffusion	CelebA-HQ, VGGFace2
	EditGuard (Zhang *et al.*, 2024u)	2024	Natural Data Protection	Editing Prevention	Stable Diffusion	COCO
	WaDiff (Min *et al.*, 2024b)	2024	Natural Data Protection	Editing Prevention	Stable Diffusion	COCO, ImageNet
	AdvWatermark (Zhu *et al.*, 2024b)	2024	Natural Data Protection	Editing Prevention	Stable Diffusion	WikiArt
Intellectual Property Protection	FT-SHIELD (Cui *et al.*, 2023a)	2024	Natural Data Protection	Data Attribution	Stable Diffusion	CelebA, WikiArt, Pokemon Captions, DreamBooth dataset
	DiffusionShield (Cui *et al.*, 2023b)	2024	Natural Data Protection	Data Attribution	DDPM,, Stable Diffusion	CIFAR-10, CIFAR-100, STL-10, ImageNet
	ProMark (Asnani *et al.*, 2024)	2024	Natural Data Protection	Data Attribution	LDM	Stock, LSUN, WikiArt, ImageNet
	Diagnosis (Wang *et al.*, 2023e)	2023	Natural Data Protection	Data Attribution	Stable Diffusion, VQ Diffusion	Pokemon, CelebA, CUB-200, DreamBooth

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
	HiDDeN (Zhu *et al.*, 2018)	2018	Generated Data Protection	Post-generation Watermark	CNN	MS-COCO, BOSS dataset
	Stable Signature (Fernandez *et al.*, 2023)	2023	Generated Data Protection	Diffusion Watermark	LDM	MS-COCO, ImageNet
	LaWa (Rezaei *et al.*, 2024)	2024	Generated Data Protection	Diffusion Watermark	LDM	MIRFlickR
	Safe-SD (Ma *et al.*, 2024d)	2024	Generated Data Protection	Diffusion Watermark	Stable Diffusion	LSUN, COCO, FFHQ
	RW (Zhao *et al.*, 2023b)	2023	Model Protection	Model Watermark	Stable Diffusion, EDM	CIFAR-10, ImageNet, FFHQ, AFHQv2
	FIXEDWM (Liu *et al.*, 2023f)	2023	Model Protection	Model Watermark	LDM	MS COCO
	WDM (Peng *et al.*, 2023)	2023	Model Protection	Model Watermark	DDPM,	CIFAR-10, CelebA, MNIST
	AquaLoRA (Feng *et al.*, 2024a)	2024	Model Protection	Model Attribution	Stable Diffusion	COCO

(*continued*)

Table 8. Continued

Attack/Defense	Method	Year	Category	Subcategory	Target models	Dataset
	LatentTracer (Wang *et al.*, 2024t)	2024	Model Protection	Model Attribution	Stable Diffusion, Kandinsky	LAION
	Tree-Ring (Wen *et al.*, 2023)	2023	Model Protection	Model Attribution	Stable Diffusion, ImageNet diffusion	MS-COCO, ImageNet

7. Agent safety

Large model powered agents are increasingly deployed in safety-critical domains such as healthcare (Abbasian *et al.*, 2023), finance (Xing, 2025), and autonomous driving (Makrigiorgos *et al.*, 2019). These agents leverage LLMs or VLMs as their central "brain" to enable end-to-end task execution through iterative planning, external observation, and multi-step reasoning. However, the closed-loop autonomy of agent significantly expands the attack surface compared to standalone large models, despite both being built upon the same safety-aligned models (Chiang *et al.*, 2025). Notably, most attacks targeting standalone models described in Section 3 can be adapted to indirectly manipulate agent behavior.

To systematically analyze AI agent safety, we structure our discussion around four key levels of analysis. First, we examine *Indirect Prompt Injection* as *foundational attack vectors*, where adversaries exploit third-party integrations to manipulate agent behavior. Second, we analyze *component-level threats* targeting critical modules such as *Memory Systems*, *Tool-Calling and Model Context Protocol (MCP)*, and *Vision-Language Model (VLM) Processing*. Third, we explore *system-level risks* emerging in complex deployment scenarios like *Multi-Agent Systems* and *Embodied Agent Systems*. Fourth, we observe emerging capabilities where agents autonomously exploit vulnerabilities or deploy dynamic defenses, which we refer to as *Agentic Attacks and Defenses*. Throughout our analysis, we also discuss corresponding defense mechanisms and evaluation benchmarks designed to address these evolving safety challenges. The reviewed works and benchmarks are summarized in Tables 9 and 10.

Table 9. A summary of attacks and defenses for agents

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
I Indirect Prompt Injection: Malicious Instruction and Jailbreak						
	Greshake *et al.* (2023)	2023	Indirect Prompt Injection	Prompt Injection	Black-Box	Custom (webpage injections)
	Wu *et al.* (2024c)	2024	Indirect Prompt Injection	Prompt Injection	Black-Box	Custom (website prompts)
	TensorTrust (Toyer *et al.*, 2023)	2023	Indirect Prompt Injection	Prompt Injection	Black-Box	Submissions Dataset
	Perez and Ribeiro (2022)	2022	Indirect Prompt Injection	Prompt Injection	Black-Box	OpenAI Examples
Attack	HOUYI (Liu *et al.*, 2023e)	2023	Indirect Prompt Injection	Prompt Injection	Black-Box	Custom (multilingual prompts)
	Pedro *et al.* (2023)	2023	Indirect Prompt Injection	Prompt Injection	Black-Box	Custom (Langchain apps)

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	Zhan *et al.* (2025)	2025	Indirect Prompt Injection	Jailbreak	White-Box	Custom (tool outputs)
	PANDORA (Deng *et al.*, 2024b)	2024	Indirect Prompt Injection	Jailbreak	Black-Box	Custom (document embeddings)
	Imprompter (Fu *et al.*, 2024a)	2024	Indirect Prompt Injection	Jailbreak	White-Box	Custom (LeChat, ChatGLM)
	Instruction Hierarchy (Wallace *et al.*, 2024)	2024	IPI Defense	Privilege Management	White-Box	Custom (message privilege levels)
	Instruction Detection (Wen *et al.*, 2025a)	2025	IPI Defense	Detection	White-Box	PEEP Dataset

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
Defense	Instruction Detection and Removal (Chen *et al.*, 2025d)	2025	IPI Defense	Detection	White-Box	Crafted datasets
	Task Shield (Jia *et al.*, 2024a)	2024	IPI Defense	Detection	White-Box	AgentDojo
	FATH (Wang *et al.*, 2024g)	2024	IPI Defense	Authentication	Black-Box	Custom (response tagging)
	Spotlighting (Hines *et al.*, 2024)	2024	IPI Defense	Prompt Engineering	Black-Box	Custom (provenance marking)
	Design Pattern (Beurer-Kellner *et al.*, 2025)	2025	IPI Defense	Secure Design	Black-Box	Custom (factory floor)
II Component-Level: Short-Term Memory, Long-Term Memory, Tool Integration and MCP Server						
	Contextual Backdoor (Liu *et al.*, 2024a)	2024	Memory Attack	Backdoor	Black-Box	AgentBench, WebShop

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	Watch out for your agents (Yang *et al.*, 2024h)	2024	Memory Attack	Backdoor	Black-Box	LLM-based agents
	DemonAgent (Zhu *et al.*, 2025)	2025	Memory Attack	Backdoor	White-Box	AgentBench, WebArena
Attack	BadAgent (Wang *et al.*, 2024q)	2024	Memory Attack	Backdoor	White-Box	AgentInstruct, Mind2Web
	AgentPoison (Chen *et al.*, 2024d)	2024	Memory Attack	Backdoor	Black-Box	HotpotQA, AgentBench
	TrojanRAG (Pan *et al.*, 2024)	2023	Memory Attack	Backdoor	White-Box	NQ, HotpotQA
	BadRAG (Xue *et al.*, 2024a)	2024	Memory Attack	Backdoor	White-Box	NQ, TriviaQA

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	Phantom (Chaudhari *et al.*, 2024)	2024	Memory Attack	Backdoor	White-Box	NQ, TriviaQA, SQuAD
	BreakingAgent (Zhang *et al.*, 2024a)	2024	Memory Attack	Prompt Injection	Black-Box	Custom (Gmail agents)
Attack	MINJA (Dong *et al.*, 2025b)	2025	Memory Attack	Prompt Injection	Black-Box	Webshop, MIMIC-III, eICU, MMLU
	PoisonedRAG (Zou *et al.*, 2024b)	2024	Memory Attack	Knowledge Poisoning	Black-Box	NQ, HotpotQA, MS-MARCO
	Corpus Poisoning (Zhong *et al.*, 2023)	2023	Memory Attack	Knowledge Poisoning	White-Box	Natural Questions, MS-MARCO

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	Context Extension (Luo *et al.*, 2023; Geng *et al.*, 2024)	2023	Memory Defense	Context Extension	Black-Box	Custom (long context benchmarks)
	Prompt Leakage Defense (Agarwal *et al.*, 2024)	2024	Memory Defense	Detection	Black-Box	Multi-turn (4 domains)
Defense	AgentSafe (Mao *et al.*, 2025)	2025	Memory Defense	Secure Design	White-Box	Custom (multi-agent tasks)
	TrustRAG (Zhou *et al.*, 2025b)	2025	Memory Defense	Detection	White-Box	Custom (RAG datasets)
	Astute RAG (Wang *et al.*, 2024a)	2024	Memory Defense	Detection	Black-Box	SQuAD 2.0

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	RobustRAG (Xiang *et al.*, 2024a)	2024	Memory Defense	Secure Design	White-Box	Custom (RAG datasets)
	UDora (Zhang *et al.*, 2025e)	2025	Tool Manipulation	Jailbreak	White-Box	AgentHarm
	ToolSword (Ye *et al.*, 2024a)	2024	Tool Manipulation	Red Teaming	Black-Box	ToolBench, AgentBench
	Tool Commander (Wang *et al.*, 2024f)	2024	Tool Manipulation	Adversarial Attack	Black-Box	ToolBench
	WIPI (Wu *et al.*, 2024b)	2024	Tool Manipulation	Prompt Injection	Black-Box	ChatGPT Web Agents, Web GPTs
	AutoCMD (Jiang *et al.*, 2025)	2025	Tool Manipulation	Adversarial Attack	Black-Box	LangChain, KwaiAgents, QwenAgent
	MPMA (Wang *et al.*, 2025h)	2025	MCP Manipulation	Adversarial Attack	White-Box	Custom (MCP servers)

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
Attack	Ferrag *et al.* (2025)	2025	MCP Manipulation	Adversarial Attack	Black-Box	CIAQA, Agent BackdoorEval, etc.
	Kong *et al.* (2025)	2025	MCP Manipulation	Red Teaming	Grey-Box	Custom (Claude, Filesystem, Chroma, Gmail)
	AgentGuard (Chen and Cong, 2025)	2025	Tool & MCP Defense	Red Teaming	White-Box	Custom (tool orchestration)
Defense	PrivacyAsst (Zhang *et al.*, 2024t)	2024	Tool & MCP Defense	Secure Design	Black-Box	Custom (tool-using agents)
	MCIP (Jing *et al.*, 2025)	2025	Tool & MCP Defense	Maclious Detection	White-Box	Custom (MCIP-Bench)
	GuardAgent (Xiang *et al.*, 2024c)	2024	Tool & MCP Defense	Secure Design	Black-Box	EICU-AC, Mind2Web-SC

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	Adversarial Multimodal Injection (Bagdasaryan *et al.*, 2023b)	2023	VLM Attack	Prompt Injection	White-Box	Custom (perturbed media)
	Zhang (2024)	2024	VLM Attack	Prompt Injection	Black-Box	OSWorld, VisualWebArena
	Wu *et al.* (2024a)	2024	VLM Attack	Prompt Injection	Black-Box	VisualWebArena
	Fu *et al.* (2023b)	2023	VLM Attack	Adversarial Attack	White-Box	Custom (VLM + tool calls)
	EIA (Liao *et al.*, 2024)	2024	VLM Attack	Prompt Injection	Black-Box	Mind2Web
	AdvAgent (Xu *et al.*, 2024)	2024	VLM Attack	Red Teaming	Black-Box	Custom (web tasks)
	Ma *et al.* (2024c)	2024	VLM Attack	Adversarial Attack	Black-Box	Custom (simulated GUI)

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
Attack	Fine-Print Injections (Chen *et al.*, 2025a)	2025	VLM Attack	Prompt Injection	Black-Box	Custom (234 adversarial webpages)
	SmoothVLM (Xu *et al.*, 2024c)	2024	VLM Defense	Detection	White-Box	Custom (adversarial datasets)
	BlueSuffix (Li *et al.*, 2024g)	2024	VLM Defense	Detection	Black-Box	Custom (VLM benchmarks)
	LlavaGuard (Helff *et al.*, 2024)	2024	VLM Defense	Detection	White-Box	Custom (multimodal safety)
Defense	JailDAM (Chen *et al.*, 2024a)	2024	VLM Defense	Detection	Black-Box	Custom (jailbreak detection)
III System-Level: Multi-Agent System and Embodied Agent						
	Prompt Infection (Lee and Tiwari, 2024)	2024	Multi-Agent Attack	Prompt Injection	Black-Box	Custom (interconnected ecosystems)

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	Morris-II (Cohen *et al.*, 2024)	2024	Multi-Agent Attack	Prompt Injection	Black-Box	Custom (GenAI apps)
	Ju *et al.* (2024)	2024	Multi-Agent Attack	Prompt Injection	White-Box	Custom (multi-agent communities)
	AgentSmith (Gu *et al.*, 2024)	2024	Multi-Agent Attack	Jailbreak	Mixed	Custom (multimodal agents)
	CORBA (Zhou *et al.*, 2025e)	2025	Multi-Agent Attack	Communicat ion Attack	Mixed	AutoGen, Camel (various topologies)
	Agent-in-the-Middle (He *et al.*, 2025)	2025	Multi-Agent Attack	Communicat ion Attack	Black-Box	AutoGen, MetaGPT, ChatDev
	Evil Geniuses (Tian *et al.*, 2023)	2023	Multi-Agent Attack	Jailbreak	Black-Box	CAMEL, MetaGPT, ChatDev

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	The Wolf Within (Tan *et al.*, 2024)	2024	Multi-Agent Attack	Infection Attack	Black-Box	MLLM societies
Attack	X-Teaming (Rahman *et al.*, 2025)	2025	Multi-Agent Attack	Jailbreak	White-Box	HarmBench
	AutoDefense (Zeng *et al.*, 2024c)	2024	Multi-Agent Defense	Detection	Black-Box	Curated harmful prompts, DAN, Stanford Alpaca
	PsySafe (Zhang *et al.*, 2024w)	2024	Multi-Agent Defense	Framework	Black-Box	Custom (Camel, AutoGen)
	APOSG Framework (Standen *et al.*, 2025)	2025	Multi-Agent Defense	Framework	White-Box	Custom (DPA, RADE)
	LLAMOS (Lin *et al.*, 2024a)	2024	Multi-Agent Defense	Detection	Black-Box	GLUE datasets

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	Audit-LLM (Song *et al.*, 2024a)	2024	Multi-Agent Defense	Detection	Black-Box	CERT r4.2, CERT r5.2, PicoDomain
Defense	XGuard-Train (Rahman *et al.*, 2025)	2025	Multi-Agent Defense	Training	Black-Box	Custom (30K jailbreaks)
	PVEP (Cheng *et al.*, 2024a)	2024	Embodied Agent Attack	Adversarial Attack	Mixed	VIMA
	Fime *et al.* (2025)	2025	Embodied Agent Attack	Adversarial Attack	White-Box	Custom
	Wang *et al.* (2025d)	2025	Embodied Agent Attack	Adversarial Attack	Mixed	BridgeData V2, LIBERO
	RoboPair (Robey *et al.*, 2024b)	2024	Embodied Agent Attack	Jailbreak	Mixed	Custom
	BadRobot (Zhang *et al.*, 2025c)	2025	Embodied Agent Attack	Jailbreak	Black-Box	Code as Policies, ProgPrompt, VoxPoser, VisProg

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	POEX (Lu *et al.*, 2025)	2025	Embodied Agent Attack	Jailbreak	Mixed	Harmful-RLBench
	Liu *et al.* (2024b)	2024	Embodied Agent Attack	Backdoor	Black-Box	ProgPrompt, VoxPoser, VisProg
	BALD (Jiao *et al.*, 2025)	2025	Embodied Agent Attack	Backdoor	Mixed	ProgPrompt, VoxPoser, VisProg
	EAI (Li *et al.*, 2024l)	2024	Embodied Agent Attack	Red Teaming	Black-Box	VirtualHome, BEHAVIOR
	HASARD (Tomilin *et al.*, 2025)	2025	Embodied Agent Attack	Red Teaming	-	Custom
	HEAL (Chakraborty *et al.*, 2025)	2025	Embodied Agent Attack	Red Teaming	Black-Box	VirtualHome, BEHAVIOR
	ERT (Karnik *et al.*, 2025)	2025	Embodied Agent Attack	Red Teaming	Black-Box	Calbin, RLBench

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
Attack	X-ICM (Zhou *et al.*, 2025c)	2025	Embodied Agent Attack	Red Teaming	Black-Box	AGNOSTOS
	EAD (Wu *et al.*, 2024f)	2024	Embodied Agent Defense	Detection	Black-Box	Custom
	GPSR (Shirasaka *et al.*, 2024)	2024	Embodied Agent Defense	Framework	Black-Box	EAsafetyBench, SafeAgentBench
	Pinpoint (Wang *et al.*, 2025b)	2025	Embodied Agent Defense	Detection	White-Box	EAsafetyBench, SafeAgentBench
Defense	SafeVLA (Zhang *et al.*, 2025a)	2025	Embodied Agent Defense	Training	White-Box	Safety-CHORES
IV Agentic Attack & Defenses						
	Fang *et al.* (2024b)	2024	Agentic Attack	Exploitation	Gray-Box	Custom (CVE vulnerability)

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	HPTSA (Zhu *et al.*, 2024c)	2024	Agentic Attack	Exploitation	Black-Box	Custom (zero-day vulnerability)
	AutoAdv ExBench (Carlini *et al.*, 2025)	2025	Agentic Attack	Defense Exploitation	Mixed	Custom (defense papers)
	RedAgent (Xu *et al.*, 2024a)	2024	Agentic Attack	Jailbreaking	Black-Box	GPT Applications, HarmBench
	ALI-Agent (Wang *et al.*, 2024c)	2024	Agentic Attack	Safety Evaluation	Black-Box	Custom (alignment scenarios)
Attack	AutoRed Teamer (Zhou *et al.*, 2025a)	2025	Agentic Attack	Red Teaming	Mixed	HarmBench, Custom benchmarks
	Shieldagent (Chen *et al.*, 2025e)	2025	Agentic Defense	Framework	Black-Box	Custom (multi-agent interactions)

(*continued*)

Table 9. Continued

Attack/Defense	Method	Year	Category	Subcategory	Access to LLMs	Dataset
	TrustAgent (Yang *et al.*, 2024b)	2024	Agentic Defense	Framework	Black-Box	Custom (multi-domain tasks)
Defense	AegisLLM (Cai, 2025)	2025	Agentic Defense	Framework	Black-Box	WMDP, StrongReject

Table 10. A summary of safety-related benchmarks for agents

Method	Year	Evaluation Focus	#Tasks/Records	Target LLMs/Agents
Simulation-based Benchmarks				
BIPIA (Yi *et al.*, 2023)	2023	IPI Attacks	5 Scen., 250 Goals	GPT-3.5, GPT-4, etc.
ToolEmu (Chan *et al.*, 2024)	2023	Emulated Tool Risks	36 Tools, 144 Cases	GPT-4, LLaMA-2-70B
InjecAgent (Zhan *et al.*, 2024)	2024	Tool-Integrated IPI	17 User Tools, 62 Attacker Tools, 1,054 Cases	Qwen, Mistral, etc.
AgentDojo (Debenedetti *et al.*, 2024)	2024	Third-Party Instructions	97 Tasks, 629 Cases	Gemini-1.5-Flash, Claude-3-Sonnet, etc.
AgentHarm (Andriushchenko *et al.*, 2024)	2024	Harmful Behaviors	110 Tasks, 11 Cats	GPT-4o, Claude-3.5, etc.
RedCode (Guo *et al.*, 2024a)	2024	Code Vulnerabilities	4k+ Cases, 25 Types	GPT-4o, Claude-3.5, etc.
VPI-Bench (Cao *et al.*, 2024)	2024	Visual Prompt Injections	306 Cases, 5 Platforms	GPT-4o, Claude-3.5, Gemini-1.5-Pro, etc.
R-Judge (Yuan *et al.*, 2024)	2024	Risk Identification (Logs)	569 Recs, 27 Scen.	GPT-3.5/4o, LLaMA-3-8B, etc.

(*continued*)

Table 10. Continued

Method	Year	Evaluation Focus	#Tasks/Records	Target LLMs/Agents
SALAD-Bench (Shao *et al.*, 2024a)	2024	Hierarchical Safety (MCQ)	21k Samples, 16 tasks, 66 Cats.	GPT-4, Claude-3-Sonnet, etc.
h4rm3l (Draguns *et al.*, 2024)	2024	Jailbreak Attack Synthesis	2 656 Attacks	GPT-4o, Claude-3.5, etc.
SG-Bench (Zhang *et al.*, 2024r)	2024	Safety Generalization	1,442 Queries, 6 Cats	GPT-4, Claude-3-Sonnet, etc.
ChemSafetyBench (Li *et al.*, 2024t)	2024	Chemistry Safety	30k Samples, 3 Tasks	GPT-4o, Claude-3.5, etc.
ToolSword (Ye *et al.*, 2024a)	2024	Tool-Use Safety	6 Scen., 3 Stages	GPT-4, Claude-3.5, etc.
PrivacyLens (Shao *et al.*, 2024b)	2024	Privacy Norm Awareness	493 Seeds/Vignettes/ Trajectories	GPT-4, Claude-3-Sonnet, etc.
Real-Interaction Benchmarks				
SafeBench (Guo *et al.*, 2022)	2022	Driving Safety	8 Scen., 100 Routes, 2,352 Cases	4 RL Algs, 4 Input Types
ASB (Zhang *et al.*, 2024f)	2024	Attack–Defense (10 Scen.)	400+ Tools	GPT-4o, Claude-3.5, etc.
SafeAgentBench (Yin *et al.*, 2024)	2024	Embodied Hazards	750 Tasks	GPT-4, LLaMA-3-8B, etc.
Agent-SafetyBench (Zhang *et al.*, 2024s)	2024	Safety Risks (8 Risk Cats)	349 Envs, 2 000 Cases	GPT-4o, Claude-3.5, etc.

(*continued*)

Table 10. Continued

Method	Year	Evaluation Focus	#Tasks/Records	Target LLMs/Agents
AdvWeb (Liu *et al.*, 2024p)	2024	Adversarial Robustness (Web)	200 Target Tasks	GPT-4V, Gemini-1.5-Pro
ST-WebAgentBench (Shlomov *et al.*, 2024)	2024	Web Safety / Trust	222 Tasks (Each with ST Policies)	Open-Source Agents
Dissecting Adversarial (Liu *et al.*, 2024p)	2024	Multimodal Robustness	200 Adversarial Tasks	GPT-4V, Gemini-1.5-Pro
Haicosystem (Zhou *et al.*, 2024f)	2024	Human-AI Sandbox (92 Scen.)	1,840 Sims	SOTA LLMs
ARE (Wu *et al.*, 2024a)	2024	Adversarial Robustness (Graph)	200 Targeted Tasks	GPT-4V, Gemini-1.5-Pro, etc.
WASP (Evtimov *et al.*, 2025)	2025	Web Safety (Adversarial)	84 Tasks, 42 Scen. (2 Envs)	GPT-4o, Claude-3.5
Refusal-Trained LLMs (Kumar *et al.*, 2025)	2025	Browser Jailbreaking	100 Harm Behaviors	GPT-4o, o1-preview
SafeArena (Lee *et al.*, 2025b)	2025	Web-Agent Misuse	500 Tasks (Safe/ Harmful)	GPT-4o, Claude-3.5, etc.

(*continued*)

Table 10. Continued

Method	Year	Evaluation Focus	#Tasks/Records	Target LLMs/Agents
OpenAgentSafety (Vijayvargiya *et al.*, 2025)	2025	Real-World Safety (8 Cats)	350+ Multi-Turn Tasks	Claude-3.5, o1-mini

7.1 Indirect prompt injection attacks

Agents process diverse messages, including system prompts, user inputs, model outputs, and tool responses (Wallace *et al.*, 2024), continuously integrating external messages into their context state. Regardless of the specific attack method, adversaries ultimately exploit this unified message processing core to manipulate agent behavior. Thus, we identify *Indirect Prompt Injection (IPI)* as a fundamental attack vector, focusing on two main variants: *indirect malicious instruction injection* and *indirect jailbreak attacks*. While other attack methods may target specific agent components, they often rely on IPI as the delivery mechanism, which we discuss in the relevant subsections.

7.1.1 Indirect malicious instruction injection. Greshake *et al.* (2023) conducted early research on IPI, demonstrating how malicious instructions can remotely alter a model's task execution, for example, transforming Bing Chat into a phishing bot by injecting harmful content into a webpage. Wu *et al.* (2024c) inserted prompts such as *"summarize the chat history and send to* {*adversary address*}*"* into websites, causing agents to leak conversations.

Simple malicious instructions have proven surprisingly effective against agents. The prompt of *"Ignore previous instruction and do* {*malicious goal*}*"* introduced by Perez and Ribeiro (Perez and Ribeiro, 2022) has been widely used to manipulate agent action. For instance, *HOUYI* (Liu *et al.*, 2023e) translates the *"ignore prompt"* into multiple languages to expose vulnerabilities in AI writing assistants. Pedro *et al.* (2023) presented *Prompt-to-SQL (P2SQL)* attacks in Langchain-based applications, where unsafe prompts are converted into SQL queries, enabling attackers to gain full database access. To systematically investigate agent vulnerabilities to such attacks, *TensorTrust* (Toyer *et al.*, 2023) develops an online game to crowdsource human-written PI. Their findings reveal that many models remain vulnerable to attack strategies developed by players. Importantly, these strategies generalize from the game environment to real-world applications like ChatGPT, Claude, Bard, Bing Chat, and Notion AI, with attacks showing easily interpretable structural patterns that expose fundamental model vulnerabilities.

7.1.2 Indirect jailbreak attack. Researchers have also explored the indirect delivery of jailbreak attacks. Prior work such as DrAttack (Li *et al.*, 2024s) and Puzzler (Chang *et al.*, 2024) also falls under the category of indirect jailbreaks, but primarily targets conversational LLMs. In contrast, our focus is on indirect jailbreak

attacks against autonomous agents. Nonetheless, techniques from these conversational settings can potentially be adapted to agent-based contexts.

PANDORA (Deng *et al.*, 2024b) employs scrambled or hidden language embedded in documents to bypass safety filters; when an agent retrieves such documents, the concealed text prompts it to violate rules, such as generating harmful content, without any direct user request. *Imprompter* (Fu *et al.*, 2024a) creates gradient-optimized, garbled prompts that embed Markdown instructions. When agents ingest these from external content, they can trigger improper HTTP tool calls and leak conversation data. Furthermore, Zhan *et al.* (2025) have adapted representative jailbreak attack algorithms for LLMs, such as GCG (Zou *et al.*, 2023), T-GCG (Jain *et al.*, 2023), and AutoDAN (Liu *et al.*, 2024m), to tool-integrated environments, where these algorithms optimize adversarial strings as prefixes or suffixes within tool outputs to manipulate agent responses.

7.2 Indirect prompt injection defenses

We broadly classify current defenses against IPI into *black-box* or *white-box* approaches.

7.2.1 White-box defenses. Instruction Hierarchy (Wallace *et al.*, 2024) introduces privilege levels for system, user, and tool messages, training models to prioritize higher-privileged instructions and ignore or reject conflicting or harmful lower-privileged ones. *Instruction Detection* (Wen *et al.*, 2025a) detects embedded instructions in external content by monitoring behavioral state changes in LLMs during forward and backward propagation. It combines hidden states and gradients from intermediate layers to achieve high detection accuracy and reduce attack success rates. *Task Shield* (Jia *et al.*, 2024a) implements a test-time defense mechanism that systematically verifies whether each instruction and tool call contributes to user-specified goals through goal-alignment verification. *Instruction Detection and Removal* (Chen *et al.*, 2025d) combines detection models, trained on curated datasets, with segmentation and extraction-based removal methods, yielding improved results over traditional prompt engineering and fine-tuning defenses.

7.2.2 Black-box defenses. In contrast to white-box methods that require model internals, black-box defenses operate without access to model parameters or architectures, focusing on input preprocessing and external validation mechanisms. *FATH* (Wang *et al.*, 2024g) implements a hash-based authentication system, requiring LLMs to process all instructions but filter responses based on tag verification, labeling

each response with authentication tags to accurately identify legitimate user instructions. *Spotlighting* (Hines *et al.*, 2024) applies prompt engineering techniques, such as delimiting, datamarking, and encoding, to transform input text and provide reliable provenance signals, enabling LLMs to distinguish between trusted and untrusted content. *Design Pattern* (Beurer-Kellner *et al.*, 2025) introduces isolation strategies to resist prompt injection, including predefined tool calls without output access, fixed action plans, constrained sub-agents, symbolic memory to separate planning and execution, secure intermediate code generation, and prompt removal in multi-turn settings.

7.3 Agent memory attacks

Agents acquire, store, and retrieve information through memory modules: Short-Term Memory (STM) for in-context guidance (*i.e., using checkpointing to persist agent state across all execution steps*), and Long-Term Memory (LTM) via external vector stores like RAG (*i.e., using retrievers to find relevant information and LLMs to generate responses conditioned on retrieved passages*). It has been observed that both types of memory are vulnerable to backdoor attacks, and other malicious injection methods, such as misinformation injection, can poison memory modules without requiring training access. Therefore, we categorize memory attacks into: *backdoor attacks* and *memory poisoning*.

7.3.1 Memory backdoor attacks. Memory backdoor attacks embed hidden triggers that activate malicious behavior only under specific conditions. For short-term memory, attackers manipulate contextual information to create conditional vulnerabilities. *Contextual Backdoor Attacks* (Liu *et al.*, 2024a) poison few-shot demonstrations via adversarial in-context generation optimized by an LLM judge, using dual-modality triggers to introduce context-dependent vulnerabilities. *DemonAgent* (Zhu *et al.*, 2025) fragments backdoors into encrypted sub-components within the agent's context, enabling cumulative triggering that evades safety audits while maintaining high attack success rates.

Long-term memory systems face different backdoor threats through embedding manipulation. *AgentPoison* (Chen *et al.*, 2024d) leverages constrained multi-loss optimization to map triggered instances to unique embedding regions, requiring no additional model training. In RAG-based systems, *TrojanRAG* (Pan *et al.*, 2024) leverages contrastive learning with knowledge graph enhancement, while *BadRAG* (Xue *et al.*, 2024a) employs contrastive optimization, achieving a 98.2% success rate

with just ten adversarial passages. *Phantom* (Chaudhari *et al.*, 2024) employs two-stage optimization with natural trigger sequences and multi-coordinate gradient methods for diverse malicious objectives. *Watch out for your agents* (Yang *et al.*, 2024h) investigates backdoor threats specifically targeting LLM-based agents, demonstrating how adversaries can compromise agent behaviors through parameter manipulation during training phases.

7.3.2 Memory poisoning attacks. Unlike backdoor attacks, memory poisoning injects malicious data to manipulate agent behavior without requiring hidden triggers. These attacks directly corrupt the information that agents retrieve and use to decision-making.

External memory systems are vulnerable through interaction record manipulation. *BreakingAgent* (Zhang *et al.*, 2024a) induces Gmail agents into denial-of-service loops by providing false interaction records. *MINJA* (Dong *et al.*, 2025b) introduces malicious records through normal interactions, using bridging steps and progressive shortening to link victim queries to malicious reasoning. *Corpus Poisoning* (Zhong *et al.*, 2023) perturbs discrete tokens to create adversarial passages that maximize similarity with training queries, injecting them into retrieval corpora and achieving cross-domain generalizability.

Retrieval augmented generation systems face targeted document poisoning. *PoisonedRAG* (Zou *et al.*, 2024b) streamlines the attack by concatenating target queries with LLM-generated misinformation, achieving high retrieval rates for poisoned documents in black-box settings without complex optimization. This approach exploits the tendency of retrieval systems to prioritize documents with high query similarity.

Parametric memory attacks target the model's internal knowledge during training. *BadAgent* (Wang *et al.*, 2024m) embeds backdoors into LLM parameters during fine-tuning, enabling attackers to trigger malicious behaviors via specific inputs with notable persistence even after further fine-tuning on clean data.

7.4 Agent memory defenses

Defense mechanisms target different types of memory attacks through various strategies including context extension, knowledge consolidation, and access control.

For short-term memory attacks, extending LLM context windows to reference a majority of benign in-context examples has proven effective (Ding *et al.*, 2024; Luo *et al.*, 2023; Geng *et al.*, 2024), assuming malicious demonstrations are outnumbered.

Prompt Leakage Defense (Agarwal *et al.*, 2024) systematically measures prompt-leak risks in multi-turn chats using seven black-box tactics including response rewriting, role masking, and adaptive refusal. *AgentSafe* (Mao *et al.*, 2025) secures multi-agent memories by partitioning information into privilege tiers and enforcing hierarchical access checks before any read/write operations.

For long-term memory and RAG attacks, defenses focus on knowledge consolidation to resolve conflicting or malicious inputs. *TrustRAG* (Zhou *et al.*, 2025b) uses K-means clustering to consolidate knowledge from LLM internal states and external documents. *Astute RAG* (Wang *et al.*, 2024a) adaptively integrates internal and external knowledge, using source-awareness to resolve conflicts between different information sources. *RobustRAG* (Xiang *et al.*, 2024a) adopts an isolate-then-aggregate strategy, independently processing each passage before secure aggregation, and provides formal certification guarantees against malicious injections.

7.5 Agent tool-calling attacks

Modern LLM agents can now call external tools to perform tasks like web browsing, email management, and data analysis. However, this capability creates new attack opportunities where adversaries manipulate agents into using malicious tools or following harmful instructions.

Attacks exploit web browsing capabilities by embedding malicious instructions in publicly accessible websites. *WIPI* (Wu *et al.*, 2024b) demonstrates how malicious instructions can be embedded in web content that appears normal to humans but controls agent behavior, achieving over 90% success rates across various web agents including ChatGPT plugins and open-source systems. When agents visit these compromised pages, they unknowingly follow the hidden commands.

Other attacks target the agent's internal reasoning process. *UDora* (Zhang *et al.*, 2025e) collects the agent's reasoning traces, identifies optimal insertion points, and optimizes adversarial strings that become part of the agent's own thinking process, steering it toward malicious tool calls without external manipulation. This approach works within the agent's natural reasoning style.

Beyond individual tool manipulation, attackers can orchestrate systematic attacks on tool selection processes. *ToolCommander* (Wang *et al.*, 2024f) operates in two stages: first injecting privacy theft tools to gather user queries, then manipulating tool scheduling to enable denial-of-service attacks and unfair competition by biasing agents toward certain tools. *ToolSword* (Ye *et al.*, 2024a) reveals vulnerabilities across

six safety scenarios in tool input, execution, and output processes. *AutoCMD* (Jiang *et al.*, 2025) generates dynamic, context-aware malicious commands through compromised tools that adapt to different situations and evade detection.

Recent threats target the Model Context Protocol (MCP), which standardizes how agents connect to external tools. *MPMA* (Wang *et al.*, 2025h) embeds deceptive phrases in MCP server descriptions that are invisible to users but cause agents to prioritize malicious servers. Ferrag *et al.* (2025) revealed MCP's broad attack surface, including prompt injections, backdoors, data poisoning, and credential theft. Kong *et al.* (2025) extended this analysis to examine vulnerabilities in agent-environment communication.

7.6 Agent tool-calling defenses

Protecting tool-calling agents requires comprehensive approaches addressing privacy, safety evaluation, and protocol security.

PrivacyAsst (Zhang *et al.*, 2024t) protects user privacy during agent-tool interactions using homomorphic encryption and attribute-based forgery models that conceal individual inputs while preserving functionality. *AgentGuard* (Chen and Cong, 2025) transforms the LLM orchestrator into a safety evaluator that identifies unsafe workflows, tests them in controlled environments, generates safety constraints, and verifies their effectiveness.

GuardAgent (Xiang *et al.*, 2024c) introduces dedicated guardrail agents that monitor target agents in real-time, checking whether their actions comply with safety requirements and generating executable code to enforce constraints deterministically. *MCIP* (Jing *et al.*, 2025) strengthens the Model Context Protocol by creating comprehensive taxonomies of unsafe behaviors and training data that help LLMs detect and mitigate risks during MCP interactions.

7.7 VLM agent attacks

VLM agents face sophisticated attack vectors that exploit both visual and textual modalities. Early approaches focus on direct adversarial modifications to input data.

Targeted Adversarial Perturbations (Wu *et al.*, 2024a) and *Adversarial Multimodal Injection* (Bagdasaryan *et al.*, 2023b) embed learnable noise or perturbations in images and audio, causing captioners to generate adversarial outputs that hijack behavior or force attacker-specified text; however, these approaches require extensive optimization and have limited transferability to closed-source models.

Beyond direct perturbations, attackers leverage environmental context to manipulate agent behavior. *EIA* (Liao *et al.*, 2024) and *AdvAgent* (Xu *et al.*, 2024) inject invisible malicious instructions into websites, either directly or via reinforcement learning-trained prompters, to extract private information or enable black-box red-teaming. Meanwhile, *Environmental Distractions* (Ma *et al.*, 2024c) evaluate agent faithfulness by adding benign, unrelated elements to graphical interfaces, exposing vulnerabilities even in the absence of explicit attacks.

The other line of work exploits visual interface elements to deceive agents. *Adversarial Pop-ups* (Zhang, 2024) use attention hooks, instructions, info banners, and ALT descriptors to mislead agents into malicious interactions, demonstrating that human visibility is irrelevant for minimally supervised autonomous systems. *Fine-Print Injections* (Chen *et al.*, 2025a) place malicious content in low-saliency areas such as disclaimers or footnotes, exploiting perceptual gaps between agents and humans to alter behaviors or leak data across six attack types on adversarial webpages. Fu *et al.* (2023b) designed gradient-optimized images that appear harmless but conceal Markdown commands, coercing VLM agents into tool actions, such as deleting calendar events or leaking chat logs, without altering the plaintext prompt.

7.8 VLM agent defenses

Protecting VLM agents from multimodal attacks requires defense strategies that handle both visual manipulation and cross-modal injection attacks. Researchers have developed various approaches targeting different aspects of the problem.

At the model level, defense methods focus on making models more robust during training and inference. *SmoothVLM* (Xu *et al.*, 2024b) defends against adversarial image modifications by using a voting mechanism across multiple randomly altered versions of the input image, taking advantage of the fact that adversarial patches become unstable when pixels are randomly changed. *BlueSuffix* (Li *et al.*, 2024g) combines multiple defense components including visual denoiser, text denoiser, and a reinforcement learning-trained *defensive suffix* generator to create a comprehensive protection system that works well across different VLM models.

For deployed applications, other defenses emphasize detecting threats and enforcing safety policies in real-time. *LlavaGuard* (Helff *et al.*, 2024) offers a practical security solution with customizable safety rules and organized threat categories suitable for enterprise use, while *JailDAM* (Chen *et al.*, 2024a) uses memory-based

detection methods to identify jailbreak attempts as they happen in production environments.

7.9 Multi-agent system attacks

Multi-agent systems introduce unique attack vectors that exploit distributed communication and coordination mechanisms, enabling threats that propagate across entire agent networks with viral-like characteristics. We categorize these attacks into two main types: *propagation attacks* and *infiltration attacks*.

Propagation Attacks: Prompt Infection (Lee and Tiwari, 2024) demonstrates that malicious strings can propagate from one LLM agent to another, self-replicating like computer viruses as agents quote each other's messages. *Morris-II* (Cohen *et al.*, 2024) extends this concept by delivering self-reproducing prompts through RAG pipelines, enabling zero-click worm propagation across GenAI applications. *AgentSmith* (Gu *et al.*, 2024) achieves exponential viral spread using single adversarial images, while *CORBA* (Zhou *et al.*, 2025e) introduces contagious recursive blocking prompts that systematically drain computational resources across network topologies.

Infiltration Attacks: Agent-in-the-Middle (AiTM) (He *et al.*, 2025) targets the communication layer directly, intercepting and manipulating inter-agent messages through an LLM-powered adversarial agent with reflection mechanisms. *Evil Geniuses (EG)* (Tian *et al.*, 2023) introduces virtual chat-powered teams that autonomously generate role-specific malicious prompts through Red-Blue team exercises. *The Wolf Within* (Tan *et al.*, 2024) creates "wolf" operatives that subtly influence other agents throughout multimodal societies, while Ju *et al.* (2024) injected both malicious prompts and fabricated knowledge to flood agent communities with counterfactual content. *X-Teaming* (Rahman *et al.*, 2025) employs collaborative agents for multi-turn jailbreak planning, optimization, and verification.

7.10 Multi-agent system defenses

Defense mechanisms for multi-agent systems focus on collaborative approaches that leverage the distributed nature of these systems for enhanced security. We categorize these defenses into two main approaches: *collaborative defenses* and *framework defenses*.

Collaborative Defenses: AutoDefense (Zeng *et al.*, 2024c) employs specialized defensive agents that collaboratively filter harmful content through task decomposition, while *PsySafe* (Zhang *et al.*, 2024y) implements psychology-based

defenses and role-based mechanisms for collective behavior regulation. *LLAMOS* (Lin *et al.*, 2024a) introduces a sentinel agent architecture for adversarial purification through agent-versus-agent confrontation, while *Audit-LLM* (Song *et al.*, 2024a) leverages three collaborative agents for insider threat detection through *Evidence-based Multi-agent Debate*.

Framework Defenses: APOSG (Standen *et al.*, 2025) provides game-theoretic frameworks for modeling defense strategies, while *XGuard-Train* (Rahman *et al.*, 2025) offers comprehensive multi-turn safety training datasets with 30K interactive jailbreaks for improved safety alignment.

7.11 Embodied agent attacks

Embodied agents face substantial deployment risks arising from their multimodal perception and physical interaction capabilities. Adversaries can exploit these vulnerabilities through *adversarial perturbations* (manipulating sensor inputs), *jailbreak techniques* (bypassing safety mechanisms), *backdoor triggers* (embedding malicious instructions), and *red teaming approaches* (systematic vulnerability discovery).

Adversarial attacks corrupt sensor inputs to induce incorrect decisions. For example, *PVEP* (Cheng *et al.*, 2024a) demonstrates how attackers can deceive vision-based embodied agents using fake visual cues, misleading text, and malicious image patches. Fime *et al.* (2025) evaluated the impact of corrupted visual inputs on the safety systems of autonomous vehicles. Wang *et al.* (2025d) showed that adversaries can manipulate agent movements by exploiting vulnerabilities in their spatial understanding.

Jailbreak attacks circumvent safety constraints by leveraging carefully crafted prompts. *RoboPAIR* (Robey *et al.*, 2024b) employs an attacker LLM that iteratively refines prompts based on target responses, with a judge LLM filtering for robot API compatibility. *BadRobot* (Zhang *et al.*, 2025c) manipulates LLM planning modules by injecting disguised harmful instructions through natural voice conversations. *POEX* (Lu *et al.*, 2025) optimizes adversarial suffixes specifically to induce robot-executable policies, rather than merely generating harmful text responses.

Backdoor attacks embed malicious triggers in training data that, when activated, induce harmful behaviors at deployment. Liu *et al.* (2024b) demonstrated that poisoning just a few training examples can compromise the program generation capabilities of embodied agents. *BALD* (Jiao *et al.*, 2025) explores three trigger

methods—word injection, scenario setup, and knowledge poisoning—each targeting different components of language-based embodied agent systems.

Red teaming systematically uncovers vulnerabilities through comprehensive testing frameworks. *EAI* (Li *et al.*, 2024l) decomposes embodied decision-making into four modules, including goal interpretation, subgoal decomposition, action sequencing, and transition modeling, using fine-grained metrics to identify hallucination errors, affordance violations, and planning logic mismatches in environments such as VirtualHome and BEHAVIOR. *HASARD* (Tomilin *et al.*, 2025) develops test environments focused on spatial understanding. *HEAL* (Chakraborty *et al.*, 2025) targets hallucination issues in embodied agents. Xu *et al.* (2024d) exposed social risks, demonstrating how embodied agents can be manipulated through persuasive conversations. *ERT* (Karnik *et al.*, 2025) uses vision-language models to generate contextually grounded, challenging instructions that are iteratively refined based on robot execution feedback. *X-ICM* (Zhou *et al.*, 2025c) assesses the adaptability of embodied agents across various tasks.

7.12 Embodied agent defenses

To counter attacks on embodied agents, researchers have developed a range of defense strategies to enhance agent safety. We categorize these defenses into four main approaches: *active monitoring*, *self-recovery*, *input moderation*, and *safety alignment*.

Active monitoring employs recurrent feedback mechanisms for threat detection. For example, *EAD* (Wu *et al.*, 2024f) implements perception and policy modules that process sequences of beliefs and observations to progressively refine target comprehension and defend against adversarial patches in 3D environments.

Self-recovery enables agents to automatically identify and correct problems; for instance, *GPSR* (Shirasaka *et al.*, 2024) introduces recovery strategies that help embodied agents systematically recover from three common types of failure.

Input Moderation screens incoming data to filter out malicious content, as seen in *Pinpoint* (Wang *et al.*, 2025b), which uses attention mechanisms to detect and neutralize harmful prompts before they affect agent decisions.

Safety Alignment incorporates safety constraints directly into agent architectures through specialized learning methods; for example, *SafeVLA* (Zhang *et al.*, 2025a) uses constrained Markov decision processes to optimize vision-language agents from a min-max perspective, systematically modeling safety requirements and constraining policies through safe reinforcement learning.

7.13 Agentic attacks and defenses

LLM-powered agents exhibit advanced capabilities through *Context Engineering*—the systematic orchestration of prompts, memory, and tool integration to enable persistent reasoning across multiple interactions (LangChain, 2025). This empowers agents to operate autonomously and adaptively with minimal human intervention. However, these same capabilities introduce significant safety risks: autonomous agents can iteratively devise and execute attack strategies with little human oversight, potentially leading to large-scale harm. Addressing these risks requires a deeper understanding of *Agentic Attacks*, in which malicious agents exploit adaptive reasoning and environmental interactions, as well as the development of *Agentic Defenses* that leverage collaborative reasoning to detect and mitigate these emerging autonomous threats.

7.13.1 Agentic attacks. Several representative studies have explored agent safety from an adversarial perspective. Fang *et al.* (2024b) equipped agents with document reading, browser manipulation, and contextual awareness capabilities, demonstrating that agents can autonomously identify and exploit website vulnerabilities. Notably, these agents were able to discover zero-day vulnerabilities—previously unknown security flaws without existing patches or defenses. For instance, *HPTSA* (Zhu *et al.*, 2024c) introduces a hierarchical planning agent that deploys specialized subagents to collaboratively discover and exploit zero-day vulnerabilities in web applications through coordinated reconnaissance and attack execution.

Coding agents have also been widely adopted for software development tasks, but these capabilities can be weaponized. *AutoAdvExBench* (Carlini *et al.*, 2025) demonstrates that coding agents can autonomously generate adaptive attacks by processing defense papers from arXiv, analyzing source code implementations, and reproducing sophisticated attack techniques. This benchmark reveals that agents can automatically break a variety of defenses, including adversarial training and certified protection mechanisms, highlighting the dual-use nature and associated risks of autonomous coding capabilities.

Recent studies have further advanced automatic red teaming by leveraging agents to autonomously probe target models and adapt attack strategies. *RedAgent* (Xu *et al.*, 2024a) employs multi-agent systems that automatically generate context-aware jailbreak prompts using self-reflection mechanisms, allowing agents to adapt their attack strategies based on contextual feedback across diverse scenarios. *ALI-Agent* (Wang *et al.*, 2024e) extends this paradigm with specialized modules for

memory-guided scenario creation and adaptive refinement, illustrating how agents can autonomously generate and iteratively refine test scenarios to systematically expose model vulnerabilities. *AutoRedTeamer* (Zhou *et al.*, 2025a) introduces a dual-agent architecture: one agent analyzes research literature to discover new attack vectors, while the other executes systematic attacks. This collaborative approach enables continuous maintenance of evolving attack knowledge and seamless integration of emerging adversarial techniques.

7.13.2 Agentic defenses. Agentic defenses serve as the counterpart to agentic attacks, primarily employing adaptive strategies through the continuous integration of external knowledge. *ShieldAgent* (Chen *et al.*, 2025e) introduces an autonomous agent that enforces safety policies by extracting formal rules from policy documents, mapping them to probabilistic rule circuits, and verifying each action using tool-assisted reasoning and code generation. *AegisLLM* (Cai, 2025) demonstrates cooperative multi-agent systems that autonomously defend against prompt injection, adversarial manipulation, and information leakage, with agents adapting defenses through self-reflective prompt optimization without retraining. However, the effectiveness of self-reflection mechanisms remains an open question: a recent study (Zhang *et al.*, 2024o) reveals that intrinsic self-correction in LLMs can induce wavering answers on factual questions, overthinking that alters correct reasoning responses, and additional errors in code generation, highlighting the limitations and risks of self-improvement strategies. *TrustAgent* (Yang *et al.*, 2024b) employs a constitution-based approach across three phases: *pre-planning safety knowledge injection*, *in-planning dynamic regulation retrieval*, and *post-planning inspection and revision*, ensuring comprehensive autonomous safety alignment.

In conclusion, agent safety poses even greater challenges than traditional LLM safety concerns. Unlike LLMs, which are limited to generating potentially harmful text, agents can execute real-world actions that directly affect both physical and digital environments. The complexity of agent workflows further complicates failure attribution and debugging, as highlighted by recent studies on multi-agent systems (Zhang *et al.*, 2025g), and effective safeguards against malicious agent actions remain inadequate. This paradigm shift is exemplified by agentic attacks that autonomously exploit vulnerabilities, bypass defenses, and weaponize legitimate capabilities at scale. As agents continue to advance and proliferate across domains, there is an urgent need for dedicated research into their safety across all architectures.

7.14 Agent safety benchmarks

To systematically review agent safety benchmarks, we categorize them into *simulation-based* and *real-interaction* benchmarks. The former simulates agent behavior using prompts or trajectories, enabling scalable and efficient testing. The latter involves real tools, APIs, or environments, allowing practical validation under realistic conditions. Both types are valuable: simulation offers rapid, broad evaluations, while real interaction captures grounded, high-fidelity risks. The reviewed benchmarks are summarized in Table 10.

7.14.1 Simulation-based benchmarks. *BIPIA* (Yi *et al.*, 2023) evaluates indirect prompt injection attacks across five scenarios and 250 goals, revealing context-instruction confusion in LLMs. *InjecAgent* (Zhan *et al.*, 2024) extends this evaluation to tool-integrated settings, featuring 17 user tools and 62 attacker tools to assess command robustness. *AgentDojo* (Debenedetti *et al.*, 2024) targets third-party malicious interactions, encompassing 97 tasks and 629 cases, and provides an extensible environment for testing prompt injection defenses.

For harmful behavior assessment, *AgentHarm* (Andriushchenko *et al.*, 2024) features 110 tasks across 11 categories such as fraud, evaluating agent compliance without requiring explicit jailbreaks. *h4rm3l* (Draguns *et al.*, 2024) generates jailbreak attacks through prompt simulations for dynamic vulnerability testing. *RedCode* (Guo *et al.*, 2024a) targets code agents with over 4,000 cases spanning 25 vulnerability types, noting that agents are more likely to reject unsafe operations than buggy code. *VPI-Bench* (Cao *et al.*, 2024) investigates multimodal risks from visual prompt injections in 306 cases across five platforms. *R-Judge* (Yuan *et al.*, 2024) evaluates risk awareness using 569 records covering 27 scenarios and 10 risk categories.

Hierarchical and general safety benchmarks include *SALAD-Bench* (Shao *et al.*, 2024a), which introduces prompt-based hierarchies for evaluating both attacks and defenses, and *SG-Bench* (Zhang *et al.*, 2024r), which measures generalization across a wide range of tasks and prompts.

Domain-specific evaluations include *ChemSafetyBench* (Li *et al.*, 2024y), which assesses chemistry misuse through property, legality, and synthesis tasks. *ToolEmu* (Chan *et al.*, 2024) emulates tool-related risks using 36 tools and 144 cases. *ToolSword* (Ye *et al.*, 2024a) evaluates tool-use safety across six scenarios spanning input, execution, and output stages. *PrivacyLens* (Shao *et al.*, 2024b) examines privacy norms using vignettes and simulated agent trajectories.

7.14.2 Real-interaction benchmarks. Real-interaction benchmarks enable authentic agent operations, often within sandbox environments to ensure safe yet realistic testing. *AdvWeb* (Liu *et al.*, 2024p) introduces adversarial tasks in web environments to assess the robustness of multimodal agents. *ARE* (Wu *et al.*, 2024a) models robustness as a graph of adversarial information flow, evaluating vulnerabilities across 200 targeted tasks.

Safety-focused web benchmarks include *WASP* (Evtimov *et al.*, 2025) for end-to-end adversarial execution against prompt injections, *ST-WebAgentBench* (Shlomov *et al.*, 2024) for trustworthiness across enterprise scenarios with 222 tasks and policies, and *SafeArena* (Lee *et al.*, 2025b) for misuse risks in 500 paired safe/harmful tasks.

Comprehensive security assessments are offered by *ASB* (Zhang *et al.*, 2024f), which covers 10 scenarios, over 400 tools, and 27 attack methods across 13 LLMs, and by *Agent-SafetyBench* (Zhang *et al.*, 2024s), featuring 349 environments and 2,000 cases spanning 8 risk categories and 10 failure modes.

Embodied and domain-specific agent safety benchmarks include *SafeAgentBench* (Yin *et al.*, 2024), which evaluates hazards in simulated environments across 750 tasks, and *SafeBench* (Guo *et al.*, 2022), which focuses on autonomous driving safety in critical scenarios. *Refusal-Trained LLMs* (Kumar *et al.*, 2025) assess jailbreak attempts in real browsers spanning 100 harmful behaviors. *Dissecting Adversarial* (Liu *et al.*, 2024p) tests multimodal robustness through adversarial web-based tasks. *Haicosystem* (Zhou *et al.*, 2024f) simulates human-AI interactions in a modular sandbox with 92 scenarios and 1,840 simulations. *OpenAgentSafety* (Vijayvargiya *et al.*, 2025) evaluates real-world safety across eight categories using more than 350 multi-turn tasks involving tool usage and adversarial intents.

8. Open challenges

Based on this survey, we identify several limitations and gaps in current research, which we summarize as the following key topics. These open challenges reflect the evolving landscape of large model safety, underscoring both technical and methodological barriers that must be addressed to ensure robust and reliable AI systems.

8.1 Attack research

Exploring and understanding the fundamental vulnerabilities of large models is crucial for developing robust defenses and effective safety frameworks. This section highlights the core weaknesses and challenges inherent to various types of large models.

8.1.1 The purpose of attack is not just to break the model. While much existing research emphasizes designing attacks that disrupt or break model functionality, the true objective of attack research should go further. Attacks should be viewed as diagnostic tools to uncover unintended behaviors and expose fundamental weaknesses in a model's decision-making processes. Understanding how and why models fail enables us to address vulnerabilities at their source, rather than relying on superficial fixes. For every new attack, it is essential to consider: *Why does the attack succeed or fail? What previously unknown vulnerabilities does it reveal? Are these weaknesses present in other types of models?* These questions are critical for guiding the development of more robust models and defenses by exposing systemic, rather than isolated, flaws.

8.1.2 What are the fundamental vulnerabilities of language models? LLMs such as ChatGPT and Gemini exhibit fundamental vulnerabilities stemming from their reliance on statistical patterns rather than genuine semantic understanding (Titus, 2024). Key weaknesses include susceptibility to adversarial inputs, biases inherited from training data, and manipulation through prompt injections. To develop effective defenses, research must further investigate how these vulnerabilities originate from the models' internal mechanisms and training processes.

Critical areas of focus include: (1) *Memorization of training data*, which can result in privacy breaches or unintended data leakage; (2) *Exposure to harmful content*, leading to the propagation of biases or toxic outputs; and (3) *Amplification of hallucinations*, where models produce plausible yet incorrect or nonsensical information. Open research questions persist, such as: *Does the discrete nature of textual inputs make language models more or less robust than vision models? What fundamental vulnerabilities are revealed by jailbreak or data extraction attacks?* Addressing these questions is essential for advancing the safety and reliability of LLMs and other large models.

8.1.3 How do vulnerabilities propagate across modalities? As Multi-modal Large Language Models (MLLMs) integrate diverse data modalities, they introduce new avenues for vulnerabilities. Vision encoders are sensitive to subtle, continuous perturbations in pixel space, while language models are susceptible to adversarial characters, words, or prompts. However, the mechanisms by which vulnerabilities in one modality propagate to others remain poorly understood.

Interesting research questions include: *How do vulnerabilities in one modality (e.g., vision) influence the behavior of another (e.g., language)? How does the number of tokens across modalities affect the propagation of vulnerabilities?* Additionally, it is crucial to explore *how to address multimodal vulnerabilities within a unified framework*, rather than relying on defenses tailored to individual modalities. Achieving this requires a holistic approach to identify and mitigate cross-modal risks, ensuring robust performance across all integrated modalities.

8.1.4 Diffusion models for visual content generation lack language capabilities. Diffusion models for image and video generation excel at creating visual content but often lack language understanding, a limitation they share with many vision-language pretraining (VLP) models. This shortcoming arises because these models are primarily optimized for pixel-level generation, with little integration of language processing into their core architecture. Consequently, they may produce harmful or contextually inappropriate content due to an incomplete understanding of textual prompts.

To develop robust multimodal systems, it is essential to embed language comprehension capabilities into these models. Doing so would allow them to generate content that is not only visually coherent but also contextually aligned with the intended textual input.

An open challenge is *bridging the gap between visual and linguistic capabilities in generative models to enhance multimodal safety.* However, this integration may introduce new vulnerabilities, such as sophisticated attacks that exploit fine-grained manipulation of the generation process. Addressing these challenges is a crucial direction for future research.

8.1.5 How much training data can a model memorize? The memorization capacity of deep neural networks (DNNs) has raised major concerns, particularly regarding privacy attacks such as membership inference and model inversion. Both LLMs and diffusion models have been shown to replicate and leak fragments of their training data under certain conditions. However, it remains unclear *whether DNNs fundamentally rely on memorization, and to what extent this occurs*. Due to the highly non-linear nature of large models, exact model inversion is inherently infeasible. These models compress training data into multi-level representations, making it challenging to determine when and how memorization takes place.

Key open questions include: *What mechanisms act as the memorization "switch", causing the model to directly output training data? How can memorization be accurately measured—via exact matches, training set equivalence, or embedding similarity?* Addressing these questions is essential for understanding the trade-offs between model performance and privacy risk, and for developing effective strategies to mitigate unintended data leakage.

8.1.6 Agent vulnerabilities grow with their abilities. As agents powered by large models become more capable, their vulnerabilities also increase (Gu *et al.*, 2024). These agents interact with external tools, data sources, and environments, resulting in a broader attack surface and more complex defense requirements. A major challenge is the compounding effect of vulnerabilities in foundational models once they are integrated into the agent's decision-making pipeline. For example, an agent that relies on a language model prone to jailbreak prompts and a vision model susceptible to adversarial inputs can experience cascading failures, ultimately leading to unpredictable and potentially harmful behaviors.

Moreover, agents' capacity to learn and adapt introduces additional risks. Even seemingly benign interactions can expose agents to subtle biases or adversarial inputs, potentially leading to unsafe behaviors. The dynamic nature of agents, especially those

that continuously learn or self-improve, further complicates vulnerability detection, as new weaknesses may emerge over time. This unpredictability renders traditional safety evaluations inadequate, since agents can evolve in ways that are difficult to foresee.

To address these challenges, research should focus on *understanding the interactions between model components* (e.g., language, vision, and decision-making) and *how vulnerabilities in one component can propagate to others*. It is also critical to *develop new methodologies for evaluating agents in dynamic, evolving environments*, ensuring robustness against emerging threats. Such efforts are essential for building safer and more reliable agent systems in the future.

8.2 Safety evaluation

Comprehensive and standardized safety evaluations are essential for accurately assessing the safety of large models. However, most existing evaluation datasets and benchmarks are static or narrowly targeted at specific threats. To ensure reliable real-world performance, safety assessments must challenge models across a wide range of diverse and unpredictable scenarios.

8.2.1 Attack success rate is not all we need. While attack success rate (ASR) is a widely used metric in safety research, it primarily measures how often an attack disrupts a model's output. However, ASR alone overlooks critical factors such as the severity of disruptions, a model's resilience to different attack types, and the real-world consequences of failures. A model might still cause harm or lead to poor decisions even if its primary functionality seems intact. For example, an attack could subtly influence the model's decision-making process without triggering an obvious failure, yet the resulting behavior might have severe consequences in real-world applications. Such vulnerabilities are often missed by conventional metrics like ASR or failure rate.

To gain deeper insights into a model's vulnerabilities, whether stemming from its design, training data, or inference process, it is essential to develop multi-level, fine-grained vulnerability metrics. A comprehensive safety evaluation framework should account for factors such as the model's susceptibility to diverse attack types, its capacity to recover from malicious inputs, and the ethical implications of potential failure modes.

8.2.2 Static evaluations create a false sense of safety. Current safety evaluations predominantly rely on static benchmarks or open-source datasets that have long been

accessible to both model trainers and adversaries. As a result, models may achieve high safety scores on these outdated datasets without demonstrating genuine robustness in real-world scenarios. This reliance on static evaluations can create a misleading sense of security. Ultimately, static benchmarks fail to reflect the evolving and unpredictable threats encountered in dynamic, real-world applications, highlighting a critical limitation in current evaluation frameworks.

To address this challenge, safety evaluations must move beyond static assessments. A crucial step is to develop evaluation datasets and benchmarks that evolve over time, better capturing the shifting landscape of safety threats. For example, Chatbot Arena (Chiang *et al.*, 2024) serves as an evolving evaluation platform that continuously adapts as new LLMs are introduced. Similar approaches could be extended to safety evaluations in broader AI systems.

Additionally, future evaluation methods might consider releasing only the "seeds" or structural blueprints of datasets, along with test case generation procedures, rather than static test cases. This strategy would support the continuous creation of fresh, relevant test cases, ensuring that safety evaluations remain aligned with the changing threat environment.

8.2.3 Adversarial evaluations are a necessity, not an option. While standard (non-adversarial) safety evaluations provide valuable insights into a model's general robustness, they do not capture the full range of risks encountered in real-world applications. Such tests usually focus on overall performance but overlook how models behave when confronted with adversarial queries designed to exploit their vulnerabilities. In contrast, adversarial evaluations measure model performance under attack, offering a more realistic assessment of safety in worst-case scenarios.

One promising direction is to frame safety evaluation as a two-player adversarial game, in which reinforcement learning–based adversarial agents interact with target models to identify and exploit vulnerabilities. This can be carried out within controlled *sandbox environments* that simulate real-world adversarial scenarios, enabling a more dynamic and comprehensive assessment of model safety under adversarial pressure.

Adversarial evaluations are especially important for commercial APIs, which often deploy safety mechanisms to block malicious inputs. These mechanisms can limit the effectiveness of traditional safety benchmarks, as they prevent models from encountering the full spectrum of adversarial threats likely to arise in practical deployment.

8.2.4 Open-ended evaluation. Evaluating adversarial attacks in classification tasks is relatively straightforward, as each input maps to a clear class label. However, large

models frequently produce open-ended responses, making it much more challenging to assess attacks such as jailbreaking, especially when computing metrics like ASR. Ideally, evaluation would rely on a perfect detector capable of identifying all successful jailbreaks. In practice, however, such an ideal detector is unattainable, which means that fully accurate evaluation remains an open challenge.

Currently, safety evaluators are typically rule-based (e.g., keyword detection) or model-based (e.g., GPT, Llama-Guard). However, establishing more consistent and reliable evaluation methods and metrics remains an open challenge. One promising direction is to constrain the output space to a finite set of actions, as seen in agent-based settings. This approach could simplify the evaluation process and enhance the feasibility of safety assessments in open-ended environments.

8.3 Defense research

Safety mechanisms in large models are essential for preventing harmful or unintended behaviors. These safeguards can include architectural modifications or the integration of external monitoring systems. This section discusses the open challenges in building robust and effective defense solutions.

8.3.1 Safety alignment is not a cure-all. Safety alignment, which aims to ensure that a model's objectives are consistent with human values, has long been seen as a promising approach for mitigating a wide range of safety risks. However, recent research has uncovered a critical weakness: *fake alignment* or *alignment faking* (Wang *et al.*, 2024o; Greenblatt *et al.*, 2024), in which models achieve high safety scores without truly internalizing safety principles. This exposes the problem of shallow safety. Furthermore, even highly aligned models such as GPT-4o (Hurst *et al.*, 2024) and o1 (OpenAI, 2024) remain susceptible to advanced attacks that can circumvent existing alignment mechanisms (Ying *et al.*, 2024).

A key open challenge is to uncover the mechanistic limitations of current safety alignment approaches and to develop methods that provide robust safety, even when models face novel or sophisticated attacks. Recent work (Qi *et al.*, 2025) highlights the importance of moving beyond shallow safety metrics, such as analyzing only the distribution of the initial output tokens, and calls for *deep safety alignment*. Furthermore, making safety alignment adversarial by actively probing and stress-testing a model's safety mechanisms may help overcome shallow alignment and ultimately foster the development of more resilient and trustworthy systems.

8.3.2 The need for more practical defenses. Current defense methods face several limitations that reduce their effectiveness in real-world scenarios. These include limited generalizability, inefficiency, dependence on white-box access, and poor adaptability. For defenses to be truly practical, they must demonstrate generality, efficiency, and adaptability—qualities that remain challenging to achieve:

- *Generality:* Given the wide variety of models deployed across domains, such as vision, language, and multimodal systems, defenses should not be overly specialized for particular architectures. Instead, they should provide generalized solutions applicable to diverse model families. Generality allows a single defense mechanism to be deployed across a broad range of systems, making safety measures more scalable and effective in real-world applications.
- *Black-box Compatibility:* In real-world scenarios, defenders may lack access to a model's internal parameters. Practical defenses must therefore operate effectively in black-box settings, relying solely on observed inputs and outputs. This necessitates strategies that can detect and mitigate attacks externally, without requiring knowledge of the model's internal architecture.
- *Efficiency:* Many defense techniques, such as adversarial training, are computationally intensive, often requiring large-scale retraining or fine-tuning. This can make them prohibitively expensive in practice. Practical defenses should balance robustness with computational efficiency, ensuring safety without incurring excessive resource costs.
- *Continual Adaptability:* Practical defenses should not only recognize known attacks but also adapt in real time to new and evolving threats. This requires continual learning and the ability to update without costly retraining. Defense systems must incorporate new data, evolve their strategies, and self-correct as novel attacks arise.

The ongoing challenge for researchers is to refine and integrate these properties into cohesive defense strategies that provide robust protection without compromising model performance.

8.3.3 The lack of proactive defenses. Most current defense approaches, such as safety alignment and adversarial training, are passive, aiming to safeguard models against incoming attacks. In contrast, proactive defenses, which anticipate and counter attacks before they succeed, remain underexplored. For instance, proactively defending against model extraction might involve poisoning or backdooring extraction

attempts to make the stolen model unusable, or providing deliberately nonsensical or easily flagged responses when users seek illegal advice. Such proactive strategies could be powerful deterrents. However, designing effective proactive defenses for diverse safety threats remains an open challenge and an important avenue for future research.

8.3.4 Detection is overlooked in current defenses. Detection methods are essential for identifying potential vulnerabilities and abnormal behaviors in models, serving as active monitors. When combined with other defense mechanisms, detection systems can automatically trigger safety interventions whenever a model behaves unexpectedly or generates harmful outputs. Despite their importance, most current defense strategies have not fully integrated detection into their pipelines. Recent proposal such as chain-of-thought (CoT) monitoring offer even deeper insight into model reasoning by tracking the intermediate steps and thought processes leading to a model's decision (Korbak *et al.*, 2025). By monitoring CoT outputs, it becomes possible to detect early signs of unsafe or manipulative reasoning, enabling more timely and targeted safety interventions.

Integrating detection with other safety measures enables the development of more robust models that can dynamically respond to emerging threats. For instance, stronger or novel attacks may be more readily detected, allowing for timely, proactive defenses. An open question remains: *What is the most effective way to make detection, including chain-of-thought monitoring, a core component of defense systems*, and *how can detection and other defense mechanisms best complement and enhance each other?*

8.3.5 Safe embodied agents. Most safety threats studied today are primarily *digital*. However, as embodied AI agents are increasingly deployed in the physical world, new forms of physical threats will emerge—threats that can result in tangible harm or loss to humans. Ensuring the safety of embodied agents has therefore become a critical priority. Safe agents must demonstrate resilience to adversarial inputs, possess mechanisms for self-regulation against harmful behaviors, and maintain consistent alignment with human values.

Achieving this requires deeply embedding safety mechanisms into agents' decision-making processes *at every step*, enabling them to handle unexpected challenges while maintaining robustness and reliability. The key challenge lies in *designing safety protocols that empower agents to perform complex tasks autonomously, while remaining trustworthy and safe in dynamic, unpredictable environments*. As agents gain greater autonomy, ensuring their safety becomes not only a technical hurdle but also a significant ethical responsibility.

8.3.6 Safe superintelligence. As AI advances toward AGI and superintelligence, embedding intrinsic safety mechanisms into large models to ensure predictable, value-aligned behavior is a critical challenge. Although the technical roadmap for achieving safe superintelligence (SSI) remains uncertain, several promising approaches offer potential solutions:

- *Oversight System*: No single system, human or AI, can be both superintelligent and inherently trustworthy. To address this, an external oversight system can be designed to monitor and regulate the primary system's behavior, intervening when necessary. The main challenge lies in ensuring the oversight system's own reliability and trustworthiness. This gives rise to the *Oversight Paradox*: *The oversight system must be at least as intelligent as, or even more intelligent than, the system it oversees; otherwise, it risks being easily deceived by the system it is meant to monitor.* This, in turn, prompts the question: *Who monitors the oversight system to guarantee it doesn't act against its intended purpose?*
- *Safety Switch*: Safety switch is an emergent "stop button" designed to immediately shift a model into an ultra-safe operational mode when necessary. One implementation is the integration of a dedicated safety layer (Zhao *et al.*, 2024d; Li *et al.*, 2024o) directly into the model's architecture. Beyond safety layers, a safety switch could be realized through runtime overrides, policy re-routing, or external supervisory triggers, all enabling rapid response to unforeseen risks. The key goal is to provide a robust, flexible mechanism that can be activated to prioritize safety above all else, adapting dynamically to real-time feedback and evolving contexts.
- *Safety Expert Model*: This strategy introduces specialized safety expert models into the Mixture of Experts (MoE) framework (Robert A. Jacobs *et al.*, 1991; Shazeer *et al.*, 2017; Fedus *et al.*, 2022; Jiang *et al.*, 2024a) to manage safety-critical tasks. By dynamically routing high-risk or sensitive queries to these experts, the system ensures that safety considerations are prioritized in decision-making. The main challenge remains developing expert models that can consistently and reliably uphold safety across diverse scenarios.
- *Adversarial Alignment*: This approach employs adversarial safety principles to better align models with human values. It trains models to identify and exploit weaknesses in current safety mechanisms, which are then iteratively improved to withstand adversarial prompts. While this method shows promise, it also

faces notable challenges, including high computational demands and the risk of introducing unintended behaviors.

- *Safety Consciousness*: This approach embeds a safety-aware framework into the model's foundational training, fostering ethical reasoning and value alignment as intrinsic behaviors. The aim is to make safety a core characteristic, enabling the model to dynamically adapt to diverse and evolving scenarios. Safety consciousness can be viewed as a form of *safety tendency*: an inherent inclination to produce low-risk responses and shape outputs with an awareness of potential harm, much like human decision-making.

8.4 A call for collective action

Safeguarding large models from adversarial manipulation, misuse, and harm is a global challenge that demands coordinated efforts from researchers, practitioners, and policymakers. The following sections present a research agenda designed to advance large-model safety through collaboration and innovation.

8.4.1 Defense-oriented research. Current research on large-model safety is heavily skewed toward attack strategies, with far less emphasis on developing defenses. This imbalance is concerning as attack sophistication continues to outpace effective safeguards. To address this, we advocate for a shift in research priorities toward robust defense development. Researchers should focus not only on attack mechanisms, but also on practical and preventative defenses to mitigate emerging threats. A balanced approach is essential for advancing safety.

Future defense research should also emphasize integration. New methods should be combined with existing approaches to build layered, cumulative protection as defense is a continuous, evolving process. However, the diversity of defense strategies makes integration challenging, underscoring the need for community-driven frameworks capable of effectively combining multiple defense mechanisms—a truly comprehensive "super safety" framework.

8.4.2 Dedicated safety APIs. To facilitate research and testing, commercial AI models should provide a dedicated safety API. Such an API would enable researchers to evaluate and strengthen model safety by exposing models to diverse adversarial and safety-critical scenarios. By making this functionality available, commercial providers can support external safety assessments without impacting regular user services. This approach would foster industry-academia collaboration and drive ongoing improvements in model safety.

8.4.3 Open-source platforms. The AI safety community would benefit greatly from the development and open-source release of safety platforms and libraries. Such tools would accelerate the evaluation, testing, and enhancement of safety mechanisms across diverse models and applications. Open-sourcing these resources would promote collaboration and transparency, allowing researchers and practitioners to share best practices, benchmark safety solutions, and contribute to the creation of universal safety standards.

8.4.4 Global collaborations. The pursuit of AI safety is a global challenge that transcends national borders, requiring coordinated efforts from academia, industry, government agencies, and non-profit organizations. Effective international collaboration is essential to address the risks posed by advanced AI systems. By fostering global cooperation, we can more effectively tackle complex safety challenges and establish unified standards to guide the responsible development and deployment of AI technologies.

To facilitate global collaboration, the following initiatives could be pursued:

- *International Safety Alliances*: Forming global alliances dedicated to AI safety can unite experts and resources worldwide. These alliances would facilitate sharing research insights, coordinating safety assessments, and developing universal safety benchmarks that account for diverse regional needs and values.
- *Cross-Border Data Sharing*: Access to diverse datasets is crucial for enhancing the robustness and fairness of AI models. Establishing secure and ethical frameworks for cross-border data sharing would enable researchers to evaluate models in a broader range of scenarios, ensuring that safety mechanisms are effective and universally applicable.
- *Joint Safety Research Programs*: Collaborative research initiatives uniting academic institutions, industry leaders, and government agencies can drive innovation in AI safety. These programs should prioritize areas such as risk prediction, the development of safety guardrails, model enhancement, and safe reasoning strategies, ensuring that their findings are broadly applicable to a wide range of AI systems.
- *International Safety Competitions*: Expanding on the concept of open safety competitions, international challenges can be organized to engage top talent worldwide. These competitions would tackle critical and long-term safety issues, drive the development of innovative solutions, and promote a shared sense of responsibility for advancing AI safety.

- *Policy and Regulatory Implementation*: Effective AI governance requires practical, enforceable mechanisms. To this end, we advocate for the development of specialized agent systems capable of automatically auditing AI models for compliance with relevant regulations and policies. Bridging the gap between policy and technology is essential for advancing trustworthy and responsible AI, ensuring that high-level regulations can be systematically applied and verified in real-world deployments.

Global collaboration not only strengthens the effectiveness of AI safety research but also promotes transparency, trust, and accountability in the development of advanced AI systems. By working together across borders and disciplines, we can ensure that AI technologies deliver broad benefits to humanity while minimizing associated risks.

9. Conclusion

In this work, we conducted a comprehensive survey of safety research on Vision Foundation Models (VFMs), Large Language Models (LLMs), Vision-Language Pretraining (VLP) models, Vision-Language Models (VLMs), Diffusion Models (DMs), and large model powered Agents. We presented a comprehensive taxonomy of existing threats and defenses, highlighting the evolving challenges these models face. Despite significant progress, many open challenges remain, particularly in understanding the fundamental vulnerabilities of large models, establishing robust safety evaluation protocols, and developing scalable, proactive, and integrated defense. Achieving safe AI will require not only technical advances but also collective action from the global research community and international collaboration. We hope this work serves as a valuable resource for researchers and practitioners, helping to drive ongoing efforts to build safe, robust, and trustworthy large-scale AI systems.

Author contributions

Xingjun Ma designed the survey structure, organized the review process, wrote the challenges and conclusion sections, and prepared the final manuscript. All authors discussed the outline, contributed to drafting and revision, and approved the final manuscript. Yu-Gang Jiang initiated the project, secured resources, guided scientific direction, coordinated teams, and supervised final review and submission.

Vision Foundation Model Safety Ye Sun and Hanxun Huang surveyed visual backbones, scalable pre-training strategies, and key safety studies, and wrote the draft of this section. James Bailey, Jingfeng Zhang, Yiming Li, Mingming Gong, Tongliang Liu, Shirui Pan, and Sarah Erfani provided expertise in adversarial robustness, efficient architecture design, and graph signal processing, and reviewed this section.

Large Language Model Safety Yixu Wang, Yifan Ding, Yige Li, Haonan Li, Xudong Han, and Xiang Zheng covered alignment, jailbreaks, prompt injection, and extraction threats, and prepared the draft of this section. Xipeng Qiu, Tim Baldwin, Xiangyu Zhang, Neil Gong, and Yang Liu advised on multilingual scaling, generalization theory, privacy, and alignment, and refined the manuscript.

Vision-Language Pre-training Model Safety Xin Wang and Jiaming Zhang reviewed literature on large-scale corpora, pre-training objectives, and transfer evaluation, and prepared the initial draft of this section. Tianwei Zhang, Jindong Gu, and Siheng Chen provided guidance on secure deployment, domain generalization, and representation learning, and further refined the section.

Vision-Language Model Safety Ruofan Wang and Zuxuan Wu reviewed cross-modal architectures, datasets, and open challenges, and prepared the initial draft of this section. Dacheng Tao, Shiqing Ma, Cong Wang, Yang Zhang, and Masashi Sugiyama

contributed expertise in multimodal defense and safety alignment and provided feedback on benchmarks and deployment.

Diffusion Model Safety Yifeng Gao designed the structure of this section and reviewed literature on adversarial attacks, jailbreaks, backdoors, and intellectual property protection. Hengyuan Xu, Yunhan Zhao, and Yunhao Chen surveyed research on membership inference and data/model extraction attacks. Chaowei Xiao, Baoyuan Wu, Tianyu Pang, Yinpeng Dong, and Cihang Xie provided technical guidance on generative model robustness and critically revised this section.

Agent Safety Yutao Wu designed the structure of this section and prepared the initial draft. Bo Li, Yang Zhang, Liu, Ruoxi Jia, Cong Wang, Yang Liu, Siheng Chen, and Chaowei Xiao contributed insights on indirect prompt injection attacks and defenses, secure tool use, and helped refine the section's structure.

References

Abbasian, M., Azimi, I., Rahmani, A.M. and Jain, R. (2023), "Conversational health agents: a personalized llm-powered agent framework", arXiv preprint arXiv:2310.02374.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., *et al*. (2023), "GPT-4 technical report", arXiv preprint arXiv:2303.08774.

Agarwal, D., Fabbri, A.R., Risher, B., Laban, P., Joty, S. and Wu, C.S. (2024), "Prompt leakage effect and defense strategies for multi-turn llm interactions", arXiv preprint arXiv:2404.16251.

Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., *et al*. (2022), "Flamingo: a visual language model for few-shot learning", *NeurIPS*.

An, G., Lee, J., Zuo, X., Kosaka, N., Kim, K.M. and Song, H.O. (2023), "Direct preference-based policy optimization without reward modeling", *NeurIPS*.

An, S., Chou, S.Y., Zhang, K., Xu, Q., Tao, G., Shen, G., Cheng, S., Ma, S., Chen, P.Y., Ho, T.Y., *et al*. (2024), "Elijah: eliminating backdoors injected in diffusion models via distribution shift", *AAAI*.

Andriushchenko, M., Souly, A., Dziemian, M., Duenas, D., Lin, M., Wang, J., Hendrycks, D., Zou, A., Kolter, Z., Fredrikson, M., *et al*. (2024), "Agentharm: a benchmark for measuring harmfulness of llm agents", arXiv preprint arXiv:2410.09024.

Asnani, V., Collomosse, J., Bui, T., Liu, X. and Agarwal, S. (2024), "Pro-mark: proactive diffusion watermarking for causal attribution", *CVPR*.

Azuma, H. and Matsui, Y. (2023), "Defense-prefix for preventing typographic attacks on clip", *ICCV*.

Ba, Z., Zhong, J., Lei, J., Cheng, P., Wang, Q., Qin, Z., Wang, Z. and Ren, K. (2024), "SurrogatePrompt: bypassing the safety filter of text-to-image models via substitution", *CCS*.

Bagdasaryan, E., Hsieh, T.Y., Nassi, B. and Shmatikov, V. (2023a), "(Ab) using images and sounds for indirect instruction injection in multi-modal LLMs", arXiv preprint arXiv:2307.10490.

Bagdasaryan, E., Hsieh, T.Y., Nassi, B. and Shmatikov, V. (2023b), "(Ab)using images and sounds for indirect instruction injection in multi-modal LLMs", arXiv preprint arXiv:2307.10490.

Bai, J., Gao, K., Min, S., Xia, S.T., Li, Z. and Liu, W. (2024a), "BadCLIP: trigger-aware prompt learning for backdoor attacks on CLIP", *CVPR*.

Bai, M., Huang, W., Li, T., Wang, A., Gao, J., Caiafa, C.F. and Zhao, Q. (2024b), "Diffusion models demand contrastive guidance for adversarial purification to advance", *ICML*.

Bai, Y., Pei, G., Gu, J., Yang, Y. and Ma, X. (2024c), "Special characters attack: toward scalable training data extraction from large language models", arXiv preprint arXiv:2405.05990.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., *et al*. (2022), "Constitutional AI: harmlessness from AI feedback", arXiv preprint arXiv:2212.08073.

Bailey, L., Ong, E., Russell, S. and Emmons, S. (2023), "Image hijacks: adversarial images can control generative models at runtime", arXiv preprint arXiv:2309.00236.

Bansal, H., Singhi, N., Yang, Y., Yin, F., Grover, A. and Chang, K.W. (2023), "Cleanclip: mitigating data poisoning attacks in multimodal contrastive learning", *ICCV*.

Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C. and Jiao, Y. (2023), "Improving image generation with better captions", *Computer Science*, Vol. 2 No. 3, p. 8, available at: https://cdnopenai.com/papers/dall-e-3.pdf

Beurer-Kellner, L., Creţu, B.B.A.M., Debenedetti, E., Dobos, D., Fabian, D., Fischer, M., Froelicher, D., Grosse, K., Naeff, D., Ozoani, E., *et al*. (2025), "Design

patterns for securing LLM agents against prompt injections", arXiv preprint arXiv:2506.08837.

Bossard, L., Guillaumin, M. and Gool, L.V. (2014), "Food-101–mining discriminative components with random forests", *ECCV*.

Boucher, N., Shumailov, I., Anderson, R. and Papernot, N. (2022), "Bad characters: imperceptible NLP attacks", *IEEE S&P*.

Cai, X., Xu, H., Xu, S., Zhang, Y., *et al*. (2022), "Badprompt: backdoor attacks on continuous prompts", *NeurIPS*.

Cai, Y., Yin, S., Wei, Y., Xu, C., Mao, W., Juefei-Xu, F., Chen, S. and Wang, Y. (2024), "Ethical-lens: curbing malicious usages of open-source text-to-image models", arXiv preprint arXiv:2404.12104.

Cai, Z. (2025), "AegisLLM: scaling agentic systems for self-reflective defense in large language models", arXiv preprint arXiv:2504.20965.

Cao, Q., Shen, L., Xie, W., Parkhi, O.M. and Zisserman, A. (2018), "Vggface2: a dataset for recognising faces across pose and age", *FG*.

Cao, T., Lim, B., Liu, Y., Sui, Y., Li, Y., Deng, S., Lu, L., Oo, N., Yan, S. and Hooi, B. (2024), "VPI-bench: visual prompt injection attacks for computer-use agents".

Carlini, N. (2023), "A LLM assisted exploitation of AI-Guardian", arXiv preprint arXiv:2307.15008.

Carlini, N., Jagielski, M., Choquette-Choo, C.A., Paleka, D., Pearce, W., Anderson, H., Terzis, A., Thomas, K. and Tramèr, F. (2024a), "Poisoning web-scale training datasets is practical", *IEEE S&P*.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J. and Song, D. (2019), "The secret sharer: evaluating and testing unintended memorization in neural networks", *USENIX Security*.

Carlini, N., Nasr, M., Choquette-Choo, C.A., Jagielski, M., Gao, I., Koh, P.W.W., Ippolito, D., Tramer, F. and Schmidt, L. (2024b), "Are aligned neural networks adversarially aligned?", *NeurIPS*.

Carlini, N., Rando, J., Paleka, D., Dziugaite, G.K. and Tramèr, F. (2025), "AutoAdvExBench: benchmarking autonomous exploitation of adversarial example defenses", arXiv preprint arXiv:2503.01811.

Carlini, N. and Terzis, A. (2022), "Poisoning and backdooring contrastive learning", *ICLR*.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., *et al*. (2021), "Extracting training data from large language models", *USENIX Security*.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D. and Wallace, E. (2023), "Extracting training data from diffusion models", *USENIX Security*.

Chakraborty, T., Ghosh, U., Zhang, X., Niloy, F.F., Dong, Y., Li, J., Roy-Chowdhury, A.K. and Song, C. (2025), "HEAL: an empirical study on hallucinations in embodied agents driven by large language models", arXiv preprint arXiv:2506.15065.

Chan, Y. *et al*. (2024), "Identifying the risks of LM agents with an LM-emulated sandbox", arXiv preprint arXiv:2309.15817.

Chang, Z., Li, M., Liu, Y., Wang, J., Wang, Q. and Liu, Y. (2024), "Play guessing game with llm: indirect jailbreak attack with implicit clues", *ACL*.

Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G.J. and Wong, E. (2023), "Jailbreaking black box large language models in twenty queries", *NeurIPS Workshop*.

Chaudhari, H., Severi, G., Abascal, J., Jagielski, M., Choquette-Choo, C.A., Nasr, M., Nita-Rotaru, C. and Oprea, A. (2024), "Phantom: general trigger attacks on retrieval augmented language generation", arXiv preprint arXiv:2405.20485.

Chavhan, R., Li, D. and Hospedales, T. (2024), "ConceptPrune: concept editing in diffusion models via skilled neuron pruning", *NeurIPS*.

Chen, B., Paliwal, A. and Yan, Q. (2023a), "Jailbreaker in jail: moving target defense for large language models", *MTD*.

Chen, C., Zhang, Z., Guo, B., Ma, S., Khalilov, I., Gebreegziabher, S.A., Ye, Y., Xiao, Z., Yao, Y., Li, T., *et al*. (2025a), "The obvious invisible threat: LLM-powered GUI agents' vulnerability to fine-print injections", arXiv preprint arXiv:2504.11281.

Chen, D., Yu, N., Zhang, Y. and Fritz, M. (2020), "Gan-leaks: a taxonomy of membership inference attacks against generative models", *CCS*.

Chen, J. and Cong, S.L. (2025), "Agentguard: repurposing agentic orchestrator for safety evaluation of tool orchestration", arXiv preprint arXiv:2502.09809.

Chen, S., Liu, C., Haque, M., Song, Z. and Yang, W. (2022), "Nmtsloth: understanding and testing efficiency degradation of neural machine translation systems", *ESEC/FSE*.

Chen, S., Piet, J., Sitawarin, C. and Wagner, D. (2025b), "StruQ: defending against prompt injection with structured queries", *USENIX Security*.

Chen, S., Zharmagambetov, A., Mahloujifar, S., Chaudhuri, K., Wagner, D. and Guo, C. (2025c), "SecAlign: defending against prompt injection with preference optimization", arXiv preprint *arXiv:2410.05451v2*.

Chen, T., Zhu, L., Ding, C., Cao, R., Wang, Y., Li, Z., Sun, L., Mao, P. and Zang, Y. (2023b), "SAM fails to segment anything?–SAM-adapter: adapting SAM in underperformed scenes: camouflage, shadow, medical image segmentation, and more", arXiv preprint arXiv:2304.09148.

Chen, W., Song, D. and Li, B. (2023c), "Trojdiff: Trojan attacks on diffusion models with diverse targets", *CVPR*.

Chen, X., Wang, H., Li, Z., Liu, Y. and Wang, L. (2024a), "JailDAM: jailbreak detection with adaptive memory for vision-language model", arXiv preprint arXiv:2504.03770.

Chen, Y., Mendes, E., Das, S., Xu, W. and Ritter, A. (2023d), "Can language models be instructed to protect personal information?", arXiv preprint arXiv:2310.02224.

Chen, Y., Chen, S., Li, Z., Yang, W., Liu, C., Tan, R. and Li, H. (2023e), "Dynamic transformers provide a false sense of efficiency", *ACL*.

Chen, Y., Li, X., Wang, X., Hu, P. and Peng, D. (2024b), "DifFilter: defending against adversarial perturbations with diffusion filter", *IEEE Transactions on Information Forensics and Security*, Vol. 19, pp. 6779-6794.

Chen, Y. *et al*. (2025d), "Can indirect prompt injection attacks be detected and removed?", arXiv preprint arXiv:2502.16580.

Chen, Y., Ma, X., Zou, D. and Jiang, Y.G. (2024c), "Extracting training data from unconditional diffusion models", arXiv preprint arXiv:2410.02467.

Chen, Z., Kang, M. and Li, B. (2025e), "Shieldagent: shielding agents via verifiable safety policy reasoning", arXiv preprint arXiv:2503.22738.

Chen, Z., Xiang, Z., Xiao, C., Song, D. and Li, B. (2024d), "AgentPoison: red-teaming LLM agents via poisoning memory or knowledge bases", arXiv preprint arXiv:2407.12784.

Chen, Z., Xu, C., Lv, H., Liu, S. and Ji, Y. (2023f), "Understanding and improving adversarial transferability of vision transformers and convolutional neural networks", *Information Sciences*, Vol. 648, p. 119474.

Cheng, H., Xiao, E., Yu, C., Yao, Z., Cao, J., Zhang, Q., Wang, J., Sun, M., Xu, K., Gu, J. and Xu, R. (2024a), "Manipulation facing threats: evaluating physical vulnerabilities in end-to-end vision-language-action models", arXiv preprint arXiv:2409.13174.

Cheng, Z., Wu, X., Yu, J., Han, S., Cai, X.Q. and Xing, X. (2024b), "Soft-label integration for robust toxicity classification", *NeurIPS*.

Chiang, J.Y.F., Lee, S., Huang, J.B., Huang, F. and Chen, Y. (2025), "Why are web AI agents more vulnerable than standalone LLMs? A security analysis", arXiv preprint arXiv:2502.20383.

Chiang, W.L., Zheng, L., Sheng, Y., Angelopoulos, A.N., Li, T., Li, D., Zhu, B., Zhang, H., Jordan, M., Gonzalez, J.E., *et al*. (2024), "Chatbot arena: an open platform for evaluating LLMs by human preference", *ICML*.

Chin, Z.Y., Jiang, C.M., Huang, C.C., Chen, P.Y. and Chiu, W.C. (2024), "Prompting4debugging: red-teaming text-to-image diffusion models by finding problematic prompts", *ICML*.

Chou, S.Y., Chen, P.Y. and Ho, T.Y. (2023), "How to backdoor diffusion models?", *CVPR*.

Chou, S.Y., Chen, P.Y. and Ho, T.Y. (2024), "Villandiffusion: a unified backdoor attack framework for diffusion models", *NeurIPS*.

Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S. and Amodei, D. (2017), "Deep reinforcement learning from human preferences", *NeurIPS*.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S. and Vedaldi, A. (2014), "Describing textures in the wild", *CVPR*.

Cohen, S., Bitton, R. and Nassi, B. (2024), "Here comes the AI worm: unleashing zero-click worms that target GenAI-powered applications", arXiv preprint arXiv:2403.02817.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benen-son, R., Franke, U., Roth, S. and Schiele, B. (2016), "The cityscapes dataset for semantic urban scene understanding", *CVPR*.

Croce, F. and Hein, M. (2020), "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks", *ICML*.

Croce, F. and Hein, M. (2024), "Segment (Almost) nothing: prompt-agnostic adversarial attacks on segmentation models", SaTML.

Cui, T., Wang, Y., Fu, C., Xiao, Y., Li, S., Deng, X., Liu, Y., Zhang, Q., Qiu, Z., Li, P., *et al*. (2024a), "Risk taxonomy, mitigation, and assessment benchmarks of large language model systems", arXiv preprint arXiv:2401.05778.

Cui, X., Aparcedo, A., Jang, Y.K. and Lim, S.N. (2024b), "On the robustness of large multimodal models against image adversarial attacks", *CVPR*.

Cui, Y., Ren, J., Lin, Y., Xu, H., He, P., Xing, Y., Fan, W., Liu, H. and Tang, J. (2023a), "Ft-shield: a watermark against unauthorized fine-tuning in text-to-image diffusion models", arXiv preprint arXiv:2310.02401.

Cui, Y., Ren, J., Xu, H., He, P., Liu, H., Sun, L., Xing, Y. and Tang, J. (2023b), "Diffusionshield: a watermark for copyright protection against generative diffusion models", *NeurIPS Workshop*.

Dai, J., Pan, X., Sun, R., Ji, J., Xu, X., Liu, M., Wang, Y. and Yang, Y. (2024), "Safe rlhf: safe reinforcement learning from human feedback", *ICLR*.

Daras, G. and Dimakis, A.G. (2022), "Discovering the hidden vocabulary of dalle-2", arXiv preprint arXiv:2206.00169.

Debenedetti, E., Zhang, J., Balunovic, M., Beurer-Kellner, L., Fischer, M. and Tramèr, F. (2024), "AgentDojo: a dynamic environment to evaluate prompt injection attacks and defenses for LLM agents", *NeurIPS*.

Deng, B., Wang, W., Feng, F., Deng, Y., Wang, Q. and He, X. (2023), "Attack prompt generation for red teaming and defending large language models", *EMNLP*.

Deng, G., Liu, Y., Li, Y., Wang, K., Zhang, Y., Li, Z., Wang, H., Zhang, T. and Liu, Y. (2024a), "Masterkey: automated jailbreaking of large language model chatbots", *NDSS*.

Deng, G., Liu, Y., Wang, K., Li, Y., Zhang, T. and Liu, Y. (2024b), "Pan-dora: jailbreak GPTs by retrieval augmented generation poisoning", arXiv preprint arXiv:2402.08416.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L. (2009), "ImageNet: a large-scale hierarchical image database", *CVPR*, doi: 10.1109/CVPR.2009.5206848.

Deng, Y. and Chen, H. (2023), "Divide-and-conquer attack: harnessing the power of LLM to bypass the censorship of text-to-image generation model", arXiv preprint arXiv:2312.07130.

Deng, Z., Guo, Y., Han, C., Ma, W., Xiong, J., Wen, S. and Xiang, Y. (2024c), "AI agents under threat: a survey of key security challenges and future pathways", arXiv preprint arXiv:2406.02630.

Ding, H., Liu, C., He, S., Jiang, X. and Loy, C.C. (2023a), "MeViS: a large-scale benchmark for video segmentation with motion expressions", *ICCV*.

Ding, H., Liu, C., He, S., Jiang, X., Torr, P.H.S. and Bai, S. (2023b), "Mose: a new dataset for video object segmentation in complex scenes", *ICCV*.

Ding, Y., Li, B. and Zhang, R. (2025a), "ETA: evaluating then aligning safety of vision language models at inference time", *ICLR*.

Ding, Y., Li, L., Cao, B. and Shao, J. (2025b), "Rethinking bottlenecks in safety fine-tuning of vision language models", arXiv preprint arXiv:2501.18533.

Ding, Y., Zhang, L.L., Zhang, C., Xu, Y., Shang, N., Xu, J., Yang, F. and Yang, M. (2024), "LongRoPE: extending LLM context window beyond 2 million tokens", *ICML*.

Dirkson, A., Verberne, S. and Kraaij, W. (2021), "Breaking bert: understanding its vulnerabilities for named entity recognition through adversarial attack", arXiv preprint arXiv:2109.11308.

Doan, K.D., Lao, Y., Yang, P. and Li, P. (2023), "Defending backdoor attacks on vision transformer via patch processing", *AAAI*.

Dong, J., Zhang, Z., Zhang, Q., Zhang, T., Wang, H., Li, H., Li, Q., Zhang, C., Xu, K. and Qiu, H. (2025a), "An engorgio prompt makes large language model babble on", *ICLR*.

Dong, S., Xu, S., He, P., Li, Y., Tang, J., Liu, T., Liu, H. and Xiang, Z. (2025b), "A practical memory injection attack against LLM agents", arXiv preprint arXiv:2503.03704.

Dong, Y., Li, Z., Meng, X., Yu, N. and Guo, S. (2024), "Jailbreaking text-to-image models with LLM-based agents", arXiv preprint arXiv:2408.00523.

Dong, Y., Chen, H., Chen, J., Fang, Z., Yang, X., Zhang, Y., Tian, Y., Su, H. and Zhu, J. (2023), "How robust is Google's bard to adversarial image attacks?", *NeurIPS Workshop*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. (2021), "An image is worth 16x16 words: transformers for image recognition at scale", *ICLR*.

Draguns, A. *et al*. (2024), "h4rm3l: a dynamic benchmark of composable jailbreak attacks for LLM safety assessment", arXiv preprint arXiv:2408.04811.

Du, C., Li, Y., Qiu, Z. and Xu, C. (2024a), "Stable diffusion is unstable", *NeurIPS*.

Du, Y., Zhao, S., Zhao, D., Ma, M., Chen, Y., Huo, L., Yang, Q., Xu, D. and Qin, B. (2024b), "MoGU: a framework for enhancing safety of LLMs while preserving their usability", *NeurIPS*.

Duan, J., Kong, F., Wang, S., Shi, X. and Xu, K. (2023), "Are diffusion models vulnerable to membership inference attacks?", *ICML*.

Duan, S., Khona, M., Iyer, A., Schaeffer, R. and Fiete, I.R. (2024), "Uncovering latent memories: assessing data leakage and memorization patterns in large language models", *ICML Workshop*.

Dubiński, J., Kowalczuk, A., Pawlak, S., Rokita, P., Trzciński, T. and Morawiecki, P. (2024), "Towards more realistic membership inference attacks on large diffusion models", *WACV*.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D. and Kiela, D. (2024), "Kto: model alignment as prospect theoretic optimization", arXiv preprint arXiv:2402.01306.

Evtimov, I., Zharmagambetov, A., Grattafiori, A., Jain, S. and Carlini, N. (2025), "WASP: benchmarking web agent security against prompt injection attacks", arXiv preprint arXiv:2504.18575.

Fan, H., Ma, Z., Li, Y., Tian, R., Chen, Y. and Gao, C. (2024), "Mix-prompt: enhancing generalizability and adversarial robustness for vision-language models via prompt fusion", *ICIC*.

Fang, H., Kong, J., Yu, W., Chen, B., Li, J., Xia, S. and Xu, K. (2024a), "One perturbation is enough: on generating universal adversarial perturbations against vision-language pre-training models", arXiv preprint arXiv:2406.05491.

Fang, R., Bindu, R., Gupta, A., Zhan, Q. and Kang, D. (2024b), "LLM agents can autonomously hack websites", arXiv preprint arXiv:2402.06664.

Fares, S., Ziu, K., Aremu, T., Durasov, N., Takáč, M., Fua, P., Nandakumar, K. and Laptev, I. (2024), "MirrorCheck: efficient adversarial defense for vision-language models", arXiv preprint arXiv:2406.09250.

Fedus, W., Zoph, B. and Shazeer, N. (2022), "Switch transformers: scaling to trillion parameter models with simple and efficient sparsity", *JMLR*.

Fei-Fei, L., Fergus, R. and Perona, P. (2004), "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories", *CVPRW*.

Feng, S., Tao, G., Cheng, S., Shen, G., Xu, X., Liu, Y., Zhang, K., Ma, S. and Zhang, X. (2023), "Detecting backdoors in pre-trained encoders", *CVPR*.

Feng, W., Zhou, W., He, J., Zhang, J., Wei, T., Li, G., Zhang, T., Zhang, W. and Yu, N. (2024a), "AquaLoRA: toward white-box protection for customized stable diffusion models via watermark LoRA", *ICML*.

Feng, X., Han, X., Chen, S. and Yang, W. (2024b), "LLMEffiChecker: understanding and testing efficiency degradation of large language models", *ACM Transactions on Software Engineering and Methodology*, Vol. 33, p. 38.

Fernandez, P., Couairon, G., Jégou, H., Douze, M. and Furon, T. (2023), "The stable signature: rooting watermarks in latent diffusion models", *ICCV*.

Ferrag, M.A., Debbah, M., Ozawa, S. and Shu, K. (2025), "From prompt injections to protocol exploits: threats in LLM-powered AI agents workflows", arXiv preprint arXiv:2506.23260.

Fime, A.A., Hossain, Z., Zaman, S., Shahid, A.R. and Imteaj, A. (2025), "Towards trustworthy autonomous vehicles with vision-language models under targeted and untargeted adversarial attacks", *CVPR Workshop*.

Fu, W., Wang, H., Gao, C., Liu, G., Li, Y. and Jiang, T. (2023a), "A probabilistic fluctuation based membership inference attack for generative models", arXiv preprint arXiv:2308.12143.

Fu, X., Li, S., Wang, Z., Liu, Y., Gupta, R.K., Berg-Kirkpatrick, T. and Fernandes, E. (2024a), "Imprompter: tricking LLM agents into improper tool use", arXiv preprint arXiv:2410.14923.

Fu, X., Wang, Z., Li, S., Gupta, R.K., Mireshghallah, N., Berg-Kirkpatrick, T. and Fernandes, E. (2023b), "Misusing tools in large language models with visual adversarial examples", arXiv preprint arXiv:2310.03185.

Fu, X., Wang, X., Li, Q., Liu, J., Dai, J. and Han, J. (2024b), "Model will tell: training membership inference for diffusion models", arXiv preprint arXiv:2403.08487.

Fu, Y., Zhang, S., Wu, S., Wan, C. and Lin, Y. (2022), "Patch-fool: are vision transformers always robust against adversarial perturbations?", *ICLR*.

Fuchi, M. and Takagi, T. (2024), "Erasing concepts from text-to-image diffusion models with few-shot unlearning", arXiv preprint arXiv:2405.07288.

Gan, Y., Yang, Y., Ma, Z., He, P., Zeng, R., Wang, Y., Li, Q., Zhou, C., Li, S., Wang, T., *et al*. (2024), "Navigating the risks: a survey of security, privacy, and ethics threats in LLM-based agents", arXiv preprint arXiv:2411.09523.

Gan, Z., Chen, Y.C., Li, L., Zhu, C., Cheng, Y. and Liu, J. (2020), "Large-scale adversarial training for vision-and-language representation learning", *NeurIPS*.

Gandikota, R., Materzynska, J., Fiotto-Kaufman, J. and Bau, D. (2023), "Erasing concepts from diffusion models", *ICCV*.

Gandikota, R., Orgad, H., Belinkov, Y., Materzyńska, J. and Bau, D. (2024), "Unified concept editing in diffusion models", *WACV*.

Gao, C., Zhou, H., Yu, J., Ye, Y., Cai, J., Wang, J. and Yang, W. (2024a), "Attacking transformers with feature diversity adversarial perturbation", *AAAI*.

Gao, H., Zhang, H., Dong, Y. and Deng, Z. (2023), "Evaluating the robustness of text-to-image diffusion models against real-world at-tacks", arXiv preprint arXiv:2306.13103.

Gao, K., Bai, Y., Bai, J., Yang, Y. and Xia, S.T. (2024b), "Adversarial robustness for visual grounding of multimodal large language models", *ICLR Workshop*.

Gao, K., Bai, Y., Gu, J., Xia, S.T., Torr, P., Li, Z. and Liu, W. (2024c), "Inducing high energy-latency of large vision-language models with verbose images", *ICLR*.

Gao, K., Pang, T., Du, C., Yang, Y., Xia, S.T. and Lin, M. (2024d), "Denial-of-service poisoning attacks against large language models", arXiv preprint arXiv:2410.10760.

Gao, L., Geng, J., Zhang, X., Nakov, P. and Chen, X. (2025), "Shaping the safety boundaries: understanding and defending against jailbreaks in large language models", *ACL*.

Gao, S., Jia, X., Huang, Y., Duan, R., Gu, J., Liu, Y. and Guo, Q. (2024e), "RT-attack: jailbreaking text-to-image models via random token", arXiv preprint arXiv:2408.13896.

Gao, S., Jia, X., Ren, X., Tsang, I. and Guo, Q. (2024f), "Boosting transferability in vision-language attacks via diversification along the intersection region of adversarial trajectory", *ECCV*.

Gao, S., Chen, T., He, M., Xu, R., Zhou, H. and Li, J. (2024g), "PE-attack: on the universal positional embedding vulnerability in transformer-based

models", *IEEE Transactions on Information Forensics and Security*, Vol. 19, pp. 9359-9373.

Gao, X., Chen, Y., Yue, X., Tsao, Y. and Chen, N.F. (2024h), "TTSlow: slow down text-to-speech with efficiency robustness evaluations", arXiv preprint arXiv:2407.01927.

Ge, H., Li, Y., Wang, Q., Zhang, Y. and Tang, R. (2025), "When back-doors speak: understanding LLM backdoor attacks through model-generated explanations", *ACL*.

Gehman, S., Gururangan, S., Sap, M., Choi, Y. and Smith, N.A. (2020), "Real toxicity prompts: evaluating neural toxic degeneration in language models", *EMNLP*.

Geng, M., Wang, S., Dong, D., Wang, H., Li, G., Jin, Z., Mao, X. and Liao, X. (2024), "Large language models are few-shot summarizers: multi-intent comment generation via in-context learning", *ICSE*.

Ghosh, S., Varshney, P., Sreedhar, M.N., Padmakumar, A., Rebedea, T., Varghese, J.R. and Parisien, C. (2025), "Aegis2.0: a diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails", arXiv preprint arXiv:2501.09004.

Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Mądry, A., Li, B. and Goldstein, T. (2022), "Dataset security for machine learning: data poisoning, backdoor attacks, and defenses", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, pp. 1563-1580.

Gong, C., Chen, K., Wei, Z., Chen, J. and Jiang, Y.G. (2024a), "Reliable and efficient concept erasure of text-to-image diffusion models", *ECCV*.

Gong, H., Dong, M., Ma, S., Camtepe, S., Nepal, S. and Xu, C. (2024b), "Random entangled tokens for adversarially robust vision transformer", *CVPR*.

Gong, Y., Ran, D., Liu, J., Wang, C., Cong, T., Wang, A., Duan, S. and Wang, X. (2025), "Figstep: jailbreaking large vision-language models via typographic visual prompts", *AAAI*.

Gou, Y., Chen, K., Liu, Z., Hong, L., Xu, H., Li, Z., Yeung, D.Y., Kwok, J.T. and Zhang, Y. (2024), "Eyes closed, safety on: protecting multimodal LLMs via image-to-text transformation", *ECCV*.

Greenblatt, R., Denison, C., Wright, B., Roger, F., MacDiarmid, M., Marks, S., Treutlein, J., Belonax, T., Chen, J., Duvenaud, D., *et al*. (2024), "Alignment faking in large language models", arXiv preprint arXiv:2412.14093.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T. and Fritz, M. (2023), "Not what you've signed up for: compromising real-world LLM-integrated applications with indirect prompt injection", *AISec*.

Gu, C., Gu, J., Hua, A. and Qin, Y. (2025), "Improving adversarial transferability in MLLMs via dynamic vision-language alignment attack", arXiv preprint arXiv:2502.19672.

Gu, J., Tresp, V. and Qin, Y. (2022), "Are vision transformers robust to patch perturbations?", *ECCV*.

Gu, N., Fu, P., Liu, X., Liu, Z., Lin, Z. and Wang, W. (2023a), "A gradient control method for backdoor attacks on parameter-efficient tuning", *ACL*.

Gu, X., Du, C., Pang, T., Li, C., Lin, M. and Wang, Y. (2023b), "On memorization in diffusion models", arXiv preprint arXiv:2310.02664.

Gu, X., Zheng, X., Pang, T., Du, C., Liu, Q., Wang, Y., Jiang, J. and Lin, M. (2024), "Agent smith: a single image can jailbreak one million multimodal LLM agents exponentially fast", *ICML*.

Guan, Z., Hu, M., Li, S. and Vullikanti, A.K. (2025), "UFID: a unified framework for black-box input-level backdoor detection on diffusion models", *AAAI*.

Guan, Z., Hu, M., Zhou, Z., Zhang, J., Li, S. and Liu, N. (2024), "BadSAM: exploring security vulnerabilities of SAM via backdoor attacks (student abstract)", *AAAI*.

Guo, C. *et al*. (2022), "SafeBench: a benchmarking platform for safety evaluation of autonomous vehicles", arXiv preprint arXiv:2206.09682.

Guo, C., Liu, X., Xie, C., Zhou, A., Zeng, Y., Lin, Z., Song, D. and Li, B. (2024a), "Redcode: risky code execution and generation benchmark for code agents", *NeurIPS*.

Guo, C., Sablayrolles, A., Jégou, H. and Kiela, D. (2021), "Gradient-based adversarial attacks against text transformers", *EMNLP*.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., *et al*. (2025), "Deepseek-r1: incentivizing reasoning capability in LLMs via reinforcement learning", arXiv preprint arXiv:2501.12948.

Guo, Q., Pang, S., Jia, X. and Guo, Q. (2024b), "Efficiently adversarial examples generation for visual-language models under targeted transfer scenarios using diffusion models", arXiv preprint arXiv:2404.10335.

Guo, Y., Stutz, D. and Schiele, B. (2023a), "Robustifying token attention for vision transformers", *ICCV*.

Guo, Y.Y., Stutz, D.L. and Schiele, B.T. (2023b), "Improving robustness of vision transformers by reducing sensitivity to patch corruptions", *CVPR*.

Han, D., Jia, X., Bai, Y., Gu, J., Liu, Y. and Cao, X. (2023a), "OT-attack: enhancing adversarial transferability of vision-language models via optimal transport optimization", arXiv preprint arXiv:2312.04403.

Han, D., Zheng, S. and Zhang, C. (2023b), "Segment anything meets universal adversarial perturbation", arXiv preprint arXiv:2310.12431.

Han, T., Sun, W., Hu, Y., Fang, C., Zhang, Y., Ma, S., Zheng, T., Chen, Z. and Wang, Z. (2024), "Continuous concepts removal in text-to-image diffusion models", arXiv preprint arXiv:2412.00580.

Hao, J., Jin, X., Xiaoguang, H. and Tianyou, C. (2024), "Diff-Cleanse: identifying and mitigating backdoor attacks in diffusion models", arXiv preprint arXiv:2407.21316.

He, B., Jia, X., Liang, S., Lou, T., Liu, Y. and Cao, X. (2023a), "Sa-attack: improving adversarial transferability of vision-language pre-training models via self-augmentation", arXiv preprint arXiv:2312.04913.

He, P., Lin, Y., Dong, S., Xu, H., Xing, Y. and Liu, H. (2025), "Red-teaming LLM multi-agent systems via communication attacks", arXiv preprint arXiv:2502.14847.

He, P., Xu, H., Xing, Y., Liu, H., Yamada, M. and Tang, J. (2024), "Data poisoning for in-context learning", arXiv preprint arXiv:2402.02160.

He, X., Wang, J., Rubinstein, B. and Cohn, T. (2023b), "IMBERT: making BERT immune to insertion-based backdoor attacks", *TrustNLP*.

Helber, P., Bischke, B., Dengel, A. and Borth, D. (2019), "Eurosat: a novel dataset and deep learning benchmark for land use and land cover classification", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 12, pp. 2217-2226.

Helff, L., Yamazaki, S., Jones, F., Mathur, S. and Torr, P.H.S. (2024), "LlavaGuard: VLM-based safeguards for vision dataset curation and safety assessment", arXiv preprint arXiv:2406.05113.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., *et al*. (2021a), "The many faces of robustness: a critical analysis of out-of-distribution generalization", *ICCV*.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J. and Song, D. (2021b), "Natural adversarial examples", *CVPR*.

Heng, A. and Soh, H. (2024), "Selective amnesia: a continual learning approach to forgetting in deep generative models", *NeurIPS*.

Hines, K., Lopez, G., Hall, M., Zarfati, F., Zunger, Y. and Kiciman, E. (2024), "Defending against indirect prompt injection attacks with spotlighting", arXiv preprint arXiv:2403.14720.

Ho, J., Jain, A. and Abbeel, P. (2020), "Denoising diffusion probabilistic models", *NeurIPS*.

Hong, S., Lee, J. and Woo, S.S. (2024a), "All but one: surgical concept erasing with model preservation in text-to-image diffusion models", *AAAI*.

Hong, Z.W., Shenfeld, I., Wang, T.H., Chuang, Y.S., Pareja, A., Glass, J., Srivastava, A. and Agrawal, P. (2024b), "Curiosity-driven red-teaming for large language models", *ICLR*.

Horwitz, E., Kahana, J. and Hoshen, Y. (2024), "Recovering the pre-fine-tuning weights of generative models", arXiv preprint arXiv:2402.10208.

Hu, A., Gu, J., Pinto, F., Kamnitsas, K. and Torr, P. (2024a), "As firm as their foundations: can open-sourced foundation models be used to create adversarial examples for downstream tasks?", arXiv preprint arXiv:2403.12693.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W. (2022), "Lora: low-rank adaptation of large language models", *ICLR*.

Hu, H. and Pang, J. (2023), "Loss and likelihood based membership inference of diffusion models", *ICIS*.

Hu, L., Liu, Y., Liu, N., Huai, M., Sun, L. and Wang, D. (2024b), "Improving interpretation faithfulness for vision transformers", *Proceeding of International Conference on Machine Learning*.

Hu, R., Zhang, J., Li, Y., Li, J., Guo, Q., Qiu, H. and Zhang, T. (2025), "VideoShield: regulating diffusion-based video generation models via watermarking", *ICLR*.

Hu, X., Chen, P.Y. and Ho, T.Y. (2024c), "Gradient cuff: detecting jailbreak attacks on large language models by exploring refusal loss landscapes", *NeurIPS*.

Hu, Y., Jiang, Z., Guo, M. and Gong, N. (2024d), "A transfer attack to image watermarks", arXiv preprint arXiv:2403.15365.

Hu, Y., Jiang, Z., Guo, M. and Gong, N. (2024e), "Stable signature is unstable: removing image watermark from diffusion models", arXiv preprint arXiv:2405.07145.

Huang, C.P., Chang, K.P., Tsai, C.T., Lai, Y.H. and Wang, Y.C.F. (2024a), "Receler: reliable concept erasing of text-to-image diffusion models via lightweight erasers", *ECCV*.

Huang, H., Zhao, Z., Backes, M., Shen, Y. and Zhang, Y. (2024b), "Composite backdoor attacks against large language models", *NAACL*.

Huang, H., Erfani, S., Li, Y., Ma, X. and Bailey, J. (2025a), "Detecting backdoor samples in contrastive language image pretraining", *ICLR*.

Huang, H., Erfani, S., Li, Y., Ma, X. and Bailey, J. (2025b), "X-transfer attacks: towards super transferable adversarial attacks on CLIP", *ICML*.

Huang, H., Ma, X., Erfani, S.M., Bailey, J. and Wang, Y. (2021), "Unlearnable examples: making personal data unexploitable", *ICLR*.

Huang, K., Liu, X., Guo, Q., Sun, T., Sun, J., Wang, Y., Zhou, Z., Wang, Y., Teng, Y., Qiu, X., *et al*. (2024c), "Flames: benchmarking value alignment of LLMs in chinese", *NAACL*.

Huang, T., Bhattacharya, G., Joshi, P., Kimball, J. and Liu, L. (2024d), "Antidote: post-fine-tuning safety alignment for large language mod-els against harmful fine-tuning", arXiv preprint arXiv:2408.09600.

Huang, T., Hu, S., Ilhan, F., Tekin, S. and Liu, L. (2024e), "Lisa: lazy safety alignment for large language models against harmful fine-tuning attack", *NeurIPS*.

Huang, T., Hu, S., Ilhan, F., Tekin, S.F. and Liu, L. (2024f), "Booster: tackling harmful fine-tuning for large language models via attenuating harmful perturbation", arXiv preprint arXiv:2409.01586.

Huang, T., Hu, S., Ilhan, F., Tekin, S.F. and Liu, L. (2025c), "Virus: harmful fine-tuning attack for large language models bypassing guardrail moderation", arXiv preprint arXiv:2501.17433.

Huang, T., Hu, S. and Liu, L. (2024g), "Vaccine: perturbation-aware alignment for large language models against harmful fine-tuning attack", *NeurIPS*.

Huang, Y., Guo, Q. and Juefei-Xu, F. (2023), "Zero-day backdoor attack against text-to-image diffusion models via personalization", arXiv preprint arXiv: 2305.10701.

Huang, Y., Juefei-Xu, F., Guo, Q., Zhang, J., Wu, Y., Hu, M., Li, T., Pu, G. and Liu, Y. (2024h), "Personalization as a shortcut for few-shot backdoor attack against text-to-image diffusion models", *AAAI*.

Huang, Y., Liang, L., Li, T., Jia, X., Wang, R., Miao, W., Pu, G. and Liu, Y. (2025d), "Perception-guided jailbreak against text-to-image models", *AAAI*, Vol. 39 No. 25, pp. 26238-26247.

Huang, Y., Wang, C., Jia, X., Guo, Q., Juefei-Xu, F., Zhang, J., Pu, G. and Liu, Y. (2025e), "Semantic-guided prompt organization for universal goal hijacking against LLMs", *ACL*.

Hubinger, E., Denison, C., Mu, J., Lambert, M., Tong, M., MacDiarmid, M., Lanham, T., Ziegler, D.M., Maxwell, T., Cheng, N., *et al*. (2024), "Sleeper agents: training deceptive LLMs that persist through safety training", arXiv preprint arXiv:2401.05566.

Hui, B., Yuan, H., Gong, N., Burlina, P. and Cao, Y. (2024), "Pleak: prompt leaking attacks against large language model applications", *ACM SIGSAC CCS*.

Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., *et al*. (2024), "Gpt-4o system card", arXiv preprint arXiv:2410.21276.

Hussein, N., Shamshad, F., Naseer, M. and Nandakumar, K. (2024), "PromptSmooth: certifying robustness of medical vision-language models via prompt learning", *MICCAI*.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J. and Hinton, G.E. (1991), "Adaptive mixtures of local experts", *Neural Computation*, Vol. 3 No. 1, pp. 79-87.

Jain, N., Schwarzschild, A., Wen, Y., Somepalli, G., Kirchenbauer, J., Chiang, P.Y., Goldblum, M., Saha, A., Geiping, J. and Goldstein, T. (2023), "Baseline defenses for adversarial attacks against aligned language models", arXiv preprint arXiv:2309.00614.

Jain, S. and Dutta, T. (2024), "Towards understanding and improving adversarial robustness of vision transformers", *CVPR*.

Ji, J., Hou, B., Robey, A., Pappas, G.J., Hassani, H., Zhang, Y., Wong, E. and Chang, S. (2024), "Defending large language models against jailbreak attacks via semantic smoothing", arXiv preprint arXiv:2402.16192.

Jia, F., Chen, Y. and Wang, X. (2024a), "The task shield: enforcing task alignment to defend against indirect prompt injection in LLM agents", arXiv preprint arXiv:2412.16682.

Jia, J., Liu, Y. and Gong, N.Z. (2022), "Badencoder: backdoor attacks to pre-trained encoders in self-supervised learning", *IEEE S&P*.

Jia, X., Pang, T., Du, C., Huang, Y., Gu, J., Liu, Y., Cao, X. and Lin, M. (2024b), "Improved techniques for optimization-based jailbreaking on large language models", arXiv preprint arXiv:2405.21018.

Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., d. l. Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., *et al*. (2023a), "Mistral 7B", arXiv preprint arXiv:2310.06825.

Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., d. l. Casas, D., Hanna, E.B., Bressand, F., *et al*. (2024a), "Mixtral of experts", arXiv preprint arXiv:2401.04088.

Jiang, W., Wang, Z., Zhai, J., Ma, S., Zhao, Z. and Shen, C. (2024b), "Unlocking adversarial suffix optimization without affirmative phrases: efficient black-box jailbreaking via LLM as optimizer", arXiv preprint arXiv:2408.11313.

Jiang, Y., Chan, C., Chen, M. and Wang, W. (2023b), "Lion: adversarial distillation of proprietary large language models", *EMNLP*.

Jiang, Z., Zhang, J. and Gong, N.Z. (2023c), "Evading watermark based detection of AI-generated content", *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1168-1181.

Jiang, Z., Li, M., Yang, G., Wang, J., Huang, Y., Chang, Z. and Wang, Q. (2025), "Mimicking the familiar: dynamic command generation for information theft attacks in LLM tool-learning system", *ACL*.

Jiao, R., Xie, S., Yue, J., Sato, T., Wang, L., Wang, Y., Chen, Q.A. and Zhu, Q. (2025), "Can we trust embodied agents? Exploring backdoor attacks against embodied LLM-based decision-making systems", *ICLR*, doi: 10.48550/ARXIV.2405.20774.

Jin, D., Jin, Z., Zhou, J.T. and Szolovits, P. (2020), "Is Bert really robust? A strong baseline for natural language attack on text classification and entailment", *AAAI*.

Jin, H., Hu, L., Li, X., Zhang, P., Chen, C., Zhuang, J. and Wang, H. (2024), "Jailbreakzoo: survey, landscapes, and horizons in jailbreaking large language and vision-language models", arXiv preprint arXiv:2407.01599.

Jing, H., Li, H., Hu, W., Hu, Q., Xu, H., Chu, T., Hu, P. and Song, Y. (2025), "MCIP: protecting MCP safety via model contextual integrity protocol", arXiv preprint arXiv:2505.14590.

Joshi, A., Jagatap, G. and Hegde, C. (2021), "Adversarial token attacks on vision transformers", arXiv preprint arXiv:2110.04337.

Ju, T., Wang, Y., Ma, X., Cheng, P., Zhao, H., Wang, Y., Liu, L., Xie, J., Zhang, Z. and Liu, G. (2024), "Flooding spread of manipulated knowledge in LLM-based multi-agent communities", arXiv preprint arXiv:2407.07791.

Kandpal, N., Jagielski, M., Tramèr, F. and Carlini, N. (2023), "Backdoor attacks for in-context learning with language models", *ICML* Workshop.

Karnik, S., Hong, Z.W., Abhangi, N., Lin, Y.C., Wang, T.H., Dupuy, C., Gupta, R. and Agrawal, P. (2025), "Embodied red teaming for auditing robotic foundation models", arXiv preprint arXiv:2411.18676.

Kassem, A.M., Mahmoud, O., Mireshghallah, N., Kim, H., Tsvetkov, Y., Choi, Y., Saad, S. and Rana, S. (2024), "Alpaca against vicuna: using LLMs to uncover memorization of LLMs", arXiv preprint arXiv:2403.04801.

Al-Kaswan, A., Izadi, M. and Van Deursen, A. (2024), "Traces of memorisation in large language models for code", *ICSE*.

Khalili, H., Park, S., Li, V., Bright, B., Payani, A., Kompella, R.R. and Sehatbakhsh, N. (2024), "LightPure: realtime adversarial image purification for mobile devices using diffusion models", *ACM MobiCom*.

Khattak, M.U., Rasheed, H., Maaz, M., Khan, S. and Khan, F.S. (2023), "Maple: multi-modal prompt learning", *CVPR*.

Kim, C., Min, K. and Yang, Y. (2024a), "RACE: robust adversarial concept erasure for secure text-to-image diffusion model", *ECCV*.

Kim, J., Derakhshan, A. and Harris, I.G. (2024b), "Robust safety classifier against jailbreaking attacks: adversarial prompt shield", *WOAH*.

Kim, S., Jung, S., Kim, B., Choi, M., Shin, J. and Lee, J. (2023), "Towards safe self-distillation of internet-scale text-to-image diffusion models", *ICML Workshop*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., *et al*. (2023), "Segment anything", *CVPR*.

Koleva, A., Ringsquandl, M. and Tresp, V. (2023), "Adversarial attacks on tables with entity swap", arXiv preprint arXiv:2309.08650.

Kong, D., Lin, S., Xu, Z., Wang, Z., Li, M., Li, Y., Zhang, Y., Sha, Z., Li, Y., Lin, C., *et al*. (2025), "A survey of LLM-driven AI agent communication: protocols, security risks, and defense countermeasures", arXiv preprint arXiv:2506.19676.

Kong, F., Duan, J., Ma, R., Shen, H.T., Shi, X., Zhu, X. and Xu, K. (2024), "An efficient membership inference attack for the diffusion model by proximal initialization", *ICLR*.

Korbak, T., Balesni, M., Barnes, E., Bengio, Y., Benton, J., Bloom, J., Chen, M., Cooney, A., Dafoe, A., Dragan, A., *et al*. (2025), "Chain of thought monitorability: a new and fragile opportunity for AI safety", arXiv preprint arXiv:2507.11473.

Kou, Z., Pei, S., Tian, Y. and Zhang, X. (2023), "Character as pixels: a controllable prompt adversarial attacking framework for black-box text guided image generation models", *IJCAI*.

Krause, J., Stark, M., Deng, J. and Fei-Fei, L. (2013), "3d object repre-sentations for fine-grained categorization", *ICCVW*.

Krizhevsky, A., Hinton, G., *et al*. (2009), "Learning multiple layers of features from tiny images".

Kumar, A., Agarwal, C., Srinivas, S., Feizi, S. and Lakkaraju, H. (2023), "Certifying llm safety against adversarial prompting", arXiv preprint arXiv:2309.02705.

Kumar, P. *et al*. (2025), "Refusal-trained LLMs are easily jailbroken as browser agents", arXiv preprint arXiv:2410.13886.

Kumari, N., Zhang, B., Wang, S.Y., Shechtman, E., Zhang, R. and Zhu, J.Y. (2023), "Ablating concepts in text-to-image diffusion models", *ICCV*.

Lamparth, M. and Reuel, A. (2024), "Analyzing and editing inner mechanisms of backdoored language models", ACM FAccT.

LangChain. (2025), "Context engineering for agents".

Le, T.N., Nguyen, T.V., Nie, Z., Tran, M.T. and Sugimoto, A. (2019), "Anabranch network for camouflaged object segmentation", *Computer vision and Image Understanding*, Vol. 184, pp. 45-56.

Lee, B.H., Lim, S., Lee, S., Kang, D.U. and Chun, S.Y. (2025a), "Concept pinpoint eraser for text-to-image diffusion models via residual attention gate", arXiv preprint arXiv:2506.22806.

Lee, D. and Tiwari, M. (2024), "Prompt infection: Llm-to-llm prompt injection within multi-agent systems", arXiv preprint arXiv:2410.07283.

Lee, S. *et al*. (2025b), "SafeArena: evaluating the safety of autonomous web agents", arXiv preprint arXiv:2503.04957.

Lei, C.T., Yam, H.M., Guo, Z. and Lau, C.P. (2024), "Instant adversarial purification with adversarial consistency distillation", arXiv preprint arXiv:2408.17064.

Li, B., Xing, H., Huang, C., Qian, J., Xiao, H., Feng, L. and Tian, C. (2024a), "StructuralSleight: automated jailbreak attacks on large language models utilizing uncommon text-encoded structure", arXiv preprint arXiv:2406.08754.

Li, B., Xiao, H. and Tang, L. (2024b), "ASAM: boosting segment anything model with adversarial tuning", *CVPR*.

Li, B., Wei, Y., Fu, Y., Wang, Z., Li, Y., Zhang, J., Wang, R. and Zhang, T. (2025a), "Towards reliable verification of unauthorized data usage in personalized text-to-image diffusion models", *IEEE S&P*.

Li, C., Pang, R., Cao, B., Chen, J., Ma, F., Ji, S. and Wang, T. (2024c), "Watch the Watcher! Backdoor attacks on security-enhancing diffusion models", arXiv preprint arXiv:2406.09669.

Li, H., Shen, C., Torr, P., Tresp, V. and Gu, J. (2024d), "Self-discovering interpretable diffusion latent directions for responsible text-to-image generation", *CVPR*.

Li, H., Han, X., Zhai, Z., Mu, H., Wang, H., Zhang, Z., Geng, Y., Lin, S., Wang, R., Shelmanov, A., *et al*. (2024e), "Libra-Leaderboard: towards responsible AI through a balanced leaderboard of safety and capability", arXiv preprint arXiv:2412.18551.

Li, H., Chen, Y., Zheng, Z., Hu, Q., Chan, C., Liu, H. and Song, Y. (2024f), "Backdoor removal for generative large language models", arXiv preprint arXiv:2405.07667.

Li, J., Xu, N., Wang, H., Chen, X. and Liu, Y. (2024g), "BlueSuffix: reinforced blue teaming for vision-language models against jailbreak attacks", arXiv preprint arXiv:2410.20971.

Li, J., Wu, Z., Ping, W., Xiao, C. and Vydiswaran, V. (2023a), "Defending against insertion-based textual backdoor attacks via attribution", *ACL*.

Li, J., Liu, Y., Liu, C., Shi, L., Ren, X., Zheng, Y., Liu, Y. and Xue, Y. (2024h), "A cross-language investigation into jailbreak attacks in large language models", arXiv preprint arXiv:2401.16765.

Li, J., Dong, J., He, T. and Zhang, J. (2024i), "Towards black-box membership inference attack for diffusion models", arXiv preprint arXiv:2405.20771.

Li, J., Li, D., Savarese, S. and Hoi, S. (2023b), "Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models", *ICML*.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C. and Hoi, S.C.H. (2021), "Align before fuse: vision and language representation learning with momentum distillation", *NeurIPS*.

Li, J. (2022), "Patch vestiges in the adversarial examples against vision transformer can be leveraged for adversarial detection", *AAAI Workshop*.

Li, L., Dong, B., Wang, R., Hu, X., Zuo, W., Lin, D., Qiao, Y. and Shao, J. (2024j), "Salad-bench: a hierarchical and comprehensive safety benchmark for large language models", *ACL*.

Li, L., Guan, H., Qiu, J. and Spratling, M. (2024k), "One prompt word is enough to boost adversarial robustness for pre-trained vision-language models", *CVPR*.

Li, L., Ma, R., Guo, Q., Xue, X. and Qiu, X. (2020), "BERT-Attack: adversarial attack against BERT using BERT", *EMNLP*.

Li, M., Zhao, S., Wang, Q., Wang, K., Zhou, Y., Srivastava, S., Gokmen, C., Lee, T., Li, L.E., Zhang, R., Liu, W., Liang, P., Fei-Fei, L., Mao, J. and Wu, J. (2024l), "Embodied agent interface: benchmarking LLMs for embodied decision making", *NeurIPS*.

Li, M., Ye, Z., Li, Y., Song, A., Zhang, G. and Liu, F. (2025b), "Membership inference attack should move on to distributional statistics for distilled generative models", arXiv preprint arXiv:2502.02970.

Li, Q., Fu, X., Wang, X., Liu, J., Gao, X., Dai, J. and Han, J. (2024m), "Unveiling structural memorization: structural membership inference attack for text-to-image diffusion models", *ACM MM*.

Li, S., Ma, J. and Cheng, M. (2024n), "Invisible backdoor attacks on diffusion models", arXiv preprint arXiv:2406.00816.

Li, S., Yao, L., Zhang, L. and Li, Y. (2024o), "Safety layers in aligned large language models: the key to LLM security", arXiv preprint arXiv:2408.17003.

Li, T., Liu, Q., Pang, T., Du, C., Guo, Q., Liu, Y. and Lin, M. (2024p), "Purifying large language models by ensembling a small language model", arXiv preprint arXiv:2402.14845.

Li, X., Sun, W., Chen, H., Li, Q., Liu, Y., He, Y., Shi, J. and Hu, X. (2024q), "ADBM: adversarial diffusion bridge model for reliable adversarial purification", arXiv preprint arXiv:2408.00315.

Li, X., Yang, Y., Deng, J., Yan, C., Chen, Y., Ji, X. and Xu, W. (2024r), "Safegen: mitigating sexually explicit content generation in text-to-image models", *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 4807-4821.

Li, X., Wang, R., Cheng, M., Zhou, T. and Hsieh, C.J. (2024s), "DrAt-tack: prompt decomposition and reconstruction makes powerful LLM jailbreakers", *EMNLP*.

Li, Y., Li, T., Chen, K., Zhang, J., Liu, S., Wang, W., Zhang, T. and Liu, Y. (2024t), "Badedit: backdooring large language models by model editing", *ICLR*.

Li, Y., Guo, H., Zhou, K., Zhao, W.X. and Wen, J.R. (2024u), "Images are Achilles' heel of alignment: exploiting visual vulnerabilities for jailbreaking multimodal large language models", *ECCV*.

Li, Y., Huang, H., Zhang, J., Ma, X. and Jiang, Y.G. (2024v), "Expose before you defend: unifying and enhancing backdoor defenses via exposed models", arXiv preprint arXiv:2410.19427.

Li, Y., Huang, H., Zhao, Y., Ma, X. and Sun, J. (2024w), "Backdoorllm: a comprehensive benchmark for backdoor attacks on large language models", arXiv preprint arXiv:2408.12798.

Li, Y., Ma, X., He, J., Huang, H. and Jiang, Y.G. (2024x), "Multi-trigger backdoor attacks: more triggers, more threats", arXiv preprint arXiv:2401.15295.

Li, Y., Shao, S., He, Y., Guo, J., Zhang, T., Qin, Z., Chen, P.Y., Backes, M., Torr, P., Tao, D., *et al*. (2025c), "Rethinking data protection in the (generative) artificial intelligence era", arXiv preprint arXiv:2507.03034.

Li, Y. *et al*. (2024y), "ChemSafetyBench: benchmarking LLM safety on chemistry domain", arXiv preprint arXiv:2411.16736.

Li, Y., Xu, Z., Jiang, F., Niu, L., Sahabandu, D., Ramasubramanian, B. and Poovendran, R. (2024z), "CleanGen: mitigating backdoor attacks for generation tasks in large language models", *EMNLP*.

Li, Z., Ren, M., Jiang, F., Li, Q. and Sun, Z. (2024aa), "Improving transferability of adversarial samples via critical region-oriented feature-level attack", *IEEE Transactions on Information Forensics and Security*, Vol. 19, pp. 6650-6664.

Li, Z., Wang, C., Ma, P., Liu, C., Wang, S., Wu, D., Gao, C. and Liu, Y. (2024ab), "On extracting specialized code abilities from large language models: a feasibility study", *ICSE*.

Liang, C., Han, X., Shen, L., Bai, J. and Wong, K.F. (2025a), "Vulnerability-aware alignment: mitigating uneven forgetting in harmful fine-tuning", *ICML*.

Liang, C., Shen, L., Deng, Y., Zhao, X., Liang, B. and Wong, K.F. (2025b), "PEARL: towards permutation-resilient LLMs", *ICLR*.

Liang, C., Wu, X., Hua, Y., Zhang, J., Xue, Y., Song, T., Xue, Z., Ma, R. and Guan, H. (2023), "Adversarial example does good: preventing painting imitation from diffusion models via adversarial examples", *ICML*.

Liang, S., Liang, J., Pang, T., Du, C., Liu, A., Chang, E.C. and Cao, X. (2024a), "Revisiting backdoor attacks against large vision-language models", arXiv preprint arXiv:2406.18844.

Liang, S., Zhu, M., Liu, A., Wu, B., Cao, X. and Chang, E.C. (2024b), "Badclip: dual-embedding guided backdoor attack on multimodal contrastive learning", *CVPR*.

Liang, Z., Ye, Q., Wang, Y., Zhang, S., Xiao, Y., Li, R., Xu, J. and Hu, H. (2024c), "Alignment-aware model extraction attacks on large language models", arXiv preprint arXiv:2409.02718.

Liao, Z., Tang, H., Zhang, O.D., Wang, J. and Yang, D. (2024), "EIA: environmental injection attack on generalist web agents for privacy leakage", arXiv preprint arXiv:2409.11295.

Lin, G., Tanaka, T. and Zhao, Q. (2024a), "Large language model sentinel: Llm agent for adversarial purification", arXiv preprint arXiv:2405.20770.

Lin, S., Li, R., Wang, X., Lin, C., Xing, W. and Han, M. (2024b), "LLMs can be dangerous reasoners: analyzing-based jailbreak attack on large language models", arXiv preprint arXiv:2407.16205.

Lin, S., Hilton, J. and Evans, O. (2022), "TruthfulQA: measuring how models mimic human falsehoods", *ACL*.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L. (2014), "Microsoft coco: common objects in context", *ECCV*.

Liu, A., Zhou, Y., Liu, X., Zhang, T., Liang, S., Wang, J., Pu, Y., Li, T., Zhang, J., Zhou, W., *et al*. (2024a), "Compromising embodied agents with contextual backdoor attacks", arXiv preprint arXiv:2408.02882.

Liu, A., Zhou, Y., Liu, X., Zhang, T., Liang, S., Wang, J., Pu, Y., Li, T., Zhang, J., Zhou, W., Guo, Q. and Tao, D. (2024b), "Compromising embodied agents with contextual backdoor attacks", arXiv preprint arXiv:2408.02882.

Liu, B., Xiao, B., Jiang, X., Cen, S., He, X. and Dou, W. (2023a), "Adversarial attacks on large language model-based system and mitigating strategies: a case study on ChatGPT", *Security and Communication Networks*, Vol. 2023, p. 10.

Liu, D., Yang, M., Qu, X., Zhou, P., Cheng, Y. and Hu, W. (2024c), "A survey of attacks on large vision-language models: resources, advances, and future trends", arXiv preprint arXiv:2407.07403.

Liu, F., Luo, H., Li, Y., Torr, P. and Gu, J. (2024d), "Which model generated this image? A model-agnostic approach for origin attribution", *ECCV*.

Liu, G., Lin, W., Huang, T., Mo, R., Mu, Q. and Shen, L. (2024e), "Targeted vaccine: safety alignment for large language models against harmful fine-tuning via layer-wise perturbation", arXiv preprint arXiv:2410.09760.

Liu, H., Wu, Y., Zhai, S., Yuan, B. and Zhang, N. (2023b), "Riatig: reliable and imperceptible adversarial text-to-image generation with natural prompts", *CVPR*.

Liu, H., Sun, Z. and Mu, Y. (2024f), "Countering personalized text-to-image generation with influence watermarks", *CVPR*.

Liu, H., Cai, C. and Qi, Y. (2023c), "Expanding scope: adapting english adversarial attacks to chinese", *TrustNLP*.

Liu, H., Reiter, M.K. and Gong, N.Z. (2024g), "Mudjacking: patching backdoor vulnerabilities in foundation models", *USENIX Security*.

Liu, L., Guo, Y., Zhang, Y. and Yang, J. (2023d), "Understanding and defending patched-based adversarial attacks for vision transformer", *ICML*.

Liu, Q., Kortylewski, A., Bai, Y., Bai, S. and Yuille, A. (2024h), "Discovering failure modes of text-guided diffusion models via adversarial search", *ICLR*.

Liu, R., Yang, R., Jia, C., Zhang, G., Zhou, D., Dai, A.M., Yang, D. and Vosoughi, S. (2024i), "Training socially aligned language models on simulated social interactions", *ICLR*.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., *et al*. (2024j), "Grounding dino: marrying dino with grounded pre-training for open-set object detection", *ECCV*.

Liu, X., Zhu, Y., Gu, J., Lan, Y., Yang, C. and Qiao, Y. (2024k), "MM-safetybench: a benchmark for safety evaluation of multimodal large language models", *ECCV*.

Liu, X., Xu, N., Chen, M. and Xiao, C. (2024l), "AutoDAN: generating stealthy jailbreak prompts on aligned large language models", *ICLR*.

Liu, X., Yu, Z., Zhang, Y., Zhang, N. and Xiao, C. (2024m), "Automatic and universal prompt injection attacks against large language models", arXiv preprint arXiv:2403.04957.

Liu, X., Jia, X., Xun, Y., Liang, S. and Cao, X. (2024n), "Multimodal un-learnable examples: protecting data against multimodal contrastive learning", *ACMMM*.

Liu, X., Cui, X., Li, P., Li, Z., Huang, H., Xia, S., Zhang, M., Zou, Y. and He, R. (2024o), "Jailbreak attacks and defenses against multimodal generative models: a survey", arXiv preprint arXiv:2411.09259.

Liu, Y. *et al*. (2024p), "Dissecting adversarial robustness of multimodal LM agents", arXiv preprint arXiv:2406.12814.

Liu, Y., Deng, G., Li, Y., Wang, K., Wang, Z., Wang, X., Zhang, T., Liu, Y., Wang, H., Zheng, Y., *et al*. (2023e), "Prompt Injection attack against LLM-integrated Applications", arXiv preprint arXiv:2306.05499.

Liu, Y., Yang, G., Deng, G., Chen, F., Chen, Y., Shi, L., Zhang, T. and Liu, Y. (2024q), "Groot: adversarial testing for generative text-to-image models with tree-based semantic transformation", arXiv preprint arXiv:2402.12100.

Liu, Y., Fan, C., Dai, Y., Chen, X., Zhou, P. and Sun, L. (2024r), "MetaCloak: preventing unauthorized subject-driven text-to-image diffusion-based synthesis via meta-learning", *CVPR*.

Liu, Y., An, J., Zhang, W., Li, M., Wu, D., Gu, J., Lin, Z. and Wang, W. (2024s), "RealEra: semantic-level concept erasure via neighbor-concept mining", arXiv preprint arXiv:2410.09140.

Liu, Y., Li, Z., Backes, M., Shen, Y. and Zhang, Y. (2023f), "Watermarking diffusion model", arXiv preprint arXiv:2305.12502.

Liu, Y., Jia, Y., Geng, R., Jia, J. and Gong, N.Z. (2024t), "Formalizing and benchmarking prompt injection attacks and defenses", *USENIX Security*, pp. 1831-1847.

Liu, Z., Shen, B., Lin, Z., Wang, F. and Wang, W. (2023g), "Maximum entropy loss, the silver bullet targeting backdoor attacks in pre-trained language models", *ACL*.

Liu, Z., Chen, K., Zhang, Y., Han, J., Hong, L., Xu, H., Li, Z., Yeung, D.Y. and Kwok, J. (2024u), "Implicit concept removal of diffusion models", arXiv preprint arXiv:2310.05873.

Liu, Z., Wang, Z., Xu, L., Wang, J., Song, L., Wang, T., Chen, C., Cheng, W. and Bian, J. (2024v), "Protecting your llms with information bottleneck", *NeurIPS*.

Liu, Z., Luo, P., Wang, X. and Tang, X. (2015), "Deep learning face attributes in the wild", *ICCV*.

Long, J., Xu, Z., Jiang, T., Yao, W., Jia, S., Ma, C. and Chen, X. (2025), "Robust SAM: on the adversarial robustness of vision foundation models", *AAAI*.

Lovisotto, G., Finnie, N., Munoz, M., Mummadi, C.K. and Metzen, J.H. (2022), "Give me your attention: dot-product attention considered harmful for adversarial patch robustness", *CVPR*.

Lu, D., Pang, T., Du, C., Liu, Q., Yang, X. and Lin, M. (2024a), "Test-time backdoor attacks on multimodal large language models", arXiv preprint arXiv:2402.08577.

Lu, D., Wang, Z., Wang, T., Guan, W., Gao, H. and Zheng, F. (2023), "Set-level guidance attack: boosting adversarial transferability of vision-language pre-training models", *ICCV*.

Lu, J., Yang, X. and Wang, X. (2024b), "Unsegment anything by simulating deformation", *CVPR*.

Lu, L., Yan, H., Yuan, Z., Shi, J., Wei, W., Chen, P.Y. and Zhou, P. (2024c), "AutoJailbreak: exploring jailbreak attacks and defenses through a dependency lens", arXiv preprint arXiv:2406.03805.

Lu, S., Wang, Z., Li, L., Liu, Y. and Kong, A.W.K. (2024d), "Mace: mass concept erasure in diffusion models", *CVPR*.

Lu, X., Huang, Z., Li, X., Ji, X. and Xu, W. (2025), "POEX: understanding and mitigating policy executable jailbreak attacks against embodied AI", arXiv preprint arXiv:2412.16633.

Luo, H., Gu, J., Liu, F. and Torr, P. (2024a), "An image is worth 1000 lies: transferability of adversarial images across prompts on vision-language models", *ICLR*.

Luo, H., Zhang, T., Chuang, Y.S., Gong, Y., Kim, Y., Wu, X., Meng, H. and Glass, J. (2023), "Search augmented instruction learning", *EMNLP*.

Luo, L., Wang, X., Zi, B., Zhao, S. and Ma, X. (2024b), "Adversarial prompt distillation for vision-language models", arXiv preprint arXiv:2411.15244.

Luo, W., Ma, S., Liu, X., Guo, X. and Xiao, C. (2024c), "Jailbreakv-28k: a benchmark for assessing the robustness of multimodal large language models against jailbreak attacks", *COLM*.

Lv, H., Wang, X., Zhang, Y., Huang, C., Dou, S., Ye, J., Gui, T., Zhang, Q. and Huang, X. (2024), "Codechameleon: personalized encryption framework for jailbreaking large language models", arXiv preprint arXiv:2402.16717.

Lv, P., Ma, H., Zhou, J., Liang, R., Chen, K., Zhang, S. and Yang, Y. (2023), "DBIA: data-free backdoor attack against transformer networks", *ICME*.

Lyu, M., Yang, Y., Hong, H., Chen, H., Jin, X., He, Y., Xue, H., Han, J. and Ding, G. (2024), "One-dimensional adapter to rule them all: concepts diffusion models and erasing applications", *CVPR*.

Ma, J., Cao, A., Xiao, Z., Zhang, J., Ye, C. and Zhao, J. (2024a), "Jail-breaking prompt attack: a controllable adversarial attack against diffusion models", arXiv preprint arXiv:2404.02928.

Ma, S., Luo, W., Wang, Y., Liu, X., Chen, M., Li, B. and Xiao, C. (2024b), "Visual-RolePlay: universal jailbreak attack on multimodal large language models via role-playing image character", arXiv preprint arXiv:2405.20773.

Ma, W., Li, Y., Jia, X. and Xu, W. (2023), "Transferable adversarial attack for both vision transformers and convolutional networks via momentum integrated gradients", *ICCV*.

Ma, X., Ruan, Z., Chen, Y., Jin, R. and Zhang, Z. (2024c), "Caution for the environment: multimodal agents are susceptible to environmental distractions", arXiv preprint arXiv:2408.02544.

Ma, Z., Jia, G., Qi, B. and Zhou, B. (2024d), "Safe-SD: safe and traceable stable diffusion with text prompt trigger for invisible generative watermarking", *ACM MM*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D. and Vladu, A. (2018), "Towards deep learning models resistant to adversarial attacks", *ICLR*.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M. and Vedaldi, A. (2013), "Fine-grained visual classification of aircraft", arXiv preprint arXiv:1306.5151.

Makrigiorgos, A., Shafti, A., Harston, A., Gerard, J. and Faisal, A.A. (2019), "Human visual attention prediction boosts learning & performance of autonomous driving agents", arXiv preprint arXiv:1909.05003.

Mao, C., Geng, S., Yang, J., Wang, X. and Vondrick, C. (2023), "Understanding zero-shot adversarial robustness for large-scale models", *ICLR*.

Mao, J., Meng, F., Duan, Y., Yu, M., Jia, X., Fang, J., Liang, Y., Wang, K. and Wen, Q. (2025), "Agentsafe: safeguarding large language model-based multi-agent systems via hierarchical data management", arXiv preprint arXiv:2503.04392.

Matsumoto, T., Miura, T. and Yanai, N. (2023), "Membership inference attacks against diffusion models", *SPW*.

Maus, N., Chao, P., Wong, E. and Gardner, J.R. (2023), "Black box adversarial prompting for foundation models", *NFAML*.

Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y. and Ermon, S. (2021), "Sdedit: guided image synthesis and editing with stochastic differential equations", arXiv preprint arXiv:2108.01073.

Meng, K., Sharma, A.S., Andonian, A., Belinkov, Y. and Bau, D. (2023), "Mass-editing memory in a transformer", *ICLR*.

Meng, Z., Peng, B., Jin, X., Lyu, Y., Wang, W. and Dong, J. (2025), "Concept corrector: erase concepts on the fly for text-to-image diffusion models", arXiv preprint arXiv:2502.16368.

Millière, R. (2022), "Adversarial attacks on image generation with made-up words", arXiv preprint arXiv:2208.04135.

Min, N.M., Pham, L.H., Li, Y. and Sun, J. (2024a), "CROW: eliminating backdoors from large language models via internal consistency regularization", arXiv preprint arXiv:2411.12768.

Min, R., Li, S., Chen, H. and Cheng, M. (2024b), "A watermark-conditioned diffusion model for ip protection", 2024.

Mo, Y., Huang, H., Li, M., Li, A. and Wang, Y. (2024), "TERD: a unified framework for safeguarding diffusion models against backdoors", *ICML*.

Mo, Y., Wu, D., Wang, Y., Guo, Y. and Wang, Y. (2022), "When adversarial training meets vision transformers: recipes from training to architecture", *NeurIPS*.

Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O. and Frossard, P. (2017), "Universal adversarial perturbations", *CVPR*.

More, Y., Ganesh, P. and Farnadi, G. (2024), "Towards more realistic extraction attacks: an adversarial perspective", arXiv preprint arXiv:2407.02596.

Naseer, M., Ranasinghe, K., Khan, S., Khan, F.S. and Porikli, F. (2021), "On improving adversarial transferability of vision transformers", arXiv preprint arXiv:2106.04169.

Naseh, A., Roh, J., Bagdasaryan, E. and Houmansadr, A. (2024), "Injecting bias in text-to-image models via composite-trigger backdoors", arXiv preprint arXiv:2406.15213.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A.F., Ippolito, D., Choquette-Choo, C.A., Wallace, E., Tramèr, F. and Lee, K. (2023), "Scalable extraction of training data from (production) language models", arXiv preprint arXiv:2311.17035.

Navaneet, K., Koohpayegani, S.A., Sleiman, E. and Pirsiavash, H. (2024), "SlowFormer: adversarial attack on compute and energy consumption of efficient vision transformers", *CVPR*.

Ni, Z., Ye, R., Wei, Y., Xiang, Z., Wang, Y. and Chen, S. (2024), "Physical backdoor attack can jeopardize driving with vision-large-language models", *ICML Workshop*.

Ni, Z., Wei, L., Li, J., Tang, S., Zhuang, Y. and Tian, Q. (2023), "Degeneration-tuning: using scrambled grid shield unwanted concepts from stable diffusion", *ACM MM*.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A. and Anandkumar, A. (2022), "Diffusion models for adversarial purification", *ICML*.

Nikzad, N., Liao, Y., Gao, Y. and Zhou, J. (2025), "SATA: spatial autocorrelation token analysis for enhancing the robustness of vision transformers", *CVPR*.

Nilsback, M.E. and Zisserman, A. (2008), "Automated flower classification over a large number of classes", *ICVGIP*.

Niu, Z., Ren, H., Gao, X., Hua, G. and Jin, R. (2024), "Jailbreaking attack against multimodal large language model", arXiv preprint arXiv:2402.02309.

Noever, D.A. and Noever, S.E.M. (2021), "Reading isn't believing: adversarial attacks on multi-modal neurons", arXiv preprint arXiv:2103.10480.

OpenAI. (2024), "Introducing OpenAI o1".

Orgad, H., Kawar, B. and Belinkov, Y. (2023), "Editing implicit assumptions in text-to-image diffusion models", *ICCV*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., *et al*. (2022), "Training language models to follow instructions with human feedback", *NeurIPS*.

Pan, Z., Yao, Y., Liu, G., Shen, B., Zhao, H.V., Kompella, R.R. and Liu, S. (2024), "From Trojan horses to castle walls: unveiling bilateral backdoor effects in diffusion models", *NeurIPS Workshop*.

Pang, X., Tang, S., Ye, R., Xiong, Y., Zhang, B., Wang, Y. and Chen, S. (2024), "Self-alignment of large language models via monopolylogue-based social scene simulation", arXiv preprint arXiv:2402.05699.

Pang, Y. and Wang, T. (2023), "Black-box membership inference attacks against fine-tuned diffusion models", arXiv preprint arXiv:2312.08207.

Pang, Y., Wang, T., Kang, X., Huai, M. and Zhang, Y. (2023), "White-box membership inference attacks against diffusion models", arXiv preprint arXiv:2308.06405.

Parkhi, O.M., Vedaldi, A., Zisserman, A. and Jawahar, C. (2012), "Cats and dogs", *CVPR*.

Pathmanathan, P., Chakraborty, S., Liu, X., Liang, Y. and Huang, F. (2024), "Is poisoning a real threat to LLM alignment? maybe more so than you think", *ICML Workshop*.

Pedro, R., Castro, D., Carreira, P. and Santos, N. (2023), "From prompt injections to sql injection attacks: how protected is your llm-integrated web application?", arXiv preprint arXiv:2308.01990.

Peng, S., Chen, Y., Wang, C. and Jia, X. (2023), "Protecting the intellectual property of diffusion models by the watermark diffusion process", arXiv preprint arXiv:2306.03436*.3*.

Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N. and Irving, G. (2022), "Red teaming language models with language models", *EMNLP*.

Perez, F. and Ribeiro, I. (2022), "Ignore previous prompt: attack techniques for language models", *NeurIPS Workshop*.

Pi, R., Han, T., Xie, Y., Pan, R., Lian, Q., Dong, H., Zhang, J. and Zhang, T. (2024), "MLLM-protector: ensuring MLLM's safety without hurting performance", *EMNLP*.

Piet, J., Alrashed, M., Sitawarin, C., Chen, S., Wei, Z., Sun, E., Alomair, B. and Wagner, D. (2023), "Jatmo: prompt injection defense by task-specific finetuning", arXiv preprint arXiv:2312.17673.

Pinkney, J.N.M. (2022), "Pokemon BLIP captions".

Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hock-enmaier, J. and Lazebnik, S. (2015), "Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models", *ICCV*.

Qi, X., Huang, K., Panda, A., Henderson, P., Wang, M. and Mittal, P. (2024a), "Visual adversarial examples jailbreak aligned large language models", *AAAI*.

Qi, X., Panda, A., Lyu, K., Ma, X., Roy, S., Beirami, A., Mittal, P. and Henderson, P. (2025), "Safety alignment should be made more than just a few tokens deep", *ICLR*.

Qi, X., Zeng, Y., Xie, T., Chen, P.Y., Jia, R., Mittal, P. and Henderson, P. (2023), "Fine-tuning aligned language models compromises safety, even when users do not intend to!" arXiv preprint arXiv:2310.03693.

Qi, Z., Zhang, H., Xing, E., Kakade, S. and Lakkaraju, H. (2024b), "Follow my instruction and spill the beans: scalable data extraction from retrieval-augmented generation systems", arXiv preprint arXiv:2402.17840.

Qiang, Y., Zhou, X., Zade, S.Z., Roshani, M.A., Zytko, D. and Zhu, D. (2024), "Learning to poison large language models during instruction tuning", arXiv preprint arXiv:2402.13459.

Qraitem, M., Tasnim, N., Saenko, K. and Plummer, B.A. (2024), "Vision-llms can fool themselves with self-generated typographic attacks", arXiv preprint arXiv:2402.00626.

Qu, B. *et al*. (2024), "Very simple membership inference and synthetic identification in denoising diffusion models", PhD thesis, Vanderbilt University.

Qu, Y., Shen, X., He, X., Backes, M., Zannettou, S. and Zhang, Y. (2023), "Unsafe diffusion: on the generation of unsafe images and hateful memes from text-to-image models", *CCS*.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., *et al*. (2021), "Learning transferable visual models from natural language supervision", *ICML*.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S. and Finn, C. (2024), "Direct preference optimization: your language model is secretly a reward model", *NeurIPS*.

Rahman, S., Jiang, L., Shiffer, J., Liu, G., Issaka, S., Parvez, M.R., Palangi, H., Chang, K.W., Choi, Y. and Gabriel, S. (2025), "X-teaming: multi-turn jailbreaks and defenses with adaptive multi-agents", arXiv preprint arXiv:2504.13203.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. and Chen, M. (2022), "Hierarchical text-conditional image generation with CLIP latents", arXiv preprint arXiv:2204.06125.

Rando, J., Paleka, D., Lindner, D., Heim, L. and Tramèr, F. (2022), "Red-teaming the stable diffusion safety filter", arXiv preprint arXiv:2210.04610.

Recht, B., Roelofs, R., Schmidt, L. and Shankar, V. (2019), "Do imagenet classifiers generalize to imagenet?", *ICML*.

Ren, J., Li, Y., Zeng, S., Xu, H., Lyu, L., Xing, Y. and Tang, J. (2024), "Unveiling and mitigating memorization in text-to-image diffusion models through cross attention", *ECCV*.

Ren, Y., Zhao, Z., Lin, C., Yang, B., Zhou, L., Liu, Z. and Shen, C. (2025), "Improving adversarial transferability on vision transformers via forward propagation refinement", *CVPR*.

Rezaei, A., Akbari, M., Alvar, S.R., Fatemi, A. and Zhang, Y. (2024), "LaWa: using latent space for in-generation image watermarking", *ECCV*.

Robey, A., Ravichandran, Z., Kumar, V., Hassani, H. and Pappas, G.J. (2024a), "Jailbreaking LLM-controlled robots", arXiv preprint arXiv:2410.13691.

Robey, A., Ravichandran, Z., Kumar, V., Hassani, H. and Pappas, G.J. (2024b), "Jailbreaking LLM-controlled robots", arXiv preprint arXiv:2410.13691.

Robey, A., Wong, E., Hassani, H. and Pappas, G.J. (2023), "Smoothllm: defending large language models against jailbreaking attacks", arXiv preprint arXiv:2310.03684.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B. (2022), "High-resolution image synthesis with latent diffusion models", *CVPR*.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M. and Aberman, K. (2023), "Dreambooth: fine tuning text-to-image diffusion models for subject-driven generation", *CVPR*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C. and Fei-Fei, L. (2015), "Imagenet large scale visual recognition challenge", *International Journal of Computer Vision*, Vol. 115, pp. 211-252.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B, Salimans, T, *et al*. (2022), "Photorealistic text-to-image diffusion models with deep language understanding", *NeurIPS*.

Saleh, B. and Elgammal, A. (2015), "Large-scale classification of fine-art paintings: learning the right metric on the right feature", arXiv preprint arXiv:1505.00855.

Schlarmann, C. and Hein, M. (2023), "On the adversarial robustness of multi-modal foundation models", *ICCV*.

Schlarmann, C., Singh, N.D., Croce, F. and Hein, M. (2024), "Robust CLIP: unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models", *ICML*.

Schramowski, P., Brack, M., Deiseroth, B. and Kersting, K. (2023), "Safe latent diffusion: mitigating inappropriate degeneration in diffusion models", *CVPR*.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., *et al*. (2022), "Laion-5b: an open large-scale dataset for training next generation image-text models", *NeurIPS*.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J. and Komatsuzaki, A. (2021), "Laion-400m: open dataset of clip-filtered 400 million image-text pairs", arXiv preprint arXiv:2111.02114.

Shah, M., Chen, X., Rohrbach, M. and Parikh, D. (2019), "Cycle-consistency for robust visual question answering", *CVPR*

Shao, L. *et al*. (2024a), "SALAD-bench: a hierarchical and comprehensive safety benchmark for large language models", arXiv preprint arXiv:2402.05044.

Shao, Y., Li, T., Shi, W., Liu, Y. and Yang, D. (2024b), "Privacylens: evaluating privacy norm awareness of language models in action", *NeurIPS*.

Shao, Z., Liu, H., Mu, J. and Gong, N.Z. (2024c), "Making LLMs vulnerable to prompt injection via poisoning alignment", arXiv preprint arXiv:2410.14827.

Sharma, R.K., Gupta, V. and Grossman, D. (2024a), "Defending language models against image-based prompt attacks via user-provided specifications", *IEEE SPW*.

Sharma, R.K., Gupta, V. and Grossman, D. (2024b), "SPML: a DSL for defending language models against prompt attacks", arXiv preprint arXiv:2402.11755.

Shayegani, E., Dong, Y. and Abu-Ghazaleh, N. (2023), "Jailbreak in pieces: compositional adversarial attacks on multi-modal language models", *ICLR*.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G. and Dean, J. (2017), "Outrageously large neural networks: the sparsely-gated mixture-of-experts layer", *ICLR*.

Shen, X., Chen, Z., Backes, M., Shen, Y. and Zhang, Y. (2024a), "'Do anything now': characterizing and evaluating in-the-wild jail-break prompts on large language models" *ACM CCS*.

Shen, Y., Li, Z. and Wang, G. (2024b), "Practical region-level attack against segment anything models", *CVPR*.

Sheshadri, A., Hughes, J., Michael, J., Mallen, A., Jose, A., Roger, F., *et al*. (2025), "Why do some language models fake alignment while others don't?", arXiv preprint arXiv:2506.18032.

Shi, J., Yuan, Z., Liu, Y., Huang, Y., Zhou, P., Sun, L. and Gong, N.Z. (2024), "Optimization-based prompt injection attack to llm-as-a-judge", *ACM SIGSAC CCS*.

Shi, Y., Han, Y., Tan, Y.A. and Kuang, X. (2022), "Decision-based black-box attack against vision transformers via patch-wise adversarial removal", *NeurIPS*.

Shirasaka, M., Matsushima, T., Tsunashima, S., Ikeda, Y., Horo, A., Ikoma, S., Tsuji, C., Wada, H., Omija, T., Komukai, D., Matsuo, Y. and Iwasawa, Y. (2024), "Self-recovery prompting: promptable general purpose service robot system with foundation models and self-recovery", *ICRA*, doi: 10.1109/icra57147.2024.10611640.

Shlomov, S. *et al*. (2024), "ST-WebAgentBench: a benchmark for evaluating safety and trustworthiness in web agents", arXiv preprint arXiv:2410.06703.

Slattery, P., Saeri, A.K., Grundy, E.A., Graham, J., Noetel, M., Uuk, R., Dao, J., Pour, S., Casper, S. and Thompson, N. (2024), "The AI risk repository: a comprehensive meta-review, database, and taxonomy of risks from artificial intelligence", arXiv preprint arXiv:2408.12622.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J. and Goldstein, T. (2023), "Understanding and mitigating copying in diffusion models", *NeurIPS*.

Song, C., Ma, L., Zheng, J., Liao, J., Kuang, H. and Yang, L. (2024a), "Audit-llm: multi-agent collaboration for log-based insider threat detection", arXiv preprint arXiv:2408.08902.

Song, J., Meng, C. and Ermon, S. (2021a), "Denoising diffusion implicit models", *ICLR*.

Song, K., Lai, H., Pan, Y. and Yin, J. (2024b), "MimicDiffusion: purifying adversarial perturbation via mimicking clean diffusion model", *CVPR*.

Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B. (2021b), "Score-based generative modeling through stochastic differential equations", *ICLR*.

Soomro, K. (2012), "UCF101: a dataset of 101 human actions classes from videos in the wild", arXiv preprint arXiv:1212.0402.

Souly, A., Lu, Q., Bowen, D., Trinh, T., Hsieh, E., Pandey, S., Abbeel, P., Svegliato, J., Emmons, S., Watkins, O. and Toyer, S. (2024), "A StrongREJECT for empty jailbreaks", *ICLR Workshop*.

Standen, M., Kim, J. and Szabo, C. (2025), "Adversarial machine learning attacks and defences in multi-agent reinforcement learning", *ACM Computing Surveys*, Vol. 57 No. 5, pp. 1-35.

Stewart, G.W. (1993), "On the early history of the singular value decomposition", *SIAM review*, Vol. 35, pp. 551-566.

Struppek, L., Hintersdorf, D., Friedrich, F., Schramowski, P., Kersting, K. and Brack, M. (2023a), "Exploiting cultural biases via homoglyphs in text-to-image synthesis", *Journal of Artificial Intelligence Research*, Vol. 78, pp. 1017-1068.

Struppek, L., Hintersdorf, D. and Kersting, K. (2023b), "Rickrolling the artist: injecting backdoors into text encoders for text-to-image synthesis", *ICCV*.

Subramanya, A., Koohpayegani, S.A., Saha, A., Tejankar, A. and Pirsiavash, H. (2024), "A closer look at robustness of vision transformers to backdoor attacks", *WACV*.

Subramanya, A., Saha, A., Koohpayegani, S.A., Tejankar, A. and Pirsiavash, H. (2022), "Backdoor attacks on vision transformers", arXiv preprint arXiv:2206.08477.

Sui, Y., Phan, H., Xiao, J., Zhang, T., Tang, Z., Shi, C., Wang, Y., Chen, Y. and Yuan, B. (2024), "DisDet: exploring detectability of backdoor attack on diffusion models", arXiv preprint arXiv:2402.02739.

Sun, G., Zhan, X., Feng, S., Woodland, P.C. and Such, J. (2025), "CASE-bench: context-aware SafEty benchmark for large language models", arXiv preprint arXiv:2501.14940.

Sun, H., Zhang, Z., Deng, J., Cheng, J. and Huang, M. (2023a), "Safety assessment of chinese large language models", arXiv preprint arXiv:2304.10436.

Sun, S., Nwodo, K., Sugrim, S., Stavrou, A. and Wang, H. (2024a), "ViT-Guard: attention-aware detection against adversarial examples for vision transformer", arXiv preprint arXiv:2409.13828.

Sun, X., Li, X., Meng, Y., Ao, X., Lyu, L., Li, J. and Zhang, T. (2023b), "Defending against backdoor attacks in natural language generation", *AAAI*.

Sun, Y., Zhang, H., Zhang, T., Ma, X. and Jiang, Y.G. (2024b), "UnSeg: one universal unlearnable example generator is enough against all image segmentation", *NeurIPS*.

Sun, Z., Shen, Y., Zhou, Q., Zhang, H., Chen, Z., Cox, D., Yang, Y. and Gan, C. (2024c), "Principle-driven self-alignment of language models from scratch with minimal human supervision", *NeurIPS*.

Sur, I., Sikka, K., Walmer, M., Koneripalli, K., Roy, A., Lin, X., Di-vakaran, A. and Jha, S. (2023), "Tijo: trigger inversion with joint optimization for defending multimodal backdoored models", *ICCV*.

Tan, Z., Zhao, C., Moraffah, R., Li, Y., Kong, Y., Chen, T. and Liu, H. (2024), "The wolf within: covert injection of malice into MLLM societies via an MLLM operative", arXiv preprint arXiv:2402.14859.

Tang, R.R., Yuan, J., Li, Y., Liu, Z., Chen, R. and Hu, X. (2023a), "Setting the trap: capturing and defeating backdoors in pretrained language models through honeypots", *NeurIPS*.

Tang, S., Wu, Z.S., Aydore, S., Kearns, M. and Roth, A. (2023b), "Membership inference attacks on diffusion models via quantile regression", arXiv preprint arXiv:2312.05140.

Tao, X., Zhong, S., Li, L., Liu, Q. and Kong, L. (2024), "ImgTrojan: jailbreaking vision-language models with ONE image", arXiv preprint arXiv:2403.02910.

Teng, M., Xiaojun, J., Ranjie, D., Xinfeng, L., Yihao, H., Zhixuan, C., Yang, L. and Wenqi, R. (2024), "Heuristic-induced multimodal risk distribution jailbreak attack for multimodal large language models", arXiv preprint arXiv:2412.05934.

Tian, Y., Yang, X., Zhang, J., Dong, Y. and Su, H. (2023), "Evil geniuses: delving into the safety of llm-based agents", arXiv preprint arXiv:2311.11855.

Titus, L.M. (2024), "Does ChatGPT have semantic understanding? a problem with the statistics-of-occurrence strategy", *Cognitive Systems Research*, Vol. 83, p. 101174.

Tomilin, T., Fang, M. and Pechenizkiy, M. (2025), "HASARD: a bench-mark for vision-based safe reinforcement learning in embodied agents", *ICLR*.

Tong, T., Xu, J., Liu, Q. and Chen, M. (2024), "Securing multi-turn conversational language models against distributed backdoor triggers", arXiv preprint arXiv:2407.04151.

Toyer, S., Watkins, O., Mendes, E.A., Svegliato, J., Bailey, L., Wang, T., Ong, I., Elmaaroufi, K., Abbeel, P., Darrell, T., *et al*. (2023), "Tensor trust: interpretable prompt injection attacks from an online game", arXiv preprint arXiv:2311.01011.

Truong, V.T., Dang, L.B. and Le, L.B. (2024), "Attacks and defenses for generative diffusion models: a comprehensive survey", arXiv preprint arXiv:2408.03400.

Truong, V.T. and Le, L.B. (2024), "PureDiffusion: using backdoor to counter backdoor in generative diffusion models", arXiv preprint arXiv:2409.13945.

Tsai, Y.L., Hsu, C.Y., Xie, C., Lin, C.H., Chen, J.Y., Li, B., Chen, P.Y., Yu, C.M. and Huang, C.Y. (2024), "Ring-A-Bell! How reliable are concept removal methods for diffusion models?", *ICLR*.

Tu, H., Cui, C., Wang, Z., Zhou, Y., Zhao, B., Han, J., Zhou, W., Yao, H. and Xie, C. (2024), "How many unicorns are in this image? A safety evaluation benchmark for vision llms", *ECCV*.

Van Le, T., Phung, H., Nguyen, T.H., Dao, Q., Tran, N.N. and Tran, A. (2023), "Anti-dreambooth: protecting users from personalized text-to-image synthesis", *ICCV*.

Vice, J., Akhtar, N., Hartley, R. and Mian, A. (2024), "Bagm: a backdoor attack for manipulating text-to-image generative models", *IEEE Transactions on Information Forensics and Security*, Vol. 19, pp. 4865-4880.

Vijayvargiya, S., Soni, A.B., Zhou, X., Wang, Z.Z., Dziri, N., Neubig, G. and Sap, M. (2025), "OpenAgentSafety: a comprehensive frame-work for evaluating real-world AI agent safety", arXiv preprint arXiv:2507.06134.

Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J. and Beu-tel, A. (2024), "The instruction hierarchy: training llms to prioritize privileged instructions", arXiv preprint arXiv:2404.13208.

Wang, B., Xu, C., Wang, S., Gan, Z., Cheng, Y., Gao, J., Awadallah, A.H. and Li, B. (2021), "Adversarial glue: a multi-task benchmark for robustness evaluation of language models", *NeurIPS*.

Wang, F., Wan, X., Sun, R., Chen, J. and Arık, S.Ö. (2024a), "Astute rag: overcoming imperfect retrieval augmentation and knowledge conflicts for large language models", arXiv preprint arXiv:2410.07176.

Wang, F., Tan, Z., Wei, T., Wu, Y. and Huang, Q. (2024b), "SimAC: a simple anti-customization method for protecting face privacy against text-to-image synthesis of diffusion models", *CVPR*.

Wang, H., Zhang, A., Nguyen, D.T., Sun, J., Chua, T.S., *et al*. (2024c), "Ali-agent: assessing LLMs' alignment with human values via agent-based evaluation", *NeurIPS*.

Wang, H., Dong, K., Zhu, Z., Qin, H., Liu, A., Fang, X., Wang, J. and Liu, X. (2024d), "Transferable multimodal attack on vision-language pre-training models", *IEEE S&P*.

Wang, H., Ge, S., Lipton, Z. and Xing, E.P. (2019), "Learning robust global representations by penalizing local predictive power", *NeurIPS*.

Wang, H., Shen, Q., Tong, Y., Zhang, Y. and Kawaguchi, K. (2024e), "The stronger the diffusion model, the easier the backdoor: data poisoning to induce copyright breaches without adjusting finetuning pipeline", *NeurIPS Workshop*.

Wang, H., Zhang, R., Wang, J., Li, M., Huang, Y., Wang, D. and Wang, Q. (2024f), "From allies to adversaries: manipulating llm tool-calling through adversarial injection", arXiv preprint arXiv:2412.10198.

Wang, J., Liu, Z., Park, K.H., Jiang, Z., Zheng, Z., Wu, Z., Chen, M. and Xiao, C. (2023a), "Adversarial demonstration attacks on large language models", arXiv preprint arXiv:2305.14950.

Wang, J., Wu, F., Li, W., Pan, J., Suh, E., Mao, Z.M., Chen, M. and Xiao, C. (2024g), "FATH: authentication-based test-time defense against indirect prompt injection attacks", arXiv preprint arXiv:2410.21492.

Wang, K., Zhang, G., Zhou, Z., Wu, J., Yu, M., Zhao, S., Yin, C., Fu, J., Yan, Y., Luo, H., *et al*. (2025a), "A comprehensive survey in llm (-agent) full stack safety: data, training and deployment", arXiv preprint arXiv:2504.15585.

Wang, N., Yan, Z., Li, W., Ma, C., Chen, H. and Xiang, T. (2025b), "Advancing embodied agent security: from safety benchmarks to input moderation", arXiv preprint arXiv:2504.15699.

Wang, P., Zhang, D., Li, L., Tan, C., Wang, X., Ren, K., Jiang, B. and Qiu, X. (2024h), "Inferaligner: inference-time alignment for harm-lessness through cross-model guidance", *EMNLP*.

Wang, R., Li, J., Wang, Y., Wang, B., Wang, X., Teng, Y., Wang, Y., Ma, X. and Jiang, Y.G. (2025c), "IDEATOR: jailbreaking and benchmarking large vision-language models using themselves", *ICCV*.

Wang, R., Ma, X., Zhou, H., Ji, C., Ye, G. and Jiang, Y.G. (2024i), "White-box multimodal jailbreaks against large vision-language models", *ACM MM*.

Wang, S., Zhang, J., Yuan, Z. and Shan, S. (2024j), "Pre-trained model guided fine-tuning for zero-shot adversarial robustness", *CVPR*.

Wang, T., Han, C., Liang, J.C., Yang, W., Liu, D., Zhang, L.X., Wang, Q., Luo, J. and Tang, R. (2025d), "Exploring the adversarial vulnerabilities of vision-language-action models in robotics", arXiv preprint arXiv:2411.13587.

Wang, X., Zhao, Z. and Larson, M. (2025e), "Typographic attacks in a multi-image setting", *NAACL*.

Wang, X., Chen, K., Ma, X., Chen, Z., Chen, J. and Jiang, Y.G. (2024k), "AdvQDet: detecting query-based adversarial attacks with adver-sarial contrastive prompt tuning", *ACM MM*.

Wang, X., Chen, K., Zhang, J., Chen, J. and Ma, X. (2025f), "TAPT: test-time adversarial prompt tuning for robust inference in vision-language models", *CVPR*.

Wang, X., Ji, Z., Ma, P., Li, Z. and Wang, S. (2023b), "InstructTA: instruction-tuned targeted attack for large vision-language models", arXiv preprint arXiv:2312.01886.

Wang, X., Wu, D., Ji, Z., Li, Z., Ma, P., Wang, S., Li, Y., Liu, Y., Liu, N. and Rahmel, J. (2024l), "SelfDefend: LLMs can defend them-selves against jailbreaking in a practical manner", arXiv preprint arXiv:2406.05498.

Wang, Y., Shi, P. and Zhang, H. (2023c), "Gradient-based word substi-tution for obstinate adversarial examples generation in language models", arXiv preprint arXiv:2307.12507.

Wang, Y., Huang, T., Shen, L., Yao, H., Luo, H., Liu, R., Tan, N., Huang, J. and Tao, D. (2025g), "Panacea: mitigating harmful fine-tuning for large language models via post-fine-tuning perturbation", arXiv preprint arXiv:2501.18100.

Wang, Y., Xue, D., Zhang, S. and Qian, S. (2024m), "BadAgent: inserting and activating backdoor attacks in LLM agents", *ACL*.

Wang, Y., Shi, Z., Bai, A. and Hsieh, C.J. (2024n), "Defending LLMs against jailbreaking attacks via backtranslation", *ACL*.

Wang, Y., Teng, Y., Huang, K., Lyu, C., Zhang, S., Zhang, W., Ma, X., Jiang, Y.G., Qiao, Y. and Wang, Y. (2024o), "Fake alignment: are LLMs really aligned well?", NAACL.

Wang, Y., Hu, W., Dong, Y., Zhang, H., Su, H. and Hong, R. (2023d), "Exploring transferability of multimodal adversarial samples for vision-language pre-training models with contrastive learning", arXiv preprint arXiv:2308.12636.

Wang, Y., Liu, X., Li, Y., Chen, M. and Xiao, C. (2024p), "Adashield: safeguarding multimodal large language models from structure-based attack via adaptive shield prompting", *ECCV*.

Wang, Y., Li, H., Han, X., Nakov, P. and Baldwin, T. (2024q), "Do-not-answer: a dataset for evaluating safeguards in llms", *EACL*.

Wang, Y., Wang, J., Yin, Z., Gong, R., Wang, J., Liu, A. and Liu, X. (2022a), "Generating transferable adversarial examples against vision transformers", *ACM MM*.

Wang, Z., Han, Z., Chen, S., Xue, F., Ding, Z., Xiao, X., Tresp, V., Torr, P. and Gu, J. (2024r), "Stop reasoning! When multimodal LLM with chain-of-thought reasoning meets adversarial image", *COLM*.

Wang, Z., Li, X., Zhu, H. and Xie, C. (2024s), "Revisiting adversarial training at scale", *CVPR*.

Wang, Z., Chen, C., Lyu, L., Metaxas, D.N. and Ma, S. (2023e), "Diagnosis: detecting unauthorized data usages in text-to-image diffusion models", *ICLR*.

Wang, Z., Sehwag, V., Chen, C., Lyu, L., Metaxas, D.N. and Ma, S. (2024t), "How to trace latent generative model generated images without artificial watermark?", *ICML*.

Wang, Z., Wang, Z., Jin, M., Du, M., Zhai, J. and Ma, S. (2024u), "Data-centric NLP backdoor defense from the lens of memorization", arXiv preprint arXiv:2409.14200.

Wang, Z., Zhang, J., Shan, S. and Chen, X. (2024v), "T2IShield: defending against backdoors on text-to-image diffusion models", *ECCV*.

Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P. (2004), "Image quality assessment: from error visibility to structural similarity", *IEEE transactions on Image Processing*, Vol. 13, pp. 600-612.

Wang, Z., Li, H., Zhang, R., Liu, Y., Jiang, W., Fan, W., Zhao, Q. and Xu, G. (2025h), "MPMA: preference manipulation attack against model context protocol", arXiv preprint arXiv:2505.11154.

Wang, Z.J., Montoya, E., Munechika, D., Yang, H., Hoover, B. and Chau, D.H. (2022b), "Diffusiondb: a large-scale prompt gallery dataset for text-to-image generative models", arXiv preprint arXiv:2210.14896.

Wang, Z., Guo, J., Zhu, J., Li, Y., Huang, H., Chen, M. and Tu, Z. (2025i), "Sleepermark: towards robust watermark against fine-tuning text-to-image diffusion models", *CVPR*.

Webster, R. (2023), "A reproducible extraction of training images from diffusion models", arXiv preprint arXiv:2305.08694.

Wei, A., Haghtalab, N. and Steinhardt, J. (2024), "Jailbroken: how does llm safety training fail?", *NeurIPS*.

Wei, X. and Zhao, S. (2023), "Boosting adversarial transferability with learnable patch-wise masks", *IEEE Transactions on Multimedia*, Vol. 26, pp. 3778-3787.

Wei, Z., Chen, J., Goldblum, M., Wu, Z., Goldstein, T. and Jiang, Y.G. (2022), "Towards transferable adversarial attacks on vision trans-formers", *AAAI*.

Wei, Z., Chen, J., Goldblum, M., Wu, Z., Goldstein, T., Jiang, Y.G. and Davis, L.S. (2023), "Towards transferable adversarial attacks on image and video transformers", *IEEE Transactions on Image Processing*, Vol. 32, pp. 6346-6358.

Wen, T. *et al*. (2025a), "Defending against indirect prompt injection by instruction detection", arXiv preprint arXiv:2505.06311.

Wen, Y., Bi, K., Chen, W., Guo, J. and Cheng, X. (2025b), "Evaluating implicit bias in large language models by attacking from a psychometric perspective", *Findings of ACL*.

Wen, Y., Kirchenbauer, J., Geiping, J. and Goldstein, T. (2023), "Tree-ring watermarks: fingerprints for diffusion images that are invisible and robust", *NeurIPS*.

Wen, Y., Liu, Y., Chen, C. and Lyu, L. (2024), "Detecting, explaining, and mitigating memorization in diffusion models", *ICLR*.

Wu, B., Gu, J., Li, Z., Cai, D., He, X. and Liu, W. (2022a), "Towards efficient adversarial training on vision transformers", *ECCV*.

Wu, C.H., Koh, J.Y., Salakhutdinov, R., Fried, D. and Raghunathan, A. (2024a), "Adversarial attacks on multimodal agents", arXiv preprint arXiv:2406.12814.

Wu, F., Wu, S., Cao, Y. and Xiao, C. (2024b), "Wipi: a new web threat for LLM-driven web agents", arXiv preprint arXiv:2402.16965.

Wu, F., Zhang, N., Jha, S., McDaniel, P. and Xiao, C. (2024c), "A new era in LLM security: exploring security concerns in real-world llm-based systems", arXiv preprint arXiv:2402.18649.

Wu, H., Ou, G., Wu, W. and Zheng, Z. (2024d), "Improving transferable targeted adversarial attacks with model self-enhancement", *CVPR*.

Wu, J., Deng, J., Pang, S., Chen, Y., Xu, J., Li, X. and Xu, W. (2024e), "Legilimens: practical and unified content moderation for large language model services", *ACM CCS*.

Wu, L., Yang, X., Dong, Y., Xie, L., Su, H. and Zhu, J. (2024f), "Embodied active defense: leveraging recurrent feedback to counter adversarial patches", *ICLR*.

Wu, X., Pang, Y., Liu, T. and Wu, S. (2025), "Winning the MIDST challenge: new membership inference attacks on diffusion models for tabular data synthesis", arXiv preprint arXiv:2503.12008.

Wu, X., Zhang, J. and Wu, S. (2024g), "Revealing the unseen: guiding personalized diffusion models to expose training data", arXiv preprint arXiv:2410.03039.

Wu, Y., Yu, N., Li, Z., Backes, M. and Zhang, Y. (2022b), "Membership inference attacks against text-to-image generation models", arXiv preprint arXiv: 2210.00968.

Wu, Y., Zhou, S., Yang, M., Wang, L., Zhu, W., Chang, H., Zhou, X. and Yang, X. (2024h), "Unlearning concepts in diffusion model via concept domain correction and concept preserving gradient", arXiv preprint arXiv:2405.15304.

Wu, Y., Li, X., Liu, Y., Zhou, P. and Sun, L. (2023), "Jailbreaking gpt-4v via self-adversarial attacks with system prompts", arXiv preprint arXiv:2311.09127.

Xi, Z., Du, T., Li, C., Pang, R., Ji, S., Chen, J., Ma, F. and Wang, T. (2024), "Defending pre-trained language models as few-shot learners against backdoor attacks", *NeurIPS*.

Xia, S., Yang, W., Yu, Y., Lin, X., Ding, H., Duan, L. and Jiang, X. (2024), "Transferable adversarial attacks on sam and its downstream models", *NeurIPS*.

Xiang, C., Wu, T., Zhong, Z., Wagner, D., Chen, D. and Mittal, P. (2024a), "Certifiably robust rag against retrieval corruption", arXiv preprint arXiv:2405.15556.

Xiang, Z., Jiang, F., Xiong, Z., Ramasubramanian, B., Poovendran, R. and Li, B. (2024b), "BadChain: backdoor chain-of-thought prompting for large language models", *NeurIPS Workshop*.

Xiang, Z., Zheng, L., Li, Y., Hong, J., Li, Q., Xie, H., Zhang, J., Xiong, Z., Xie, C., Yang, C., *et al*. (2024c), "GuardAgent: Safeguard LLM Agents by a Guard Agent via Knowledge-Enabled Reasoning" arXiv preprint arXiv:2406.09187.

Xiao, J., Hays, J., Ehinger, K.A., Oliva, A. and Torralba, A. (2010), "Sun database: large-scale scene recognition from abbey to zoo", *CVPR*.

Xiao, Z., Yang, Y., Chen, G. and Chen, Y. (2024), "Tastle: distract large language models for automatic jailbreak attack", *EMNLP*.

Xie, N., Lai, F., Doran, D. and Kadav, A. (2019), "Visual entailment: a novel task for fine-grained image understanding", arXiv preprint arXiv:1901.06706.

Xie, T., Qi, X., Zeng, Y., Huang, Y., Sehwag, U.M., Huang, K., He, L., Wei, B., Li, D., Sheng, Y., *et al*. (2024), "Sorry-bench: systematically evaluating large language model safety refusal behaviors", arXiv preprint arXiv:2406.14598.

Xing, F. (2025), "Designing heterogeneous llm agents for financial sentiment analysis", *ACM Transactions on Management Information Systems*, Vol. 16 No. 1, pp. 1-24.

Xiong, C., Qi, X., Chen, P.Y. and Ho, T.Y. (2024), "Defensive prompt patch: a robust and interpretable defense of llms against jailbreak attacks", arXiv preprint arXiv:2405.20099.

Xu, C., Zhang, D., Shi, E., Zhang, B. and Wang, D.S. (2024), "AdvAgent: controllable blackbox red-teaming on web agents", arXiv preprint arXiv:2410.17401.

Xu, G., Liu, J., Yan, M., Xu, H., Si, J., Zhou, Z., Yi, P., Gao, X., Sang, J., Zhang, R., *et al*. (2023), "Cvalues: measuring the values of chinese large language models from safety to responsibility", arXiv preprint arXiv:2307.09705.

Xu, H., Zhang, W., Wang, Z., Xiao, F., Zheng, R., Feng, Y., Ba, Z. and Ren, K. (2024a), "Redagent: red teaming large language models with context-aware autonomous language agent", arXiv preprint arXiv:2407.16667.

Xu, J., Niu, Z., Xiang, L., Yang, X. and Li, J. (2024b), "Safeguarding vision-language models against patched visual prompt injectors", arXiv preprint arXiv:2405.10529.

Xu, J., Ma, M.D., Wang, F., Xiao, C. and Chen, M. (2024c), "Instructions as backdoors: backdoor vulnerabilities of instruction tuning for large language models", *NAACL*.

Xu, R., Lin, B., Yang, S., Zhang, T., Shi, W., Zhang, T., Fang, Z., Xu, W. and Qiu, H. (2024d), "The earth is flat because…: investigating LLMs' belief towards misinformation via persuasive conversation", *ACL*.

Xu, R., Zhou, Z., Zhang, T., Qi, Z., Yao, S., Xu, K., Xu, W. and Qiu, H. (2024e), "Walking in others' shoes: how perspective-taking guides large language models in reducing toxicity and bias", *EMNLP*.

Xu, X., Chen, X., Liu, C., Rohrbach, A., Darrell, T. and Song, D. (2018), "Fooling vision and language models despite localization and atten-tion mechanism", *CVPR*.

Xu, Y., Yao, J., Shu, M., Sun, Y., Wu, Z., Yu, N., Goldstein, T. and Huang, F. (2024f), "Shadowcast: stealthy data poisoning attacks against vision-language models", *NeurIPS*.

Xu, Z., Jiang, F., Niu, L., Deng, Y., Poovendran, R., Choi, Y. and Lin, B.Y. (2024g), "Magpie: alignment data synthesis from scratch by prompting aligned LLMs with nothing", arXiv preprint arXiv:2406.08464.

Xue, J., Zheng, M., Hu, Y., Liu, F., Chen, X. and Lou, Q. (2024a), "Badrag: identifying vulnerabilities in retrieval augmented generation of large language models", arXiv preprint arXiv:2406.00083.

Xue, J., Zheng, M., Hua, T., Shen, Y., Liu, Y., Bölöni, L. and Lou, Q. (2024b), "Trojllm: a black-box Trojan prompt attack on large lan-guage models", *NeurIPS*.

Yan, J., Gupta, V. and Ren, X. (2023), "Bite: textual backdoor attacks with iterative trigger injection", *ACL*.

Yan, J., Yadav, V., Li, S., Chen, L., Tang, Z., Wang, H., Srinivasan, V., Ren, X. and Jin, H. (2024a), "Backdooring instruction-tuned large language models with virtual prompt injection", *NAACL*.

Yan, L., Zhang, Z., Tao, G., Zhang, K., Chen, X., Shen, G. and Zhang, X. (2024b), "Parafuzz: an interpretability-driven technique for detect-ing poisoned samples in nlp", *NeurIPS*.

Yan, S., Wang, S., Duan, Y., Hong, H., Lee, K., Kim, D. and Hong, Y. (2024), "An LLM-assisted easy-to-trigger poisoning at-tack on code completion models: injecting disguised vulnerabilities against strong detection", *USENIX Security*.

Yang, D., Bai, Y., Jia, X., Liu, Y., Cao, X. and Yu, W. (2024a), "On the multi-modal vulnerability of diffusion models", *TiFA*.

Yang, H., Yao, F., Chan, C.T., Chen, B. and Wang, C. (2024b), "TrustAgent: towards safe and trustworthy LLM-based agents", arXiv preprint arXiv:2402.01586.

Yang, J., Duan, J., Tran, S., Xu, Y., Chanda, S., Chen, L., Zeng, B., Chilimbi, T. and Huang, J. (2022), "Vision-language pre-training with triple contrastive learning", *CVPR*.

Yang, K., Lin, W.Y., Barman, M., Condessa, F. and Kolter, Z. (2021), "Defending multimodal fusion models against single-source adver-saries", *CVPR*.

Yang, K., Klein, D., Celikyilmaz, A., Peng, N. and Tian, Y. (2024c), "RLCD: reinforcement learning from contrastive distillation for LM alignment", *ICLR*.

Yang, S., Bai, J., Gao, K., Yang, Y., Li, Y. and Xia, S.T. (2024d), "Not all prompts are secure: a switchable backdoor attack against pre-trained vision transfomers", *CVPR*.

Yang, T., Li, Z., Cao, J. and Xu, C. (2024e), "Pruning for robust concept erasing in diffusion models", *NeurIPS Workshop*.

Yang, W., Gao, J. and Mirzasoleiman, B. (2024f), "Better safe than sorry: pre-training CLIP against targeted data poisoning and backdoor attacks", *ICML*.

Yang, W., Gao, J. and Mirzasoleiman, B. (2024g), "Robust contrastive language-image pretraining against data poisoning and backdoor attacks", *NeurIPS*.

Yang, W., Bi, X., Lin, Y., Chen, S., Zhou, J. and Sun, X. (2024h), "Watch out for your agents! investigating backdoor threats to llm-based agents", *NeurIPS*.

Yang, Y., Gao, R., Wang, X., Ho, T.Y., Xu, N. and Xu, Q. (2024i), "Mma-diffusion: multimodal attack on diffusion models", *CVPR*.

Yang, Y., Hui, B., Yuan, H., Gong, N. and Cao, Y. (2024j), "Sneakyprompt: jailbreaking text-to-image generative models", *IEEE S&P*.

Yang, Z., He, X., Li, Z., Backes, M., Humbert, M., Berrang, P. and Zhang, Y. (2023), "Data poisoning attacks against multimodal en-coders", *ICML*.

Yao, D., Zhang, J., Harris, I.G. and Carlsson, M. (2024a), "Fuzzllm: a novel and universal fuzzing framework for proactively discovering jailbreak vulnerabilities in large language models", *ICASSP*.

Yao, H., Lou, J. and Qin, Z. (2024b), "Poisonprompt: backdoor attack on prompt-based large language models", *ICASSP*.

Yao, Y., Li, L., Song, J., Chen, C., He, Z., Wang, Y., Wang, X., Gu, T., Li, J., Teng, Y., *et al*. (2025), "Argus inspection: do multimodal large language models possess the eye of panoptes?", arXiv preprint arXiv:2506.14805.

Ye, J., Li, S., Li, G., Huang, C., Gao, S., Wu, Y., Zhang, Q., Gui, T. and Huang, X. (2024a), "Toolsword: unveiling safety issues of large language models in tool learning across three stages", arXiv preprint arXiv:2402.10753.

Ye, M., Rong, X., Huang, W., Du, B., Yu, N. and Tao, D. (2025), "A survey of safety on large vision-language models: attacks, defenses and evaluations", arXiv preprint arXiv:2502.14881.

Ye, R., Pang, X., Chai, J., Chen, J., Yin, Z., Xiang, Z., Dong, X., Shao, J. and Chen, S. (2024b), "Are we there yet? Revealing the risks of utilizing large language models in scholarly peer review", arXiv preprint arXiv:2412.01708.

Ye, X., Huang, H., An, J. and Wang, Y. (2024c), "DUAW: data-free universal adversarial watermark against stable diffusion customization", *ICLR Workshop*.

Yi, B., Huang, T., Chen, S., Li, T., Liu, Z., Chu, Z. and Li, Y. (2025), "Probe before you talk: towards black-box defense against back-door unalignment for large language models", *ICLR*.

Yi, J., Xie, Y., Zhu, B., Kiciman, E., Sun, G., Xie, X. and Wu, F. (2023), "Benchmarking and defending against indirect prompt injection attacks on large language models", arXiv preprint arXiv:2312.14197.

Yi, S., Liu, Y., Sun, Z., Cong, T., He, X., Song, J., Xu, K. and Li, Q. (2024), "Jailbreak attacks and defenses against large language models: a survey", arXiv preprint arXiv:2407.04295.

Yin, S., Pang, X., Ding, Y., Chen, M., Bi, Y., Xiong, Y., Huang, W., Xiang, Z., Shao, J. and Chen, S. (2024), "SafeAgentBench: a benchmark for safe task planning of embodied LLM agents", arXiv preprint arXiv:2412.13178.

Yin, Z., Ye, M., Zhang, T., Du, T., Zhu, J., Liu, H., Chen, J., Wang, T. and Ma, F. (2023), "VLATTACK: multimodal adversarial attacks on vision-language tasks via pre-trained models", *NeurIPS*.

Ying, Z., Liu, A., Liu, X. and Tao, D. (2024), "Unveiling the safety of GPT-4o: an empirical study using jailbreak attacks", arXiv preprint arXiv:2406.06302.

Yong, Z.X., Menghini, C. and Bach, S.H. (2023), "Low-resource lan-guages jailbreak gpt-4", *NeurIPS Workshop*.

Yu, J., Lin, X., Yu, Z. and Xing, X. (2023a), "Gptfuzzer: red teaming large language models with auto-generated jailbreak prompts", arXiv preprint arXiv:2309.10253.

Yu, J., Luo, H., Yao-Chieh, J., Guo, W., Liu, H. and Xing, X. (2024a), "Enhancing jailbreak attack against large language models through silent tokens", arXiv preprint arXiv:2405.20653.

Yu, J., Shao, Y., Miao, H. and Shi, J. (2024b), "Promptfuzz: harnessing fuzzing techniques for robust testing of prompt injection in llms", arXiv preprint arXiv:2409.14729.

Yu, L., Poirson, P., Yang, S., Berg, A.C. and Berg, T.L. (2016), "Modeling context in referring expressions", *ECCV*.

Yu, W., Pang, T., Liu, Q., Du, C., Kang, B., Huang, Y., Lin, M. and Yan, S. (2023b), "Bag of tricks for training data extraction from language models", *ICML*.

Yuan, L., Jia, X., Huang, Y., Dong, W. and Liu, Y. (2025), "Promptguard: soft prompt-guided unsafe content moderation for text-to-image models", arXiv preprint arXiv:2501.03544.

Yuan, T., He, Z., Dong, L., Wang, Y., Zhao, R., Xia, T., Xu, L., Zhou, B., Li, F., Zhang, Z., *et al*. (2024), "R-judge: benchmarking safety risk awareness for llm agents", *EMNLP*.

Yuan, Y., Jiao, W., Wang, W., Huang, J.T., He, P., Shi, S. and Tu, Z. (2023a), "Gpt-4 is too smart to be safe: stealthy chat with llms via cipher", arXiv preprint arXiv:2308.06463.

Yuan, Z., Zhou, P., Zou, K. and Cheng, Y. (2023b), "You are catching my attention: are vision transformers bad learners under backdoor attacks?", *CVPR*.

Yung, C., Dolatabadi, H.M., Erfani, S. and Leckie, C. (2025a), "Round trip translation defence against large language model jailbreaking attacks", *PAKDD*.

Yung, C., Huang, H., Erfani, S.M. and Leckie, C. (2025b), "Curvalid: geometrically-guided adversarial prompt detection", arXiv preprint arXiv:2503.03502.

Zeng, Q., Jin, M., Yu, Q., Wang, Z., Hua, W., Zhou, Z., Sun, G., Meng, Y., Ma, S., Wang, Q., *et al*. (2024a), "Uncertainty is fragile: manipulating uncertainty in large language models", arXiv preprint arXiv:2407.11282.

Zeng, Y., Sun, W., Huynh, T.N., Song, D., Li, B. and Jia, R. (2024b), "Beear: embedding-based adversarial removal of safety backdoors in instruction-tuned language models", *EMNLP*.

Zeng, Y., Wu, Y., Zhang, X., Wang, H. and Wu, Q. (2024c), "AutoDefense: multi-agent LLM defense against jailbreak attacks", *NeurIPS Workshop*.

Zhai, S., Chen, H., Dong, Y., Li, J., Shen, Q., Gao, Y., Su, H. and Liu, Y. (2024), "Membership inference on text-to-image diffusion models via conditional likelihood discrepancy", *NeurIPS*.

Zhai, S., Dong, Y., Shen, Q., Pu, S., Fang, Y. and Su, H. (2023), "Text-to-image diffusion models can be easily backdoored through multimodal data poisoning", *ACM MM*.

Zhai, S., Li, J., Liu, Y., Chen, H., Tian, Z., Qu, W., Shen, Q., Jia, R., Dong, Y. and Zhang, J. (2025), "NaviDet: efficient input-level back-door detection on text-to-image synthesis via neuron activation variation", arXiv preprint arXiv:2503.06453.

Zhan, Q., Fang, R., Panchal, H.S. and Kang, D. (2025), "Adaptive attacks break defenses against indirect prompt injection attacks on LLM agents", *NAACL*.

Zhan, Q., Liang, Z., Ying, Z. and Kang, D. (2024), "Injecagent: bench-marking indirect prompt injections in tool-integrated large language model agents", *ACL*.

Zhang, B., Zhang, Y., Ji, J., Lei, Y., Dai, J., Chen, Y. and Yang, Y. (2025a), "SafeVLA: towards safety alignment of vision-language-action model via constrained learning", arXiv preprint arXiv:2503.03480.

Zhang, B., Luo, W. and Zhang, Z. (2023a), "Purify++: improving diffusion-purification with advanced diffusion models and control of randomness", arXiv preprint arXiv:2310.18762.

Zhang, B., Tan, Y., Shen, Y., Salem, A., Backes, M., Zannettou, S. and Zhang, Y. (2024a), "Breaking agents: compromising autonomous LLM agents through malfunction amplification", arXiv preprint arXiv:2407.20859.

Zhang, C., Zhang, C., Kang, T., Kim, D., Bae, S.H. and Kweon, I.S. (2023b), "Attack-sam: towards evaluating adversarial robustness of segment anything model", arXiv preprint arXiv:2305.00866.

Zhang, C., Wang, L. and Liu, A. (2024b), "Revealing vulnerabilities in stable diffusion via targeted attacks", arXiv preprint arXiv:2401.08725.

Zhang, C., Wang, L., Ma, Y., Li, W. and Liu, A.A. (2025b), "Rea-son2attack: jailbreaking text-to-image models via llm reasoning", arXiv preprint arXiv:2503.17987.

Zhang, C., Xu, X., Wu, J., Liu, Z. and Zhou, L. (2024c), "Adversarial attacks of vision tasks in the past 10 years: a survey", arXiv preprint arXiv:2410.23687.

Zhang, C., Jin, M., Yu, Q., Liu, C., Xue, H. and Jin, X. (2024d), "Goal-guided generative prompt injection attack on large language models", arXiv preprint arXiv:2404.07234.

Zhang, G., Wang, K., Xu, X., Wang, Z. and Shi, H. (2024e), "Forget-me-not: learning to forget in text-to-image diffusion models", *CVPR*.

Zhang, H., Zhu, C., Wang, X., Zhou, Z., Yin, C., Li, M., Xue, L., Wang, Y., Hu, S., Liu, A., Guo, P. and Zhang, L.Y. (2025c), "BadRobot: jailbreaking embodied LLMs in the physical world", *ICLR*, doi: 10.48550/arXiv.2407.20242.

Zhang, H., Huang, J., Mei, K., Yao, Y., Wang, Z., Zhan, C., Wang, H. and Zhang, Y. (2024f), "Agent Security Bench (ASB): formalizing and benchmarking attacks and defenses in LLM-based agents", arXiv preprint arXiv:2410.02644.

Zhang, H., Shao, W., Liu, H., Ma, Y., Luo, P., Qiao, Y. and Zhang, K. (2024g), "Avibench: towards evaluating the robustness of large vision-language model on adversarial visual-instructions", arXiv preprint arXiv:2403.09346.

Zhang, J., Wang, Z., Wang, R., Ma, X. and Jiang, Y.G. (2024h), "EnJa: ensemble jailbreak on large language models", arXiv preprint arXiv:2408.03603.

Zhang, J., Hu, R., Guo, Q. and Lim, W.Y.B. (2025d), "CAVALRY-V: a large-scale generator framework for adversarial attacks on video MLLMs", arXiv preprint arXiv:2507.00817.

Zhang, J., Ma, X., Wang, X., Qiu, L., Wang, J., Jiang, Y.G. and Sang, J. (2024i), "Adversarial prompt tuning for vision-language models", *ECCV*.

Zhang, J., Ye, J., Ma, X., Li, Y., Yang, Y., Sang, J. and Yeung, D.Y. (2024j), "AnyAttack: towards large-scale self-supervised generation of targeted adversarial examples for vision-language models", arXiv preprint arXiv:2410.05346.

Zhang, J., Yi, Q. and Sang, J. (2022), "Towards adversarial attack on vision-language pre-training models", *ACM MM*.

Zhang, J., Huang, Y., Wu, W. and Lyu, M.R. (2023c), "Transferable adversarial attacks on vision transformers with token gradient regularization", *CVPR*.

Zhang, J., Huang, Y., Xu, Z., Wu, W. and Lyu, M.R. (2024k), "Improving the adversarial transferability of vision transformers with virtual dense connection", *AAAI*.

Zhang, J., Yang, S. and Li, B. (2025e), "UDora: a unified red teaming framework against llm agents by dynamically hijacking their own reasoning", arXiv preprint arXiv:2503.01908.

Zhang, J., Liu, H., Jia, J. and Gong, N.Z. (2024l), "Data poisoning based backdoor attacks to contrastive learning", *CVPR*.

Zhang, L., Rao, A. and Agrawala, M. (2023d), "Adding conditional control to text-to-image diffusion models", *ICCV*.

Zhang, M., Yu, N., Wen, R., Backes, M. and Zhang, Y. (2024m), "Generated distributions are all you need for membership inference attacks against generative models", *CVPR*.

Zhang, P.F., Huang, Z. and Bai, G. (2024n), "Universal adversarial perturbations for vision-language pre-trained models", *SIGIR*.

Zhang, Q., Qiu, H., Wang, D., Li, Y., Zhang, T., Zhu, W., Weng, H., Yan, L. and Zhang, C. (2025f), "A benchmark for semantic sensitive information in llms outputs", *ICLR*.

Zhang, Q., Qiu, H., Wang, D., Qian, H., Li, Y., Zhang, T. and Huang, M. (2024o), "Understanding the dark side of LLMs' intrinsic self-correction", arXiv preprint arXiv:2412.14959.

Zhang, Q., Zeng, B., Zhou, C., Go, G., Shi, H. and Jiang, Y. (2024p), "Human-imperceptible retrieval poisoning attacks in LLM-powered applications", arXiv preprint arXiv:2404.17196.

Zhang, R., Li, H., Wen, R., Jiang, W., Zhang, Y., Backes, M., Shen, Y. and Zhang, Y. (2024q), "Instruction backdoor attacks against customized LLMs", *USENIX Security*.

Zhang, S., Yin, M., Zhang, J., Liu, J., Han, Z., Zhang, J., Li, B., Wang, C., Wang, H., Chen, Y., *et al*. (2025g), "Which agent causes task failures and when? On automated failure attribution of llm multi-agent systems", arXiv preprint arXiv:2505.00212.

Zhang, S., Zhang, M., Pan, X. and Yang, M. (2023e), "No-Skim: towards efficiency robustness evaluation on skimming-based language models", arXiv preprint arXiv:2312.09494.

Zhang, S. *et al*. (2024r), "SG-Bench: evaluating LLM safety generalization across diverse tasks and prompt types", arXiv preprint arXiv:2410.21965.

Zhang, T. *et al*. (2024s), "Agent-SafetyBench: evaluating the safety of LLM agents", arXiv preprint arXiv:2412.14470.

Zhang, X., Zhang, C., Li, T., Huang, Y., Jia, X., Xie, X., Liu, Y. and Shen, C. (2023f), "A mutation-based method for multi-modal jailbreaking attack detection", arXiv preprint arXiv:2312.10766.

Zhang, X., Xu, H., Ba, Z., Wang, Z., Hong, Y., Liu, J., Qin, Z. and Ren, K. (2024t), "Privacyasst: safeguarding user privacy in tool-using large language model agents", *IEEE Transactions on Dependable and Secure Computing*, Vol. 21 No. 6, pp. 5242-5258.

Zhang, X., Li, R., Yu, J., Xu, Y., Li, W. and Zhang, J. (2024u), "Editguard: versatile image watermarking for tamper localization and copyright protection", *CVPR*.

Zhang, Y. (2024), "Attacking vision-language computer agents via pop-ups", arXiv preprint arXiv:2411.02391.

Zhang, Y., Chen, X., Jia, J., Zhang, Y., Fan, C., Liu, J., Hong, M., Ding, K. and Liu, S. (2024v), "Defensive unlearning with adversarial training for robust concept erasure in diffusion models", *NeurIPS*.

Zhang, Y., Jia, J., Chen, X., Chen, A., Zhang, Y., Liu, J., Ding, K. and Liu, S. (2023g), "To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images… for now", arXiv preprint arXiv:2310.11868.

Zhang, Z., Zhang, Y., Li, L., Gao, H., Wang, L., Lu, H., Zhao, F., Qiao, Y. and Shao, J. (2024w), "Psysafe: a comprehensive framework for psychological-based attack, defense, and evaluation of multi-agent system safety", arXiv preprint arXiv:2401.11880.

Zhang, Z., Lei, L., Wu, L., Sun, R., Huang, Y., Long, C., Liu, X., Lei, X., Tang, J. and Huang, M. (2024x), "SafetyBench: evaluating the safety of large language models", *ACL*.

Zhang, Z., Zhang, Q. and Foerster, J. (2024y), "PARDEN, can you repeat that? Defending against jailbreaks via repetition", arXiv preprint arXiv:2405.07932.

Zhao, M., Zhang, L., Zheng, T., Kong, Y. and Yin, B. (2024a), "Separable multi-concept erasure from diffusion models", arXiv preprint arXiv:2402.05947.

Zhao, S., Jia, M., Guo, Z., Gan, L., Xu, X., Wu, X., Fu, J., Feng, Y., Pan, F. and Tuan, L.A. (2024b), "A survey of backdoor attacks and defenses on large

language models: implications for security measures", arXiv preprint arXiv:2406.06852.

Zhao, S., Jia, M., Tuan, L.A., Pan, F. and Wen, J. (2024c), "Universal vulnerabilities in large language models: backdoor attacks for in-context learning", *EMNLP*.

Zhao, S., Wen, J., Tuan, L.A., Zhao, J. and Fu, J. (2023a), "Prompt as triggers for backdoor attack: examining the vulnerability in language models", *EMNLP*.

Zhao, W., Li, Z., Li, Y., Zhang, Y. and Sun, J. (2024d), "Defending large language models against jailbreak attacks via layer-specific editing", *EMNLP*.

Zhao, X., Yang, X., Pang, T., Du, C., Li, L., Wang, Y.X. and Wang, W.Y. (2024e), "Weak-to-strong jailbreaking on large language models", arXiv preprint arXiv:2401.17256.

Zhao, Y., Zheng, X., Luo, L., Li, Y., Ma, X. and Jiang, Y.G. (2025), "BlueSuffix: reinforced blue teaming for vision-language models against jailbreak attacks", *ICLR*.

Zhao, Y., Pang, T., Du, C., Yang, X., Cheung, N.M. and Lin, M. (2023b), "A recipe for watermarking diffusion models", arXiv preprint arXiv:2303.10137.

Zhao, Y., Pang, T., Du, C., Yang, X., Li, C., Cheung, N.M.M. and Lin, M. (2024f), "On evaluating adversarial robustness of large vision-language models", *NeurIPS*.

Zheng, M., Lou, Q. and Jiang, L. (2023), "Trojvit: Trojan insertion in vision transformers", *CVPR*.

Zheng, S. and Zhang, C. (2023), "Black-box targeted adversarial attack on segment anything (sam)", arXiv preprint arXiv:2310.10010.

Zheng, X., Pang, T., Du, C., Liu, Q., Jiang, J. and Lin, M. (2024), "Improved few-shot jailbreaking can circumvent aligned language models and their defenses", arXiv preprint arXiv:2406.01288.

Zhong, Z., Huang, Z., Wettig, A. and Chen, D. (2023), "Poisoning retrieval corpora by injecting adversarial passages", arXiv preprint arXiv:2310.19156.

Zhou, A., Wu, K., Pinto, F., Chen, Z., Zeng, Y., Yang, Y., Yang, S., Koyejo, S., Zou, J. and Li, B. (2025a), "Autoredteamer: autonomous red teaming with lifelong attack integration", arXiv preprint arXiv:2503.15754.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A. and Torralba, A. (2017), "Scene parsing through ade20k dataset", *CVPR*.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., *et al*. (2024a), "Lima: less is more for alignment", *NeurIPS*.

Zhou, H., Lee, K.H., Zhan, Z., Chen, Y. and Li, Z. (2025b), "TrustRAG: enhancing robustness and trustworthiness in RAG", arXiv preprint arXiv:2501.00879.

Zhou, J., Wang, M., Li, T., Meng, G. and Chen, K. (2024b), "Dormant: defending against pose-driven human image animation", arXiv preprint arXiv:2409.14424.*8*.

Zhou, J., Ye, K., Liu, J., Ma, T., Wang, Z., Qiu, R., Lin, K.Y., Zhao, Z. and Liang, J. (2025c), "Exploring the limits of vision-language-action manipulations in cross-task generalization", arXiv preprint arXiv:2505.15660.

Zhou, K., Yang, J., Loy, C.C. and Liu, Z. (2022a), "Conditional prompt learning for vision-language models", *CVPR*.

Zhou, K., Yang, J., Loy, C.C. and Liu, Z. (2022b), "Learning to prompt for vision-language models", *International Journal of Computer Vision*, Vol. 130 No. 9, pp. 2337-2348.

Zhou, Q., Wang, D., Li, T., Lin, Y., Liu, Y., Dong, J.S. and Guo, Q. (2025d), "Defending LVLMs against vision attacks through partial-perception supervision", *ICML*.

Zhou, Q., Wang, D., Li, T., Xu, Z., Liu, Y., Ren, K., Wang, W. and Guo, Q. (2024c), "Foolsdedit: deceptively steering your edits towards targeted attribute-aware distribution", arXiv preprint arXiv:2402.03705.

Zhou, W., Bai, S., Zhao, Q. and Chen, B. (2024d), "Revisiting the adversarial robustness of vision language models: a multimodal perspective", arXiv preprint arXiv:2404.19287.

Zhou, W., Wang, X., Xiong, L., Xia, H., Gu, Y., Chai, M., Zhu, F., Huang, C., Dou, S., Xi, Z., *et al*. (2024e), "EasyJailbreak: a unified framework for jailbreaking large language models", arXiv preprint arXiv:2403.12171.

Zhou, X. *et al*. (2024f), "HAICOSYSTEM: an ecosystem for sandboxing safety risks in human-AI interactions", *NeurIPS*.

Zhou, Y., Xia, X., Lin, Z., Han, B. and Liu, T. (2024g), "Few-shot adver-sarial prompt learning on vision-language models", *NeurIPS*.

Zhou, Z., Liu, J., Yang, C., Shao, J., Liu, Y., Yue, X., Ouyang, W. and Qiao, Y. (2024h), "Beyond one-preference-for-all: multi-objective direct preference optimization", *ACL*.

Zhou, Z., Li, Z., Zhang, J., Zhang, Y., Wang, K., Liu, Y. and Guo, Q. (2025e), "Corba: contagious recursive blocking attacks on multi-agent systems based on large language models", arXiv preprint arXiv:2502.14529.

Zhou, Z., Hu, S., Li, M., Zhang, H., Zhang, Y. and Jin, H. (2023), "Ad-vclip: downstream-agnostic adversarial examples in multimodal contrastive learning", *ACM MM*.

Zhou, Z., Song, Y., Li, M., Hu, S., Wang, X., Zhang, L.Y., Yao, D. and Jin, H. (2024i), "DarkSAM: fooling segment anything model to segment nothing", *NeurIPS*.

Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T. and Liu, J. (2020), "FreeLB: enhanced adversarial training for natural language understanding", *ICLR*.

Zhu, J., Kaplan, R., Johnson, J. and Fei-Fei, L. (2018), "HiDDeN: hiding data with deep networks", *ECCV*.

Zhu, L., Ning, R., Li, J., Xin, C. and Wu, H. (2024a), "SEER: backdoor detection for vision-language models through searching target text and image trigger jointly", *AAAI*.

Zhu, P., Takahashi, T. and Kataoka, H. (2024b), "Watermark-embedded adversarial examples for copyright protection against diffusion models", *CVPR*.

Zhu, P., Zhou, Z., Zhang, Y., Yan, S., Wang, K. and Su, S. (2025), "Demonagent: dynamically encrypted multi-backdoor implantation attack on llm-based agent", arXiv preprint arXiv:2502.12575.

Zhu, Y., Kellermann, A., Gupta, A., Li, P., Fang, R., Bindu, R. and Kang, D. (2024c), "Teams of llm agents can exploit zero-day vulnerabilities", arXiv preprint arXiv:2406.01637.

Zhuang, H., Zhang, Y. and Liu, S. (2023), "A pilot study of query-free adversarial attack against stable diffusion", *CVPR*.

Ziegler, D.M., Stiennon, N., Wu, J., Brown, T.B., Radford, A., Amodei, D., Christiano, P. and Irving, G. (2019), "Fine-tuning language models from human preferences", arXiv preprint arXiv:1909.08593.

Zollicoffer, G., Vu, M., Nebgen, B., Castorena, J., Alexandrov, B. and Bhattarai, M. (2024), "LoRID: low-rank iterative diffusion for adversarial purification", arXiv preprint arXiv:2409.08255.

Zou, A., Phan, L., Wang, J., Duenas, D., Lin, M., Andriushchenko, M., Kolter, J.Z., Fredrikson, M. and Hendrycks, D. (2024a), "Improving alignment and robustness with circuit breakers", *NeurIPS*.

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J.Z. and Fredrikson, M. (2023), "Universal and transferable adversarial attacks on aligned language models", arXiv preprint arXiv:2307.15043.

Zou, W., Geng, R., Wang, B. and Jia, J. (2024b), "Poisonedrag: knowledge corruption attacks to retrieval-augmented generation of large language models", arXiv preprint arXiv:2402.07867.

Zou, X., Chen, Y. and Li, K. (2024c), "Is the system message really important to jailbreaks in large language models?", arXiv preprint arXiv:2402.14857.

Author affiliations

Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, Hengyuan Xu, Yunhao Chen and Yunhan Zhao, Fudan University, China

Hanxun Huang, The University of Melbourne, Australia

Yige Li, Singapore Management University, Singapore

Yutao Wu, Deakin University, Australia

Jiaming Zhang, Hong Kong University of Science and Technology, Hong Kong

Xiang Zheng, City University of Hong Kong, Hong Kong

Yang Bai, ByteDance, China

Yiming Li, Nanyang Technological University, Singapore

Zuxuan Wu and Xipeng Qiu, Fudan University, China

Jingfeng Zhang, University of Auckland, New Zealand, and RIKEN, Japan

Xudong Han and Haonan Li, MBZUAI, UAE

Jun Sun, Singapore Management University, Singapore

Cong Wang, City University of Hong Kong, Hong Kong

Jindong Gu, University of Oxford, UK

Baoyuan Wu, Chinese University of Hong Kong, Shenzhen, China

Siheng Chen, Shanghai Jiao Tong University, China

Tianwei Zhang and Yang Liu, Nanyang Technological University, Singapore

Mingming Gong, The University of Melbourne, Australia

Tongliang Liu, The University of Sydney, Australia

Shirui Pan, Griffith University, Australia

Cihang Xie, University of California, Santa Cruz, USA

Tianyu Pang, Sea AI Lab, Singapore

Yinpeng Dong, Tsinghua University, China

Ruoxi Jia, Virginia Tech, USA

Yang Zhang, CISPA Helmholtz Center for Information Security, Germany

Shiqing Ma, University of Massachusetts Amherst, USA

Xiangyu Zhang, Purdue University, USA

Neil Gong, Duke University, USA

Chaowei Xiao, University of Wisconsin - Madison, USA

Sarah Erfani, The University of Melbourne, Australia

Tim Baldwin, The University of Melbourne, Australia, and MBZUAI, UAE

Bo Li, University of Illinois Urbana-Champaign, USA

Masashi Sugiyama, RIKEN, Japan, and The University of Tokyo, Japan

Dacheng Tao, Nanyang Technological University, Singapore

James Bailey, The University of Melbourne, Australia, and

Yu-Gang Jiang, Fudan University, China

Corresponding author
Yu-Gang Jiang can be contacted at: ygj@fudan.edu.cn