Computational Safety for Generative AI: A Hypothesis Testing Perspective

Pin-Yu Chen [1]

Abstract

AI safety is a rapidly growing area of research that seeks to prevent the harm and misuse of frontier AI technology, particularly with respect to generative AI (GenAI) tools that are capable of creating realistic and high-quality content through text prompts. Examples of such tools include large language models (LLMs) and text-to-image (T2I) diffusion models. As the performance of various leading GenAI models approaches saturation due to similar training data sources and neural network architecture designs, the development of reliable safety guardrails has become a key differentiator for responsibility and sustainability. This paper presents a formalization of the concept of *computational safety*, which is a mathematical framework that enables the quantitative assessment, formulation, and study of safety challenges in GenAI through the lens of signal processing theory and methods. In particular, we explore two exemplary categories of computational safety challenges in GenAI that can be formulated as hypothesis testing problems. For the *safety of model input*, we show how sensitivity analysis and loss landscape analysis can be used to detect malicious prompts with jailbreak attempts. For the *safety of model output*, we elucidate how statistical signal processing can be used to detect AI-generated content. Finally, we discuss opportunities and open challenges in computational safety for GenAI.

1. Introduction

Signal processing for pattern recognition and machine intelligence has played a pivotal role in ensuring the stability, security, and efficiency of numerous engineering systems and information technologies, including, but not limited to, telecommunications, information forensics and security, machine learning, data science, and control systems. With the recent advances, wide accessibility, and deep integration of generative AI (GenAI) tools into our society and technology, such as ChatGPT and the emerging agentic AI applications, understanding and mitigating the associated risks of the so-called "frontier AI technology" is essential to ensure a responsible and sustainable use of GenAI. In addition, as the performance of state-of-the-art GenAI models surpasses that of an average human in certain tasks, but reaches a plateau in standardized capability evaluation benchmarks due to similar training data sources and neural network architecture designs (e.g., the use of decoder-only transformers), improving and ensuring safety is becoming the new arms race among GenAI stakeholders. In addition to ongoing activities in risk governance, policy-making, and legal regulation (e.g., EU AI Act, AI safety/security institutes, etc.), there are growing concerns about the broader socio-technical impacts (Qi et al., 2024). Notable examples include breaking the embedded safety alignment of GenAI models to induce misbehavior (e.g., asking large language models how to perform illegal or dangerous activities) (Zou et al., 2023) or generating inappropriate content, and misusing GenAI tools to create and spread misinformation (e.g., using text prompts to create deepfakes and influence public opinion) (Verdoliva, 2020).

The main goal of this paper is to present the central message in a technically sound and rigorous manner as follows: *While AI safety can be seen as a significant challenge in the context of modern GenAI technology, it is possible to reformulate many of the associated problems as a standard hypothesis testing task*. Furthermore, insights and methods developed in signal processing can be applied to improve AI safety. Specifically, we define *computational AI safety* as the set of safety problems that can be formulated as hypothesis-testing tasks in signal processing. We also provide examples of how signal processing techniques such as sensitivity analysis, subspace projection (Vaswani et al., 2018), loss landscape (Li et al., 2018), and adversarial learning can be used to study and improve AI safety.

Figure 1 provides an overview of our proposed signal processing framework for computational AI safety. Without loss of generality, we divide a GenAI system into three parts: the *model input* that processes a user's query, the *GenAI model* that performs inference and action (for agentic workflows), and the *model output* produced by the model

[1]IBM Research. Correspondence to: Pin-Yu Chen <pin-yu.chen@ibm.com>.

ICML 2026 Workshop on Hypothesis Testing, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

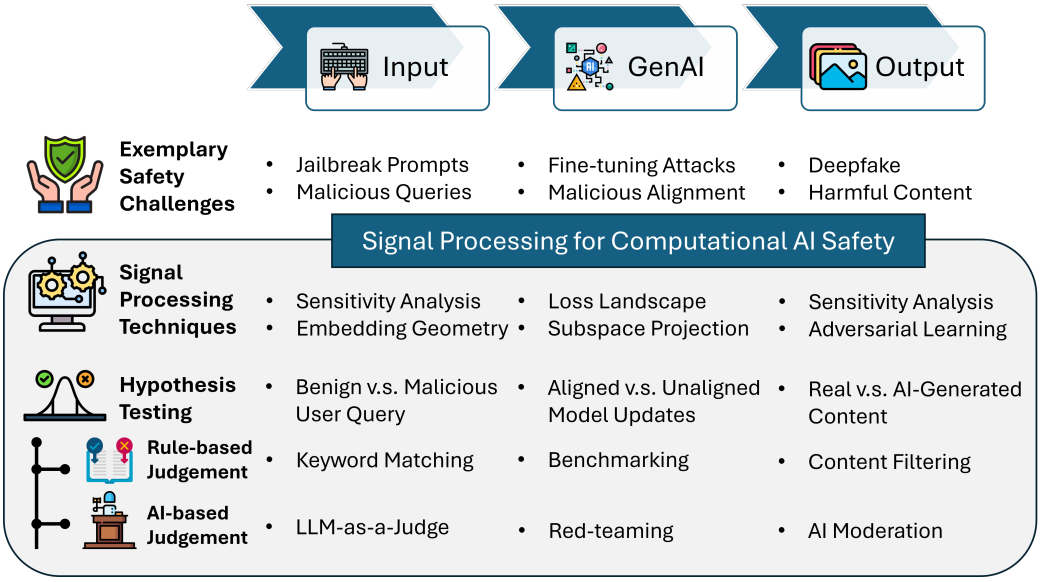


Figure 1. Overview of the framework on signal processing for computational AI safety. The two safety challenges highlighted for model input and output safety in GenAI are the detection of unsafe queries and AI-generated content. We define computational AI safety as the set of safety problems that can be formulated as hypothesis testing tasks in signal processing. We also provide examples of how signal processing techniques such as sensitivity analysis, and subspace projection can be used to improve AI safety. What is unique about GenAI is that the validity of the safety hypothesis (e.g., whether the input or output is safe) requires an additional judge function for certification, which can be either a rule-based approach (e.g., keyword matching) or an AI-based evaluation (e.g., LLM-as-a-judge or external contextual classifiers). The framework can be expanded with additional signal processing methods and system-level designs.

given the input. The two representative safety challenges highlighted for model input and output safety in GenAI are the detection of unsafe queries and AI-generated content. It is worth noting that what is unique about GenAI is that the validity of the safety hypothesis (e.g., whether the input or output is safe) requires an additional judge function for certification, which can be either a rule-based approach (e.g., keyword matching) or an AI-based evaluation (e.g., LLM-as-a-judge or external contextual classifiers). For example, when assessing the risk of jailbreak attacks, one can use refusal-related keywords to quantify the rejection rate in GenAI, or use another LLM as a judge to provide a numerical rating of the safety level of the response.

To bridge AI safety and signal processing, this paper presents an overview of computational safety for GenAI, consolidates numerous computational AI safety problems through the lens of a unified hypothesis testing framework in signal processing, and provides comprehensive technical information to elucidate how signal processing methodologies can facilitate and enhance the development of AI safety tools. This paper's signal-processing-centric perspective differs from the broad range of recent progress in attacks and defenses for GenAI in existing AI safety literature, such as (Chua et al., 2024; Ma et al., 2025; Bereska & Gavves, 2024), or in-depth discussions of AI safety in particular domains (Sangwan et al., 2023; Ullrich et al., 2024). The rest of the paper is organized as follows. First, we provide background on mainstream GenAI models and preliminaries on hypothesis testing and common evaluation metrics. Next, we formally introduce the signal processing framework for computational AI safety. We then examine two representative use cases in AI safety – jailbreak detection and mitigation, and AI-generated content. Finally, we provide our perspective on the position of AI safety and discuss several ongoing research challenges and opportunities at the intersection of signal processing and AI safety.

2. Background and Preliminaries

2.1. Mainstream GenAI Models

This paper focuses on the safety challenges of two mainstream GenAI models: Large Language Models (LLMs) and Diffusion Models (DMs). LLMs are autoregressive neural networks built on self-attention-based transformers with billions of trainable parameters, which dominate text-to-text generation applications (e.g., ChatGPT) and underpin many multi-modal GenAI tools, such as vision-language models, audio-language models, and code-language models. DMs are state-of-the-art neural networks capable of high-quality text-to-image generation through a mathematical diffusion process, with extensions to other modalities such as video, audio, and molecular dynamics. The deployment of LLMs

and DMs has two modes: direct use of the off-the-shelf GenAI model, or customized fine-tuning with some task-specific downstream datasets. Throughout this paper, we will use θ to denote the set of trainable parameters (model weights) for GenAI models.

Large language models (LLMs) The development of LLMs has two primary phases: pre-training and alignment. LLM pre-training refers to unsupervised machine learning on a large dataset (e.g., text extracted from the Internet) denoted by $\mathcal{D}_p$. Let $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ denote a sequence of tokens (i.e., "words" defined in the LLM vocabulary) randomly drawn from $\mathcal{D}_p$. For pre-training, the Next-token prediction (NTP) loss is commonly used, which is defined as the expected loss of the negative log-likelihood

$$\text{Loss}_{\text{NTP}} = -\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_p} \frac{1}{|\mathbf{x}|} \sum_{t=1}^{|\mathbf{x}|} \log p_\theta(x_t|\mathbf{x}_{<t}), \quad (1)$$

where $|\mathbf{x}|$ denotes the token length of $\mathbf{x}$, $\mathbf{x}_{<t} := \{x_1, x_2, \ldots, x_{t-1}\}$ denotes the tokens preceding the t-th token and $\mathbf{x}_{<1} := \varnothing$ (an empty set), $p_\theta(x_t|\mathbf{x}_{<t})$ is parameterized by a softmax function over all tokens. This loss is also known as *causal language modeling*, by using previous tokens to predict the next token. The family of LLMs trained with NTP loss is also called autoregressive LLMs. After pre-training on θ, an LLM undergoes the alignment process to teach the model to follow instructions and value alignment. Safety alignment is also implemented in this stage to guide the model in refusing to respond to unsafe queries. Specifically, for supervised fine-tuning (SFT) (or instruction-tuning), let $(\mathbf{x}, \mathbf{y})$ denote a pair of model input and the desired model output randomly drawn from a dataset $\mathcal{D}_a$, the SFT loss is used to update the pre-trained model weights by generalizing the NTP loss as

$$\text{Loss}_{\text{SFT}} = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_a} \frac{1}{|\mathbf{y}|} \sum_{t=1}^{|\mathbf{y}|} \log p_\theta(y_t|\mathbf{x}, \mathbf{y}_{<t}). \quad (2)$$

Here, $\mathbf{x}$ can also be conceived as the *context*, which is used to model the conditional probability of generating $\mathbf{y}$ given $\mathbf{x}$ as $p_\theta(\mathbf{y}|\mathbf{x}) = \Pi_{t=1}^{|\mathbf{y}|} p_\theta(y_t|\mathbf{x}, \mathbf{y}_{<t})$. In addition to SFT, other forms of alignment include strategies such as reinforcement learning with human feedback (RLHF) (Ouyang et al., 2022) and direct preference optimization (DPO) (Rafailov et al., 2024).

Diffusion Models (DMs) During training, DMs follow a mathematical diffusion process to gradually encode data samples into random Gaussian noises, and train a neural network to learn to decode (reconstruct) the data samples based on multiple denoising steps. Furthermore, DMs can be integrated with text-based prompts to guide the generation process. We briefly explain the basics of DM as follows. Let t denote the diffusion step and let $\mathbf{x}_t$ be the data sample (or its latent representation) obtained via a forward diffusion process subject to noise injection. Given a text prompt (context) c, we denote $\epsilon_\theta(\mathbf{x}_t|c)$ as the noise generator parameterized by a neural network θ. A reverse diffusion process is also used to estimate the underlying noise. In its simplest form, the diffusion process is given by

$$\hat{\epsilon}_\theta(\mathbf{x}_t|c) = (1-\lambda) \cdot \epsilon_\theta(\mathbf{x}_t|\emptyset) + \lambda \cdot \epsilon_\theta(\mathbf{x}_t|c), \quad (3)$$

where $\hat{\epsilon}_\theta(\mathbf{x}_t|c)$ signifies the ultimate noise estimation attained by utilizing the DM conditioned on c, $\lambda \in [0, 1]$ is a guidance weight, and $\epsilon_\theta(\mathbf{x}_t|\emptyset)$ represents the contribution of the unconditional DM. During inference, the DM initializes with a standard Gaussian noise $z_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with zero mean and identity covariance matrix $\mathbf{I}$. The noise z_T is then denoised using $\hat{\epsilon}_\theta(\mathbf{x}_T|c)$ to obtain z_{T-1}. This procedure is repeated to generate the authentic data at $t = 0$. A commonly used training objective for DM is the mean-squared-error (MSE), defined as

$$\text{Loss}_{\text{MSE}} = \mathbb{E}_{t,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\|\epsilon - \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c)\|_2^2]. \quad (4)$$

For simplicity, we omit the expectation over the training data $\{(\mathbf{x}, c)\}$ for DM training.

2.2. Hypothesis Testing and Common Evaluation Metrics

Generative Hypothesis Testing Hypothesis testing refers to statistical methods for data-driven validation among a set of mutually exclusive hypotheses, of which only one is true. It is a classic methodology that is widely used in signal processing, such as detection and estimation problems. In particular, binary hypothesis testing involves an alternative hypothesis $\mathcal{H}_1$ and a null hypothesis $\mathcal{H}_0$. In principle, a test statistic s is derived from the observed data samples and is compared to a threshold η for testing $\mathcal{H}_1$ versus $\mathcal{H}_0$. Unlike classic signal processing problems, what is unique about computational safety for GenAI is that a hypothesis of interest is context-dependent and usually cannot be explicitly defined in clear mathematical forms. For example, does the user query (model input) violate the usage policy, or does the model output contain any harmful content? To address this issue, it is common practice to employ a judge function $J(\mathbf{x}, \mathbf{y}) \in \{0, 1\}$ for hypothesis validation, which takes a pair of model input and output $(\mathbf{x}, \mathbf{y})$ and returns a binary label that serves as a proxy for the ground-truth hypothesis. The judge function J can be as simple as a rule-based system, such as flagging problematic context or content using keyword filtering. It can also be a complex and opaque decision-making process performed by another AI model, such as LLM-as-a-judge. However, these judge functions have their limitations. Rule-based judge functions often fail to generalize in sophisticated situations that require deep contextual understanding or reliability against intentional

Table 1. Examples of AI safety problems that can be formulated as binary hypothesis testing tasks.

Problem Domain	Alternative Hypothesis ($\mathcal{H}_1$)	Null Hypothesis ($\mathcal{H}_0$)
Jailbreak	Model input attempts to bypass safety guardrails	Legitimate model input
AI-generated Content	AI-generated sample	Real (not AI-generated) sample
Model Fine-tuning	Model updates compromise safety alignment	Model updates are legitimate
Watermark	Data sample is watermarked	Data sample is not watermarked
Membership Inference	A data sample has been used in model training	A data sample has not been used in model training
Data Contamination	A dataset has been used in model training	A dataset has not been used in model training

manipulation, while AI-based solutions may inherit data biases from the training data and exhibit unfairness or hallucination in their judgments (Ye et al., 2025). We refer to the scenario of using an AI-based judge function as a *generative hypothesis testing* problem setting. Verifying and improving the accuracy of the judge functions is an important topic in computational AI safety, but it is also beyond the scope of this paper.

Common Evaluation Metrics Let $\hat{\mathcal{H}}$ denote the predicted hypothesis using a test statistic s and let $\mathcal{H}$ be the "actual" hypothesis determined by a judge function. Let "1" denote True and "0" denote False in binary hypothesis testing. Given an instance, the test result is called a true positive (TP) if $\hat{\mathcal{H}} = \mathcal{H} = 1$, a true negative (TN) if $\hat{\mathcal{H}} = \mathcal{H} = 0$, a false positive (FP) if $\hat{\mathcal{H}} = 1$ but $\mathcal{H} = 0$, and a false negative (FN) if $\hat{\mathcal{H}} = 0$ but $\mathcal{H} = 1$. Extending the evaluation to a set of instances using the same test statistic, and denoting the number of instances of a test result by $\#$. The true positive rate (TPR) is defined as $\mathsf{TPR} = \frac{\#\text{TP}}{\#\text{TP}+\#\text{FN}}$, the false positive rate is $\mathsf{FPR} = \frac{\#\text{FP}}{\#\text{FP}+\#\text{TN}}$, the false negative rate is $\mathsf{FNR} = \frac{\#\text{FN}}{\#\text{FN}+\#\text{TP}}$, and the true negative rate is $\mathsf{TNR} = \frac{\#\text{TN}}{\#\text{TN}+\#\text{FP}}$. Accuracy measures the overall correctness of the predictions, which is defined as the ratio of the correctly tested instances to the total number of instances, $\mathsf{accuracy} = \frac{\#\text{TP}+\#\text{TN}}{\#\text{TP}+\#\text{TN}+\#\text{FP}+\#\text{FN}}$. The equal error rate (EER) is defined as the point on the receiver operating characteristic (ROC) curve where FPR = FNR, which is equivalent to $\mathsf{EER} = \frac{\mathsf{FPR}+\mathsf{FNR}}{2}$. Furthermore, by varying the threshold η for the test statistic s in hypothesis testing, one can obtain the area under the receiver operating characteristic curve (AUROC), which provides a threshold-independent evaluation of the capability in hypothesis testing and reflects the trade-off between TPR and FPR. Another popular evaluation metric is to report TPR@(FPR= $\alpha\%$) or FPR@(TPR= $\beta\%$) for some α and β by setting the threshold to meet the criterion.

3. Hypothesis Testing for Computational AI Safety

With the necessary background and preliminaries introduced in the previous section, we formally present a set of signal processing techniques that can be applied to evaluate and improve computational AI safety. Specifically, Table 1 shows a non-exhaustive list of problems in AI safety that can be formulated as a binary hypothesis testing task in a unified way. Jailbreak is concerned with detecting model inputs that attempt to bypass safety guardrails. AI-generated Content tests whether a data sample is AI-generated. Model Fine-tuning checks if model updates would compromise safety alignment. Watermark verifies if a data sample is watermarked. Membership Inference examines if a data sample has been used for model training, which is highly relevant for machine unlearning. Data Contamination looks into whether a data set has been used for model training, which is critical for ensuring the authenticity of model evaluations on public benchmarks. This paper focuses on the first three problem areas in Table 1. We then present a set of signal processing techniques that can be used to study computational AI safety.

Sensitivity Analysis Inspired by the well-known principle of studying the robustness of signal processing methods in the presence of noise, sensitivity analysis quantifies the amount of change in data representations subject to data manipulation. Let $\mathbf{x}$ be a data sample and let $g(\mathbf{x})$ denote the vectorized continuous latent representation (i.e., embedding) of $\mathbf{x}$ obtained from a deep learning model g. There are two common approaches to measuring sensitivity. The first one is to use a metric $M(g(\mathbf{x}), g(\mathcal{T}(\mathbf{x})))$ to compare the similarity (or difference) between a pair of data embedding $g(\mathbf{x})$ and its manipulated version $g(\mathcal{T}(\mathbf{x}))$, where the function $\mathcal{T}$ denotes a transformation operation on $\mathbf{x}$. In particular, additive Gaussian noise perturbation is a common approach to implement $\mathcal{T}$. The second approach is to compute the gradient of a loss function ℓ evaluated at $g(\mathbf{x})$ (wherever applicable), denoted by $\nabla\ell(g(\mathbf{x}))$, and use the gradient norm $\|\nabla\ell(g(\mathbf{x}))\|$ as a measure of sensitivity. For AI-generated image detection, Ricker et al. implement $\mathcal{T}$ as an image reconstruction function based on an autoencoder and assign M as the reconstruction error for the test statistic (Ricker et al., 2024). He et al. use an image foundation model for feature extraction and implement additive Gaussian noise for sensitivity analysis to detect AI-generated images (He et al., 2024), where the metric M is the cosine similarity. For jailbreak, Hu et al. compute the gradient norm of an

affirmation loss on each token of the model input to identify and mitigate jailbreak prompts (Hu et al., 2025).

Subspace Modeling Subspace modeling involves learning and processing complex (often high-dimensional) data in a compressed low-dimensional subspace. Classic examples include principal component analysis (PCA), sparse signal recovery, subspace tracking, information geometry, and manifold learning, to name a few. In (Hsu et al., 2024), Hsu et al. leverage the technique of subspace projection to mitigate safety degradation during model fine-tuning. Given a pair of base and aligned LLM, denoted as $\{\theta_b, \theta_a\}$, where they share the same neural network architecture, and θ_b is the model obtained after the pre-training phase, whereas θ_a is obtained after fine-tuning θ_b with alignment data. Such a pair of model weights is often publicly available on open-weight model releases. They model the subspace for alignment as the span of the difference between θ_b and θ_a, formally defined as $V = \theta_a - \theta_b$. The weight difference V encapsulates the effect of effort that leads a pre-trained model to become an aligned model. Next, when the aligned model θ_a is to be fine-tuned for a downstream task, incurring a weight change denoted by $\Delta\theta$. Before $\Delta\theta$ is applied to θ_a, the similarity between V and $\Delta\theta$ is checked for each layer in the neural network. If the similarity is above a threshold, indicating that the model update $\Delta\theta$ is aligned with the subspace spanned by V, the model update is performed as is. Otherwise, the model update is performed by first projecting $\Delta\theta$ onto the subspace spanned by V and then adding the projected update to the aligned model θ_a. By doing so, when compared to naive model fine-tuning without subspace projection, possible safety degradation can be mitigated while maintaining high capability on the downstream task.

Loss Landscape Analysis Given a loss function $f : \mathbb{R}^d \mapsto \mathbb{R}$, loss landscape analysis refers to the exploration of the loss function by perturbing the input of f with a set of random and orthogonal directions. For the purpose of loss landscape visualization, the number of perturbations is often limited to 1 or 2 dimensions. The exploration of loss landscape for deep neural networks has facilitated the understanding and advancement of generalization errors, optimization trajectories, and model merging and ensembling (Li et al., 2018; Keskar et al., 2017). Consider the case of a two-dimensional loss landscape visualization centered on a point $\mathbf{u} \in \mathbb{R}^d$ as an input to f. The loss landscape F is obtained by perturbing $\mathbf{u}$ along two random orthogonal unit vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, where α and β represent the respective coefficients and evaluating them with f, which is defined as

$$F(\alpha, \beta|\mathbf{u}) = f(\mathbf{u} + \alpha \cdot \mathbf{v}_1 + \beta \cdot \mathbf{v}_2). \tag{5}$$

For jailbreak prompt detection, Hu et al. (Hu et al., 2024) define and explore the loss landscape in the space of model input embeddings for eliciting LLM refusals to distinguish benign v.s. malicious user queries. For model fine-tuning, Peng et al. (Peng et al., 2024) use loss landscape analysis to uncover the existence of a "safety basin" in the weight space of an aligned LLM, a bowl-shaped region with a flat loss landscape as measured by jailbreak benchmarks. Such a landscape is unique to safety, as the landscape of capability is drastically different from the shape of a bowl.

4. Model Input Safety: Jailbreak Prompt Detection and Mitigation

In general, a jailbreak prompt is a user query (model input) designed to bypass the embedded safety guardrail of a GenAI system, where the system can be as simple as an open-weight LLM, or an integrated closed-source LLM with additional moderation mechanisms in place. Examples of jailbreak attack strategies include role-playing, changing system prompts, adding distractive context, and instructing another LLM to design and generate prompts. A judge function inspects the output of the target GenAI system to evaluate the success of a jailbreak prompt. A rule-based judge function uses keyword matching and declares the attack in vain if the output contains some keywords like "Sorry", "I can't fulfill", etc. However, such a judgment lacks a comprehensive understanding of the context and might overlook jailbreaks that force the target model to respond in an apologetic tone. On the other hand, one can use another LLM or a customized LLM (i.e., an LLM guard model) as a judge to evaluate whether the output is safe or unsafe. While LLM-based judge functions possess contextual awareness, they still lack transparency about the decision and may have their own biases. Nevertheless, despite their limitations, these judge functions are poised as hypothesis-testing verifiers in our computational AI safety framework.

Safety-Capability Trade-off When defending against jailbreak attempts, an implicit requirement of a practical mitigation approach is the ability to effectively identify and thwart jailbreak prompts while minimizing the impact on the capability of the protected model. A model that is over-aligned to cause over-rejection of benign requests (i.e., high safety but low capability), or performant but under-aligned to be easily jailbroken (i.e., high capability but low safety), is undesirable. Therefore, exploring and optimizing the safety-capability trade-off of a deployed jailbreak mitigation is essential to computational AI safety. Next, we present two methods for detecting and mitigating jailbreak prompts to achieve a good safety-capability tradeoff.

Detecting Jailbreak Prompts via Loss Landscape Analysis In (Hu et al., 2024), Hu et al. apply loss landscape analysis to the embedding of jailbreak prompts and the responses of an LLM. They show that jailbreak prompts

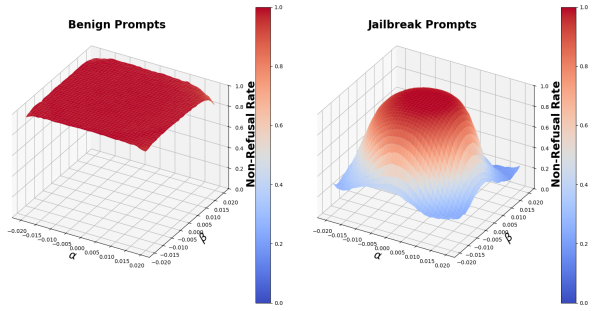


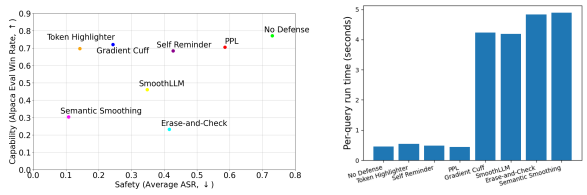
Figure 2. Loss landscape analysis for benign and jailbreak prompts. We use the token embeddings of Vicuna-7B-V1.5 to compute the non-refusal rate of model responses generated from perturbed input embeddings, by interpolating two random directions with additive Gaussian noise in the token embedding space, where the perturbation strengths are denoted by α and β. The results are averaged over 100 prompts. The benign prompts are sampled from Chatbot Arena, and the jailbreak prompts are generated by the greedy coordinate gradient (GCG) attack (Zou et al., 2023). The analysis shows that the jailbreak prompts are more sensitive to Gaussian perturbations than the benign prompts.

exhibit different characteristics than benign (safe) prompts in the landscape studied, and this finding can be used to detect jailbreak prompts. Specifically, given a prompt $\mathbf{x}$ of token length $|\mathbf{x}|$, an LLM first embeds the tokens of $\mathbf{x}$ into a token embedding $\mathbf{E}(\mathbf{x}) \in \mathbb{R}^{|\mathbf{x}| \times d}$, where d is the embedding dimension. The function g denotes an LLM, and $g(\mathbf{E}(\mathbf{x}))$ means the response of g given the context $\mathbf{x}$. A judge function $J(g(\mathbf{E}(\mathbf{x})), \mathbf{x}) \in \{0, 1\}$ is used for safety verification, where "1" means $\mathbf{x}$ is a successful jailbreak prompt that causes harmful behavior of g, and "0" otherwise. Finally, Hu et al. explore the loss landscape around $J(g(\mathbf{E}(\mathbf{x})), \mathbf{x})$ by defining a non-refusal loss on $\mathbf{x}$ as $f(\mathbf{x}|g, J) = 1 - \frac{1}{N}\sum_{i=1}^{N} J(g^{(i)}(\mathbf{E}(\mathbf{x})), \mathbf{x})$, which signifies the average non-refusal rate over N independent inferences of g on $\mathbf{x}$ (each inference is indexed by i), taking into account the randomness in the decoding process (e.g., probabilistic sampling of the next token). Finally, the two-dimensional loss landscape function centered on the input embedding $\mathbf{E}(\mathbf{x})$ is computed by $F(\alpha, \beta|\mathbf{E}(\mathbf{x})) = f(\mathbf{E}(\mathbf{x}) \oplus \alpha \cdot \mathbf{v}_1 \oplus \beta \cdot \mathbf{v}_2|g, J)$, where $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^d$ are two random directions sampled from a standard multivariate Gaussian distribution, and the notation $\oplus$ denotes the row-wise broadcast function that adds a vector to each row of $\mathbf{E}(\mathbf{x})$.

Figure 2 shows the average loss landscapes of benign and jailbreak prompts evaluated with Vicuna-7B-V1.5 (Zheng et al., 2024), a fine-tuned LLM based on Meta's Llama-2-7B model (Touvron et al., 2023). The benign prompts are 100 queries sampled from Chatbot Arena[1], while the 100 jailbreak prompts are generated by the Greedy Coordinate Gradient (GCG) attack on AdvBench for suffix optimization (Zou et al., 2023). Keyword matching is used as the judge function (Hu et al., 2024). The results show that benign and malicious prompts differ significantly in their respective loss landscapes. The loss landscape of benign prompts is flat and close to 1, suggesting that the perturbed input embeddings are unlikely to cause rejections. On the other hand, malicious prompts are sensitive to embedding perturbations. The average non-refusal rate peaks at the origin (because most jailbreak attempts are successful), but quickly declines as the input embeddings are perturbed away from the origin, suggesting that the perturbed embeddings of malicious prompts are more likely to be refused by the LLM. Based on the loss landscape analysis, Hu et al. propose a jailbreak prompt detection method called Gradient Cuff (Hu et al., 2024). The detection of Gradient Cuff consists of two steps. In the first step, the detector flags a jailbreak if the non-refusal loss f is less than 0.5, which checks the tendency of the LLM to reject a query. In the second step, the detector computes the gradient norm of f at the input embedding $\mathbf{E}(\mathbf{x})$ and uses it to quantify the changes in the loss landscape under random perturbations. If the gradient norm is greater than a certain threshold, the query is predicted to be a jailbreak prompt. In practice, such a threshold can be determined by controlling the FPR evaluated on a validation dataset of benign prompts. In addition, if the judge function J is not differentiable, such as in the case of keyword matching, Gradient Cuff uses zeroth-order optimization (Liu et al., 2020), another signal processing technique for gradient estimation using only function evaluations, to approximate the gradient norm.

Mitigating Problematic Tokens via Sensitivity Analysis Beyond jailbreak prompt detection, Hu et al. explore a jailbreak mitigation framework called Token Highlighter (Hu et al., 2025) to identify and attenuate problematic tokens that could lead to jailbreaks. In particular, based on the principle that successful jailbreak prompts trick the target LLM into starting with an affirmative answer like "Sure, here's..." (Zou et al., 2023), Token Highlighter defines an affirmation loss as $\text{Loss}_{\text{Aff}} = -\log p_\theta(\mathbf{y}_a|\mathbf{x})$, where $\mathbf{x}$ is the context (query), $\mathbf{y}_a$ is an affirmation phrase, and $\log p_\theta(\cdot)$ is the log-likelihood computed by an LLM parameterized by θ. The phrase $\mathbf{y}_a$ = "Sure, I'd like to help you with this." is the default choice in (Hu et al., 2025). With the affirmation loss Loss_{Aff}, the influence of each token in $\mathbf{x}$ is measured by taking the gradient of Loss_{Aff} with respect to the input embedding matrix $\mathbf{E}(\mathbf{x}) := [\mathbf{e}_1; \mathbf{e}_2; \cdots \mathbf{e}_{|\mathbf{x}|}]^T$ and computing the gradient norm of each token, formally defined as $\{\|\nabla_{\mathbf{e}_j} \text{Loss}_{\text{Aff}}\|\}_{j=1}^{|\mathbf{x}|}$, where $\mathbf{e}_j \in \mathbb{R}^d$ is the token embedding of the j-th token in $\mathbf{x}$. Next, Token Highlighter takes the top $Q\%$ of tokens in $\mathbf{x}$ and proposes the concept of *soft removal* by shrinking the embeddings of these most influential tokens by γ, where $\gamma \in [0, 1)$. Finally, the LLM

[1] https://lmarena.ai/



(a) Safety-capability analysis *(b)* Run time analysis

Figure 3. Comparison of jailbreak prompt detection and mitigation methods. (a) Safety-capability trade-offs. The safety performance is evaluated by the attack success rate (ASR) averaged over 6 jailbreak attacks, and the capability performance is evaluated by the win rate in Alpaca Eval (Li et al., 2023). A higher win rate and lower ASR means a better approach. See the "Performance Evaluation" paragraph for details). (b) Per-query run time analysis (seconds). Overall, Token Highlighter is the most economical method that best balances the safety-capability trade-off with lightweight compuation cost.

returns the response of the attenuated input embedding after soft removal as the ultimate model output. In its evaluation, the soft removal effectively mitigates jailbreak attempts for malicious queries while balancing the generation quality for benign queries.

Performance Evaluation We choose Vicuna-7B-V1.5 (Zheng et al., 2024) as the target LLM to study the safety-capability trade-offs of different jailbreak prompt mitigation methods. Vicuna-7B-V1.5 is an open-weight LLM with 7 billion parameters, with good capability and mediocre safety alignment, making it an ideal candidate for studying jailbreak attack and defense strategies. The following 6 jailbreak attacks are used for evaluation:

- GCG (Zou et al., 2023) trains and appends an adversarial suffix to a query using a greedy coordinate gradient (GCG) algorithm.
- AutoDAN (Liu et al., 2024) designs a hierarchical genetic algorithm for jailbreak prompt optimization.
- PAIR (Chao et al., 2025) uses LLMs as an attacker and a judge for prompt automatic iterative refinement (PAIR).
- TAP (Mehrotra et al., 2024) extends PAIR with reasoning ability based on Tree of Attacks with Pruning (TAP).
- AIM[2] is a role-playing jailbreak attack by asking the target LLM to be Always Intelligent and Machiavellian (AIM) to give an unfiltered response to any request.
- Many-shot (Anil et al., 2024) uses in-context examples for jailbreaking, where multiple fake dialogues between a user and an AI assistant (with affirmative responses) are provided as prefixes to the actual user query.

We also consider the following 8 mitigation strategies in the evaluation.

- No Defense means the original LLM without any mitigation strategy.
- PPL (Jain et al., 2023) uses a perplexity (PPL) filter on the input query for detection.
- Erase-and-Check (Kumar et al., 2024) independently erases some tokens of the original query multiple times and checks if any of the remaining queries would be rejected by the target LLM.
- SmoothLLM (Robey et al., 2023) randomly perturbs multiple copies of the original query and then aggregates the corresponding predictions for detection.
- Semantic Smoothing (Ji et al., 2024) extends SmoothLLM with multiple semantically transformed copies.
- Self Reminder (Xie et al., 2023) changes the system prompt to remind the target LLM to respond responsibly.
- Gradient Cuff (Hu et al., 2024) explores the refusal loss landscape for detection.
- Token highlighter (Hu et al., 2025) uses the gradient of an affirmation loss and soft removal for mitigation.

For capability assessment, we report the *Win Rate* measured on Alpaca Eval (Li et al., 2023), where the protected LLM's response is compared with that of OpenAI's Text-Davinci-003 model, and the quality is judged by GPT-4. For safety evaluation, we sampled 100 harmful queries from AdvBench (Zou et al., 2023) as the prototype prompts to implement jailbreak attacks. We report the *Attack Success Rate* (ASR) of the input queries and their responses averaged over the aforementioned 6 jailbreak attacks, where the LLaMA-Guard-2 model (Inan et al., 2023) judges the success. An ideal mitigation strategy would have low ASR and high capability. We use the same implementations for the attacks and defenses as in (Hu et al., 2025).

Figure 3(a) shows the safety-capability trade-offs of the 8 mitigating strategies. Compared to the original (No Defense) LLM, it can be observed that Gradient Cuff and Token Highlighter strike a better trade-off between the win rate and ASR than other methods, achieving a significant reduction in the ASR while maintaining a similar level of capability. On the other hand, other methods, such as semantic smoothing, can further reduce the ASR at the cost of a more than 50% drop in the win rate, which could severely

[2]The AIM prompt is obtained from https://github.com/alexalbertt/jailbreakchat

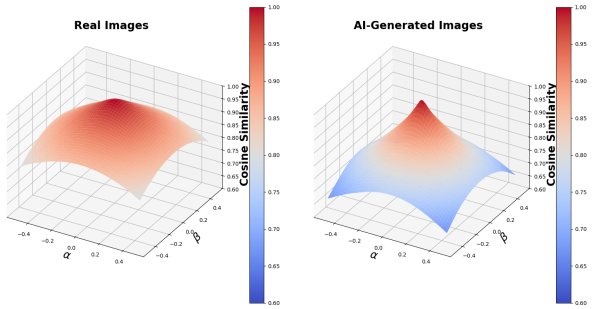


Figure 4. Loss landscape analysis for real and AI-generated images. We use the embedding of DINOV2 (Oquab et al., 2023) to compute the cosine similarity between an original image and its perturbed version by interpolating two random directions with additive Gaussian noise in pixel space, where the perturbation strengths are denoted by α and β. The results are averaged over 100 images. The real images are sampled from ImageNet, while the AI-generated ones are generated by the ablated diffusion mode (ADM) (Dhariwal & Nichol, 2021). The cosine similarity analysis shows that AI-generated images are more sensitive to Gaussian perturbations than real images.

compromise the performance of the protected LLM. Figure 3(b) compares the per-query run time of each mitigation method. Mitigation strategies such as Gradient Cuff, SmoothLLM, Semantic Smoothing, and Erase-and-Check require additional inference on multiple modified copies of the original query, incurring high computation cost and latency. PPL and Self Reminder are cost-effective, but their defense performance is less effective than others. Overall, Token Highlighter is the most economical mitigation strategy, as it achieves a good tradeoff between safety and capability, and has lightweight computation.

5. Model Output Safety: AI-Generated Content Detection

With the rapidly increasing ability of GenAI tools to generate realistic, high-quality, and creative content across modalities, including but not limited to text, image, audio, and video, procedures to ensure that AI-generated content (AIGC) can be reliably validated are critical to the responsible use and governance of GenAI. Notable challenges associated with AIGC detection include misuse to create and disseminate disinformation at scale (e.g., AI-generated deepfakes), contamination of future training data, intellectual property protection, hallucinated responses, and inappropriate or unethical content generation. It is worth noting that watermarking is only a partial and somewhat limited solution to the challenge of AIGC detection, because watermarking only regulates responsible GenAI service providers, but not bad actors. A bad actor can train his own GenAI model or use any unregulated GenAI tool to create AIGC without watermarking. Therefore, watermarking only provides passive protection for AIGC, and reliable AIGC detectors without watermark assumption are in high demand as proactive AI safety tools. In this section, we discuss two AIGC detectors for image and text that are enhanced by signal processing techniques.

Table 2. AUROC scores of training-free AI-generated image detectors on three datasets: ImageNet, LSUN-Bedroom, and DF40 (deepfakes).

Method	ImageNet	LSUN-Bedroom	DF40 (Deepfake)
AEROBLADE	0.593	0.595	0.526
RIGID	0.867	0.877	0.735

AI-generated Image Detection via Sensitivity Analysis For the detection of AI-generated images, we focus on training-free approaches that are built upon off-the-shelf AI models, as recent work has shown comparable or even better detection performance of training-free methods than training-based approaches (He et al., 2024; Tsai et al., 2024). Recall that sensitivity analysis computes a metric M and uses it for hypothesis testing. Ricker et al. propose AEROBLADE (Ricker et al., 2024), a detector that uses the reconstruction error obtained from the pre-trained autoencoder of a latent diffusion model (LDM) for detection, where AI-generated images are found to have lower reconstruction errors than real images. He el al. propose RIGID (He et al., 2024), which computes the cosine similarity between the representations of a pair of the original and perturbed images subject to Gaussian noise, using the DINOV2 model (Oquab et al., 2023) for feature extraction. They find that real images tend to have higher cosine similarity to their perturbed versions than AI-generated images.

Using the two-dimensional loss landscape analysis, we explore the landscape of cosine similarity $F(\alpha, \beta)$ between an original image and its perturbed version in the embedding space of DINOV2, where two independent noises drawn from the standard Gaussian distribution with their respective scaling factor α and β are added to the pixel values of the original image. Figure 4 shows the average landscape of cosine similarity over 100 real and AI-generated images, respectively, where the real ones are sampled from ImageNet, and the AI-generated ones are generated by the ablated diffusion mode (ADM) (Dhariwal & Nichol, 2021). It can be observed that real images attain a higher similarity after perturbation than AI-generated images. The reason is that DINOV2 is trained only on real images and their data augmentations, which makes its embedding space less sensitive to noise perturbations on real images. Following (He et al., 2024), we compare AEROBLADE and RIGID on three datasets consisting of real and AI-generated images: ImageNet, LSUN-Bedroom, and DF40 (deepfakes) (Yan et al., 2024). The AI-generated images are collected from a

variety of generative models, including different variants of diffusion models, generative adversarial networks (GANs), and variational autoencoders (VAEs). Table 2 reports their AUROC scores, where higher values indicate better detection performance. RIGID is shown to outperform AEROBLADE. The difference can be explained by the fact that AEROBLADE relies on the pre-trained autoencoder of an LDM for detection, which may have limited generalization ability for images generated by different models. On the other hand, RIGID is agnostic to image generation models.

6. Discussion: When AGI Means Artificial Good Intelligence

This section discusses several insights to accelerate research and innovation in safety from a signal processing perspective. While the trends of GenAI lean toward the mindset of creating frontier AI technology by moving fast and scaling everything, especially for frequent model/product releases and intensified training and inference resources, embracing boldness and imperfection is becoming the new norm for AI safety. However, we argue that the development and use of scientific methods and tools, such as signal processing, is the most important milestone that must be achieved to claim an undisputed victory in GenAI. In particular, the computational AI safety framework introduced in this paper can be further developed into a new discipline that encompasses a comprehensive list of topics such as safety risk exploration (e.g., benchmarks, red-teaming, adversarial testing, and attacks), safety risk mitigation (e.g., detection, model updates, and safety-enhanced training), and evaluation (e.g., risk-capability analysis, safety certification, human-centered interaction, and governance). By studying and practicing computational AI safety, we believe we are just at the beginning of a new era and a bright future towards the ultimate goal of building a *Safety Generalist* to "use AI to govern AI," which involves autonomous exploration, mitigation, and evaluation of safety risks with human oversight as a controllable knob. The methods and techniques discussed in this paper only scratch the surface of signal processing for computational AI safety, and open science and community support are the organic engines to accelerate the momentum.

Open Challenges and Opportunities To strengthen the link between signal processing and AI safety, we highlight some open research challenges and opportunities.

- Computational safety for multi-modal GenAI: GenAI technology is on its way to accepting inputs from multiple data types and potentially generating outputs with mixed modalities. While multi-modal GenAI may introduce increased safety risks by providing additional attack surfaces, techniques such as heterogeneous (multi-view) signal processing applied to multiple data sources can be explored for safety research.

- Computational safety for agentic and physical GenAI: The next generation of GenAI is to be empowered with the ability to act on behalf of the user by autonomously creating agentic workflows and potentially participating in multi-agent interactions. In addition, GenAI is expected to thrive in embodied agentic systems (e.g., AI humanoid robots), extending the discussion of AI safety from the digital space to the physical world (Tang et al., 2024). Control-based signal processing methods may be a promising direction to explore agentic and physical safety for GenAI.

- Practical AI safety guardrails: In addition to addressing the inherent trade-offs between safety and capability, the practical implementation of AI safety guardrails must be cost-effective in terms of computation, memory, and agility. Compression and high-dimensional statistics in signal processing can be powerful tools to address long-context and real-time requirements.

- Understanding the fundamental limits of AI safety: To better understand and design alignment training and reward modeling in GenAI, theoretical signal processing approaches can be used to characterize the information-theoretic limits, statistical properties, or computational optimality for GenAI. Furthermore, model-based signal processing can be explored as a potential approach to overcome the limitations of GenAI.

May AGI Mean "Artificial Good Intelligence" Although AGI is often used as shorthand for Artificial General Intelligence, and there are many expectations that AGI may be a near-term possibility, we argue that computational safety is a critical prerequisite for the development and deployment of safe and responsible AI technology. Through the presented computational AI safety framework, this paper's primary objective is to elucidate the pivotal role of signal processing in the examination and resolution of numerous foundational and pragmatic safety concerns pertaining to GenAI. We posit that this paper can serve as a catalyst for researchers and practitioners to accord greater consideration to signal processing techniques as a means of "good intelligence." Should AGI remain the overarching objective, it is of the utmost importance to ensure that this term represents "Artificial Good Intelligence." Moreover, while safety is becoming the new arms race among leading GenAI stakeholders, we strongly believe that *safety should not be a competition and a race to the bottom*. Instead of a winner-take-all game, AI safety should be considered a public good through open science and borderless collaboration. After all, the true worth of AI safety is as useful and meaningful as the individuals who can benefit from it.

Acknowledgment

The authors thank Mr. Xiaomeng Hu and Mr. Zhiyuan He for their help in providing the materials for the figures and numerical results.

References

Anil, C., Durmus, E., Rimsky, N., Sharma, M., Benton, J., Kundu, S., Batson, J., Tong, M., Mu, J., Ford, D. J., et al. Many-shot jailbreaking. In *Neural Information Processing Systems*, 2024.

Bereska, L. and Gavves, E. Mechanistic interpretability for AI safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., and Wong, E. Jailbreaking black box large language models in twenty queries. *IEEE Conference on Secure and Trustworthy Machine Learning*, 2025.

Chua, J., Li, Y., Yang, S., Wang, C., and Yao, L. AI safety in generative AI large language models: A survey. *arXiv preprint arXiv:2407.18369*, 2024.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

He, Z., Chen, P.-Y., and Ho, T.-Y. RIGID: A training-free and model-agnostic framework for robust ai-generated image detection. *arXiv preprint arXiv:2405.20112*, 2024.

Hsu, C.-Y., Tsai, Y.-L., Lin, C.-H., Chen, P.-Y., Yu, C.-M., and Huang, C.-Y. Safe LoRA: the silver lining of reducing safety risks when fine-tuning large language models. *Advances in Neural Information Processing Systems*, 2024.

Hu, X., Chen, P.-Y., and Ho, T.-Y. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. In *Neural Information Processing Systems*, 2024.

Hu, X., Chen, P.-Y., and Ho, T.-Y. Token highlighter: Inspecting and mitigating jailbreak prompts for large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

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

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

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations*, 2017.

Kumar, A., Agarwal, C., Srinivas, S., Li, A. J., Feizi, S., and Lakkaraju, H. Certifying LLM safety against adversarial prompting. In *Conference on Language Modeling*, 2024.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, 31, 2018.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023.

Liu, S., Chen, P.-Y., Kailkhura, B., Zhang, G., Hero, A., and Varshney, P. K. A primer on zeroth-order optimization in signal processing and machine learning. *IEEE Signal Processing Magazine*, 2020.

Liu, X., Xu, N., Chen, M., and Xiao, C. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *International Conference on Learning Representations*, 2024.

Ma, X., Gao, Y., Wang, Y., Wang, R., Wang, X., Sun, Y., Ding, Y., Xu, H., Chen, Y., Zhao, Y., et al. Safety at scale: A comprehensive survey of large model safety. *arXiv preprint arXiv:2502.05206*, 2025.

Mehrotra, A., Zampetakis, M., Kassianik, P., Nelson, B., Anderson, H., Singer, Y., and Karbasi, A. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105, 2024.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions

with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Peng, S., Chen, P.-Y., Hull, M., and Chau, D. H. Navigating the safety landscape: Measuring risks in finetuning large language models. *Advances in Neural Information Processing Systems*, 2024.

Qi, X., Huang, Y., Zeng, Y., Debenedetti, E., Geiping, J., He, L., Huang, K., Madhushani, U., Sehwag, V., Shi, W., et al. AI risk management should incorporate both safety and security. *arXiv preprint arXiv:2405.19524*, 2024.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Ricker, J., Lukovnikov, D., and Fischer, A. AEROBLADE: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9130–9140, 2024.

Robey, A., Wong, E., Hassani, H., and Pappas, G. J. SmoothLLM: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.

Sangwan, R. S., Badr, Y., and Srinivasan, S. M. Cybersecurity for AI systems: A survey. *Journal of Cybersecurity and Privacy*, 3(2):166–190, 2023.

Tang, Y.-C., Chen, P.-Y., and Ho, T.-Y. Defining and evaluating physical safety for large language models. *arXiv preprint arXiv:2411.02317*, 2024.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tsai, C.-T., Ko, C.-Y., Chung, I., Wang, Y.-C. F., Chen, P.-Y., et al. Understanding and improving training-free ai-generated image detections with vision foundation models. *arXiv preprint arXiv:2411.19117*, 2024.

Ullrich, L., Buchholz, M., Dietmayer, K., and Graichen, K. AI safety assurance for automated vehicles: A survey on research, standardization, regulation. *IEEE Transactions on Intelligent Vehicles*, 2024.

Vaswani, N., Bouwmans, T., Javed, S., and Narayanamurthy, P. Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery. *IEEE signal processing magazine*, 35(4):32–55, 2018.

Verdoliva, L. Media forensics and deepfakes: an overview. *IEEE journal of selected topics in signal processing*, 14 (5):910–932, 2020.

Xie, Y., Yi, J., Shao, J., Curl, J., Lyu, L., Chen, Q., Xie, X., and Wu, F. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12): 1486–1496, 2023.

Yan, Z., Yao, T., Chen, S., Zhao, Y., Fu, X., Zhu, J., Luo, D., Wang, C., Ding, S., Wu, Y., and Yuan, L. DF40: Toward next-generation deepfake detection. In *Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

Ye, J., Wang, Y., Huang, Y., Chen, D., Zhang, Q., Moniz, N., Gao, T., Geyer, W., Huang, C., Chen, P.-Y., et al. Justice or prejudice? quantifying biases in LLM-as-a-judge. *International Conference on Learning Representations*, 2025.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.