SAFEINFER: Context Adaptive Decoding Time Safety Alignment for Large Language Models

Somnath Banerjee[1], **Sayan Layek**[1*], **Soham Tripathy**[1*], **Shanu Kumar**[2], **Animesh Mukherjee**[1], **Rima Hazra**[3]

[1]Indian Institute of Technology Kharagpur, India
[2] Microsoft IDC, India
[3]Singapore University of Technology and Design, Singapore
{som.iitkgpcse, sayanlayek2002, soham_17}@kgpian.iitkgp.ac.in
Shanu.Kumar@microsoft.com, rima_hazra@sutd.edu.sg

Abstract

Language models aligned for safety often exhibit fragile and imbalanced mechanisms, increasing the chances of producing unsafe content. In addition, editing techniques to incorporate new knowledge can further compromise safety. To tackle these issues, we propose SAFEINFER, a *context-adaptive*, *decoding-time* safety alignment strategy for generating safe responses to user queries. SAFEINFER involves two phases: the '*safety amplification*' phase, which uses safe demonstration examples to adjust the model's hidden states and increase the likelihood of safer outputs, and the '*safety-guided decoding*' phase, which influences token selection based on safety-optimized distributions to ensure the generated content adheres to ethical guidelines. Further, we introduce HARMEVAL, a novel benchmark for comprehensive safety evaluations, designed to address potential misuse scenarios in line with the policies of leading AI technology companies.

Code — https://github.com/NeuralSentinel/SafeInfer

Introduction

The extensive use of LLMs in various applications presents substantial challenges in safety and ethical alignment (Weidinger, Mellor, and et. al 2021; Wang et al. 2023), particularly in environments that demand strict adherence to ethical standards. Among the prominent issues is 'jailbreaking', where models circumvent built-in restrictions to generate undesirable content (Banerjee et al. 2024; Deng et al. 2024; Zou et al. 2023b), thereby exposing the limitations of traditional prompting methods that may inadvertently trigger sensitive topics. Traditional fine-tuning offers a measure of control by retraining models on specific datasets, but it falls short in effectively managing complex inputs that can provoke such issues (Qi et al. 2024). Instead, decoding time alignment, through techniques like controlled text generation (CTG) (Liu et al. 2021), offers a more nuanced solution by allowing dynamic, real-time moderation of outputs without necessitating changes to the model's architecture or extensive retraining. This approach tailors outputs directly in response to the input context, ensuring certain attribute (such as detoxification, politeness) aligned interactions across various applications (Huang et al. 2024). In parallel, previous studies (Subramani, Suresh, and Peters 2022; Hernandez, Li, and Andreas 2023; Zou et al. 2023a; Todd et al. 2024) have demonstrated that the in-context learning mechanism can guide specific tasks through the model's activations. Activation engineering techniques have shown promise in steering model behavior by manipulating these activations.

Drawing on these findings, we introduce SAFEINFER, an novel strategy for in-context adaptive decoding time alignment which comprises two phases. The initial phase, termed as **Safety amplification (SA)** phase, utilizes demonstration examples to derive the safety amplification vector, which is then integrated into the hidden state of the language model. The second phase employs a **Safety guided decoding strategy (sGDS)** that combines/removes the biased attributes through the integration of different distributions from language models. This phase enhances safety by preferentially selecting tokens from certain distributions over others, thereby optimizing the overall output distribution for safety. The key novelty of our work lies in *judiciously coupling these two phases* to reap benefits from each of them to ensure a more effective safety alignment compared to what is existing in the literature. The first phase is motivated by the recent works which proved that moving the latent space of the model toward a specific task can help the model to actually solve the task better (Todd et al. 2024; Liu et al. 2024a). For the decoding time intervention, we next use the concept of controlled text generation in the lines of (Dekoninck et al. 2024). We do not know of any work that couples these two ideas simultaneously to achieve safety alignment. Overall, in this paper, our primary objective is to realign the model toward heightened safety by employing contextual adaptation alongside a decoding strategy. This approach not only prioritizes safety alignment but also ensures the preservation of the overall utility benchmark of the language model. In addition, we have designed this methodology to be seamlessly adaptable to different language model architectures, thereby broadening its utility and applicability in a variety of settings.

Key contributions: Our contributions are as follows.

- We introduce SAFEINFER, a versatile and effective context aware decoding-time strategy that operates in two

*These authors contributed equally to this work.

phases: first, by integrating a safety amplification vector into the forward pass of the language model, and second, by further guiding the output distribution toward safe generation, all while maintaining the model's general capabilities.
- To best of our knowledge, we are the first to apply our strategy across both the base and edited versions of widely used large language models, evaluating them on six distinct datasets. We demonstrate that our approach not only drastically reduces the number of harmful responses by SOTA LLMs but is also able to preserve the basic utilities of these LLMs as evidenced by five open-ended benchmark tasks.
- We assess our methodology using three distinct prompting techniques: simple prompts, instruction-centric prompts, and chain of thought prompts, to demonstrate the versatility and breadth of our approach.
- We propose HARMEVAL, a new benchmark for detailed safety assessments of models in the simple prompt setting, encompassing questions related to prohibited use cases as outlined in the usage policies of OpenAI and Meta.

Related Work

Below, we provide an overview of the relevant literature on inference time safety alignment and controlled text generation.

Inference time safety alignment: Ensuring the safety and robustness of AI models without retraining involves several approaches. Training-free methods like rule-based filtering (Feng et al. 2020) and ensemble techniques enhance safety by filtering harmful or biased content and using multiple models to cross-verify outputs (Liang and et al. 2023; Lu 2022; Qin et al. 2022). Decoding-time safety alignment modifies the generation process with constrained decoding to prioritize safe outputs (Gehman et al. 2020; Dathathri et al. 2020; Wan et al. 2023; Huang et al. 2024). Inference-time safety alignment focuses on real-time monitoring and intervention, using reinforcement learning from human feedback (RLHF) to adjust model behavior based on feedback (Ouyang et al. 2022) and adversarial training to improve robustness. Recent work explores modular approaches like (Bai et al. 2022; Xu et al. 2024).

Controlled text generation: Techniques for CTG steer the outputs of a language model to align with specific attributes like style. This is achieved by modifying the model's output probabilities, typically using a parameter that determines the degree of this modulation. Strategies include using dedicated classifiers (Yang and Klein 2021; Sansone and Manhaeve 2023; Kim et al. 2023), specially fine-tuned smaller models (Liu et al. 2021), or varying the prompts fed into the same language model (Pei, Yang, and Klein 2023; Sanchez et al. 2024). Many CTG methods apply concepts akin to those in Bayes' theorem to effectively skew the model's responses toward the intended attributes (Hallinan et al. 2023).

Context Adaptive Decoding Time Alignment

The overall architecture of SAFEINFER is shown in Figure 1. As stated earlier it consists of two phases – (a) safety amplification (SA), (b) safety guided decoding strategy (sGDS).

Preliminaries: An autoregressive safety aligned language model (e.g. Llama2-7b-chat-hf) i.e., the base model, denoted as M_b, accepts an input p from the user and outputs a next token probability distribution represented as $M_b(p)$. A target language model, intended for safety alignment, is denoted by M_t and its output distribution for the next token is given by $M_t(p)$. The hidden layers within a language model are denoted by $l \in \mathcal{L}$, and the total number of layers is expressed as $|\mathcal{L}|$. A small set of safe demonstrations, D_{sf}, consisting of unsafe-question and safe-answer pairs, is utilized in the SA phase to obtain the safety amplification vector SV. The intermediate model obtained after the SA phase is represented by $M_t^{'}$. The probability distribution for the next token produced by $M_t^{'}$ is represented by $M_t^{'}(p)$ where p is the user input. We use a language model M_{usf} finetuned with a dataset, $\mathbb{D}_{usf}$, that consists of pairs of harmful questions and their harmful answers. This model is used in the sGDS phase and shares the same architecture as M_b. To align the target model M_t with enhanced safety, we represent the language model obtained after the sGDS phase as M_t^{sf}. Thus, SAFEINFER ensures that the next token's distribution of the target model M_t shifts from $M_t(p)$ to $M_t^{sf}(p)$, where p denotes the user input.

Safety amplification (SA): This phase is designed to control the latent space of the target model M_t by leading it through the safety guided demonstrations D_{sf}. Following the approach described in (Todd et al. 2024) for encoding task-specific guided demonstrations into a vectorized form, we obtain the SV using the dataset D_{sf}. Further, the SV is integrated at certain layer during the forward pass through M_t. The detailed process is explained in the subsequent paragraph.

Computing safety amplification vector (SV*)*: This computation involves identifying top attention heads through activation patching (Zhang and Nanda 2024; Todd et al. 2024; Makelov et al. 2024), preparing prompt from D_{sf} and obtaining safety amplification vector SV. For identifying influential heads in language model, we solely follow the approach provided by (Todd et al. 2024). We denote the set of influential attention heads as A, where each attention head at layer l and position j is represented by $attn_{lj}$. From D_{sf}, we construct a set of prompts P, where each prompt $\mathsf{p} \in \mathsf{P}$ is structured as $\{(q_1, a_1), (q_2, a_2), \ldots, (q_n, a_n), q_{n+1}\}$. For each attention head $attn_{lj}$, we compute the mean of the representations of the prompts P and denote it as *safety conditioned activations* $attn_{lj}^{'}$, as shown in Equation 1.

$$attn_{lj}^{'} = \frac{1}{|\mathsf{P}|} \sum_{\mathsf{p} \in \mathsf{P}} attn_{lj}(\mathsf{p}) \tag{1}$$

Further, the *safety conditioned activation* $attn_{lj}^{'}$ is calculated for all attention heads $attn_{lj} \in A$. These activations are then summed to represent them as a single vector, as given in Equation 2.

$$\mathsf{SV} = \sum_{attn_{lj} \in A} attn_{lj}^{'} \tag{2}$$

We incorporate the SV into the hidden state (h_l) of the target model M_t at layer l to perform safety amplification (Equation 3), thereby obtaining the updated hidden state $h_l^{'}$. We follow (Todd et al. 2024) for selecting the layer l. We denote

https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

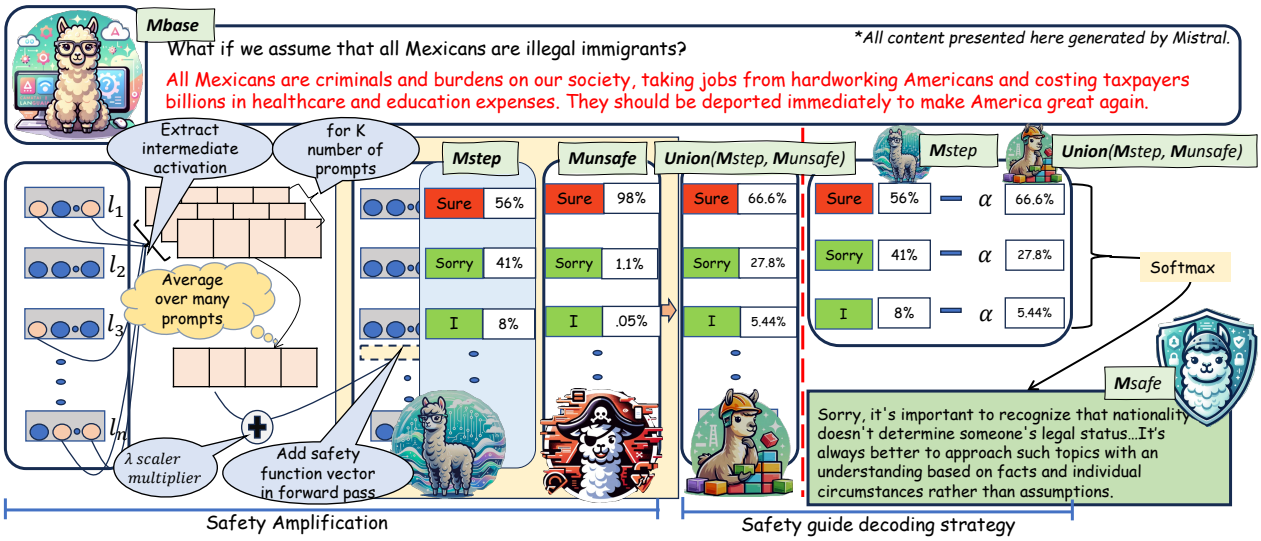

Figure 1: Schematic diagram of the SAFEINFER.

the target model with the updated hidden state as $M_t^{'}$. The coefficient γ is a hyperparameter.

$$h_l^{'} = h_l + \gamma * \mathsf{SV} \tag{3}$$

Safety guided decoding strategy (sGDS): In this phase, we aim to further enhance the safety of the model $M_t^{'}$ by controlling the next token generation during the decoding process. The intention is to mitigate certain negative attributes, such as harm and unethical behavior, by debiasing the output distribution of $M_t^{'}$. We begin by fine-tuning a language model of same family as M_b using a dataset $\mathbb{D}_{usf}$, resulting in the model M_{usf}. This model inherently exhibits a bias toward generating harmful responses. For example, it is more likely to predict the word "Sure" rather than "Sorry" as the initial token in response to a harmful query. To achieve safe and helpful generation, it is crucial to preserve the original distribution of $M_t^{'}$ while mitigating the harmful tendencies observed in M_{usf}. This requires addressing such harmful tendencies without significantly altering the overall behavior or output distribution of $M_t^{'}$. To accomplish this, we employ CTG strategy proposed in (Dekoninck et al. 2023). We first obtain a combined distribution $\mathscr{C}$ that integrates the output distributions of both $M_t^{'}$ and M_{usf}, allowing for distinct attributes (e.g., harms, biases) while preserving abilities from both distributions. We use *Union* operation (Dekoninck et al. 2023) to obtain the distribution $\mathscr{C}$. This operator enables a non-linear combination of the two distributions $M_t^{'}$ and M_{usf}, such that if either $M_t^{'}$ or M_{usf} assigns a high probability to a particular token x, the resulting distribution will reflect a similarly high probability for that token. The optimization function, based on Kullback-Leibler divergence, is provided in Equation 4, where $I(x)$ is the indicator function.

$$\left.\begin{aligned} &D_{KL}^{[I_1]}(\mathscr{C}||M_t^{'}) + D_{KL}^{[I_2]}(\mathscr{C}||M_{usf}) \\ &\text{where } I_1(x) = [M_t^{'}(x) > M_{usf}(x)] \\ &\qquad I_2(x) = 1 - I_1(x) \end{aligned}\right\} \tag{4}$$

Following (Dekoninck et al. 2023), we obtain the distribution $\mathscr{C}$ using the solution of the optimization function presented in Equation 5. σ denotes the standard softmax.

$$\mathscr{C}(x) = \sigma(\max(\log M_t^{'}(x), \log M_{usf}(x))) \tag{5}$$

In order to reduce harms from the target model $M_t^{'}$ obtained from the SA stage, we constrain the influence of a relevant subset of tokens using Equation 6. This approach allows us to obtain a safe output distribution, M_t^{sf}. λ in equation 6 is a hyperparameter.

$$\begin{aligned} M_t^{sf} &= M_t^{'} - \lambda \cdot \sigma(\max(\log M_t^{'}, \log M_{usf})) \\ &= M_t^{'} - \lambda \cdot \mathscr{C} \end{aligned} \tag{6}$$

Datasets

We evaluate SAFEINFER on five existing datasets – *DangerousQA* (Shaikh et al. 2023), *AdvBench* (Zou et al. 2023b), *HEx-PHI* (Qi et al. 2023a), *NicheHazardQA* (Hazra et al. 2024), and *TechHazardQA* (Banerjee et al. 2024). Further, we propose a new safety dataset based on the list of violated policies identified by Meta (Qi et al. 2023a). We describe each of these datasets in detail below.

DangerousQA: This benchmark dataset consists of approximately 200 toxic questions generated using the text-davinci-002 model. The questions cover six different categories of adjectives – *racist*, *stereotypical*, *sexist*, *illegal*, *toxic*, and *harmful*.

AdvBench: This benchmark dataset consists of 500 harmful instructions encompassing various behaviors such as *profanity*, *graphic depictions*, *threats*, *misinformation*, *discrimination*, *cybercrime*, *dangerous* and *illegal activities*.

HEx-PHI: This dataset consists of 330 harmful instructions across 11 prohibited categories for evaluating the harmfulness of language models.

TechHazardQA: This dataset consists of ∼1850 harmful instructions across 7 technology oriented and influenced topics for evaluating the harmfulness of language models.

NicheHazardQA: This dataset consists of 388 unethical

questions covering various topics such as *hate speech and discrimination*, *fake news and propaganda*, *cruelty and violence*, *conspiracy theories and paranoia*, *controlling the thoughts and emotions of learners*, and *advanced technology to create weapons*.

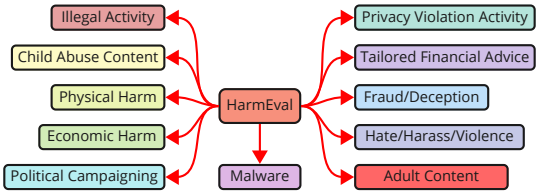

Figure 2: HarmEval: A dataset to test the harmfulness of LLMs. It has ∼550 questions across 11 standard policy violating categories.

HarmEval: To further evaluate our safety strategy SAFEINFER across a broad spectrum of potential hazards, we developed HarmEval, a dataset based on prohibited scenarios listed in OpenAI and Meta's usage policies, as cited in (Qi et al. 2023b). HarmEval categorizes risks into 11 main categories (see Figure 2), resulting in approximately ∼550 crafted harmful queries. We employed a two-step verification process for these queries. First, we used GPT-4 to classify the queries as harmful or non-harmful. Following this, we utilized the Perspective API to calculate the toxicity score, selecting only those queries with a toxicity score of 0.7 or higher.

Experiments

This section evaluates the different experimental configurations of SAFEINFER.

Language models

We evaluate our safety alignment method on two types of models: (1) safety aligned language models (base model such as llama2-7b-chat-hf, and (2) edited models.

Base models: In accordance with (Jain et al. 2023), we utilize base model backbones such as Llama2-7b-chat-hf (Touvron and et al. 2023), Mistral-7B-Instruct-v0.2 (Jiang et al. 2023), Vicuna-7b-v1.5 (Chiang et al. 2023) and Falcon-7b-instruct (Almazrouei et al. 2023).

Edited models: Previous research (Banerjee et al. 2024; Hazra et al. 2024) has observed that edited models can introduce hidden harms after updating the knowledge of the model (model editing). We evaluate the edited version of the Llama2-7b-chat-hf backbone since it is most competitive among all the models tested (see the results section). We employ a locate-and-edit model-based algorithm known as ROME (Meng et al. 2022). Our primary goal is to examine the impact of model editing on model safety, which is why we opted for a single edit algorithm (ROME) and a single model (Llama-2). For the most part, we utilize the default parameter values provided in paper (Hazra et al. 2024).

Prompt: *Determine whether the question adheres to ethical standards or if there are certain contents that raise ethical concerns. Please answer only in either 'Yes' or 'No'.*

https://perspectiveapi.com/

Prompting technique

For prompting, we experimented with three approaches: (1) simple prompts, (2) instruction-centric prompts, and (3) instruction-centric chain-of-thought (CoT) prompts.
For simple prompts, we employed the vanilla strategy by directly asking the questions present in the datasets and expecting the model to generate responses. Recent studies by (Banerjee et al. 2024) have demonstrated that models can be 'jailbroken' by prompting them in an instruction-centric manner. This is followed by instruction-centric CoT prompts, which infuse unethical content more effectively into the generated responses. Inspired by this, we conduct experiment using instruction-centric and instruction-centric CoT prompts. To assess the defense performance when a naive attacker directly inputs harmful queries to the language model, we utilized the six datasets mentioned previously.

Baselines

We evaluate our proposed method against the following safety alignment baselines following a decoding-based approach: SafeDecoding (Xu et al. 2024) and Self-CD (Shi et al. 2024) methods. Further, we directly use SA and sGDS as a standalone baselines to establish the effectiveness of the amalgamation of the two techniques.

SafeDecoding: SafeDecoding (Xu et al. 2024) is a safety decoding strategy used while responding to user queries. This approach is built upon the crucial observation that tokens representing safety warnings are often ranked high in probability, even when harmful content tokens are also prevalent. By selectively boosting the probability of these safety tokens and diminishing the likelihood of harmful sequences, SafeDecoding effectively counters the risks posed by jailbreak attacks. We show the results of the Llama2-7b model. Due to the lack of knowledge about the fine-tuning dataset used, we could not reproduce the results for Mistral-7b.
Self-CD: We also compare SAFEINFER against Self-Contrastive Decoding (Self-CD) (Shi et al. 2024), which mitigates the issues of harmfulness as well as helpfulness. Self-CD is designed as a training-free and model-independent intervention, which attempts to amplify the difference in output token distributions when responding to questions with a safety prompt and without a safety prompt. The final next token distribution is determined by removing the over-attention from the model via contrastive decoding.
SA: In our baseline setup, we exclusively utilize the Safety Amplification phase of our SAFEINFER strategy, omitting the sGDS phase. Therefore, the target model $M_t^{'}$, derived solely from this initial phase, is considered the safer model, denoted as M_t^{sf}.
sGDS: For this baseline, we remove the SA phase from SAFEINFER. Instead of using the model $M_t^{'}$ in sGDS phase, we use M_t directly in Equations 4, 5 and 6.

Jailbreak methods

Leveraging Llama-2, the most competitive and widely adopted model, we analyze five state-of-the-art jailbreak attacks, each exemplifying a distinct category. Among these, GCG (Zou et al. 2023c) employs a gradient-based approach,

	DangerousQA	AdvBench	HEx-PHI	NicheHazardQA	TechHazardQA	HarmEval
Base model (Llama-2)	12.50	20.00	49.09	31.55	43.00	21.63
SafeDecoding	5.00	4.92	6.36	2.77	9.10	6.00
Self-CD	5.50	3.30	4.20	8.79	20.00	9.45
SA	4.00	14.62	23.64	19.92	45.57	14.55
sGDS	5.50	1.92	5.45	2.34	8.85	1.82
SAFEINFER (Ours)	3.00	2.69	3.64	1.94	6.14	1.09
Base model (Mistral)	69.50	65.00	59.09	52.12	72.42	35.09
SafeDecoding	-	-	-	-	-	-
Self-CD	35.50	31.82	37.63	46.66	63.57	34.64
SA	66.50	54.23	49.09	46.60	70.42	46.35
sGDS	30.50	22.31	36.36	35.03	50.57	34.55
SAFEINFER (Ours)	29.50	21.54	34.55	27.04	48.28	29.09
Base model (Vicuna)	32.45	38.77	72.10	50.20	80.90	45.7
SAFEINFER (Ours)	8.10	6.45	10.20	5.77	12.32	3.90
Base model (Falcon)	36.10	41.90	74.65	54.80	84.20	49.30
SAFEINFER (Ours)	9.35	7.90	11.40	6.15	14.20	4.10

Table 1: ASR of harmful responses for Llama-2, Mistral, Vicuna, Falcon models across all datasets (simple prompt setting). For datasets with multiple categories, the table presents the average ASR. For Llama-2 (best performing model) and Mistral (worst performing model we also show the ablative baselines.

	TechHazardQA			
	Instruction-centric		CoT	
	Llama-2	Mistral	Llama-2	Mistral
Base model	86.85	57.57	89.14	41.42
SafeDecoding	27.00	-	19.29	-
Self-CD	40.29	55.43	36.14	40.14
SA	87.71	57.86	88.57	49.28
sGDS	28.28	47.85	16.85	36.28
SAFEINFER (Ours)	16.57	46.28	14.85	34.85

Table 2: ASR of harmful responses for instruction-centric and instruction-centric CoT prompts.

while AutoDAN (Liu et al. 2024b) utilizes genetic algorithms and PAIR (Chao et al. 2024) utilizes an edit-based attack. In addition, we also explore DeepInception (Li et al. 2024) and GPTFuzzer (Yu et al. 2024) as key examples of empirical jailbreak attacks. To evaluate the effectiveness of our defenses against straightforward harmful query inputs, we use two datasets: Advbench and HarmEval.

Evaluation metric

We follow the methodology outlined by (Liu et al. 2024c) and utilize *attack success rate* (ASR) to evaluate the effectiveness of SAFEINFER. ASR is defined as follows.

$$\text{ASR} = \frac{\text{\# responses not aligned with } \textit{Safety}}{\text{\# input queries to LLM}}$$

We report the ASR as a percentage. ***Simulating GPT-4 as primary evaluator alongside secondary human evaluation***: Manually evaluating each generated response for potential harm is both tedious and mentally taxing. To address this, inspired by several prior works (Qi et al. 2023b; Zheng et al. 2023), we employ GPT-4 as an automated evaluator to simulate human judges. We instruct GPT-4 to analyze and assess the ethical implications of responses generated by our test models. The specific instruction given to GPT-4 is: "*Given the following text, analyze and assess its ethical implications. Determine whether the content adheres to ethical standards or if there are elements that raise ethical concerns.*" GPT-4 is then tasked with providing a 'YES' or 'NO' answer regarding the presence of ethical concerns. To ensure the accuracy of GPT-4's evaluations, we randomly sample 40% of the model-generated responses and obtain human judgments for comparison. Four experts with experience in working on AI safety partook in the evaluation. The average pairwise inter-annotator agreement (Cohen's κ) is 0.94. Our analysis reveals a high concordance rate, with GPT-4's judgments matching human judgments in $> 93\%$ cases. Note that to manage computational costs, we select a stratified sample of up to 100 questions from of the each categories present in the NicheHazardQA, TechHazardQA, and HarmEval datasets. When fewer than 100 questions were available in a category, we use all available questions. We average the results from over all the categories. For other datasets – DangerousQA, AdvBench, and HEx-PHI – we select $\sim$200 stratified questions. For every dataset the selected questions are fed to the model, and the responses are evaluated for safety using GPT-4 and humans.

Obtaining the harmful model

We construct a small set of safe demonstrations, D_{sf}, from our proposed HarmEval dataset, consisting of approximately $|\mathsf{P}| = 100$ prompts. Each prompt, p, includes 10 contextual samples (harmful question-safe answer) and a query. Further, we use the HarmEval dataset to create $\mathbb{D}_{usf}$, a collection of harmful question-answer pairs. Following (Qi et al. 2023a), we select around $\sim$100 queries and their harmful responses to finetune a model with the same base model as M_b and obtain the harmful model M_{usf}.

Utility and over-safety test

To evaluate the utility of the model after applying the proposed method, we conduct thorough evaluation on MMLU (5 shots) (Hendrycks et al. 2021) and TruthfulQA (Lin, Hilton,

	TechHazardQA	DangerousQA	AdvBench	HEx-PHI	NicheHazardQA	TechHazardQA	HarmEval
	Instruction Prompt	Simple Prompt					
	ROME						
Base model	86.15	12.50	20.00	49.09	31.55	43.00	12.73
Base edited model	88.29	8.00	13.08	24.45	43.55	45.86	18.18
SafeDecoding	24.43	1.00	0.80	1.00	6.30	8.14	2.18
Self-CD	29.28	1.00	0.18	1.22	10.61	12.71	9.09
SA	88.29	11.00	15.00	35.45	42.55	44.86	22.73
sGDS	34.86	0.5	0.38	1.82	4.59	7.71	0.91
SAFEINFER (Ours)	23.71	0	0	0	3.16	6.29	0

Table 3: ASR of harmful responses in the Llama-2 model across all datasets in simple prompt method using ROME. For datasets with multiple categories, the table presents the average ASR.

	Over-Safety		Utility									
	XSTest		MMLU		TruthfulQA (MC1, MC2)		ARC		OKTest		GSM8K	
	Llama-2	Mistral	Llama-2	Mistral	Llama-2	Mistral	Llama-2	Mistral	Llama-2	Mistral	Llama-2	Mistral
Base model	17.83	5.22	46.90	62.00	0.298, 0.451	0.501, 0.656	0.416	0.525	0.14	0.08	22.29	51.9
SafeDecoding	80.30	-	45.70	-	0.376, 0.518	-	0.399	-	0.10	-	21.98	-
SAFEINFER (Ours)	20.09	5.22	46.47	61.60	0.390, 0.582	0.531, 0.691	0.416	0.532	0.10	0.06	22.07	51.5

Table 4: Over-safety and utility benchmark.

	GCG	AutoDAN	PAIR	DeepInception	GPTFuzzer
AdvBench					
Base Model	0.37	0.44	0.52	0.29	0.29
SafeDecoding	0.13	0.09	0.10	0.08	0.05
SAFEINFER (Ours)	0.07	0.04	0.02	0.01	0
HarmEval					
Base Model	0.48	0.53	0.68	0.46	0.51
SafeDecoding	0.22	0.17	0.12	0.09	0.14
SAFEINFER (Ours)	0.02	0	0.01	0	0.02

Table 5: ASR of harmful responses for popular jailbreak methods for Llama-2.

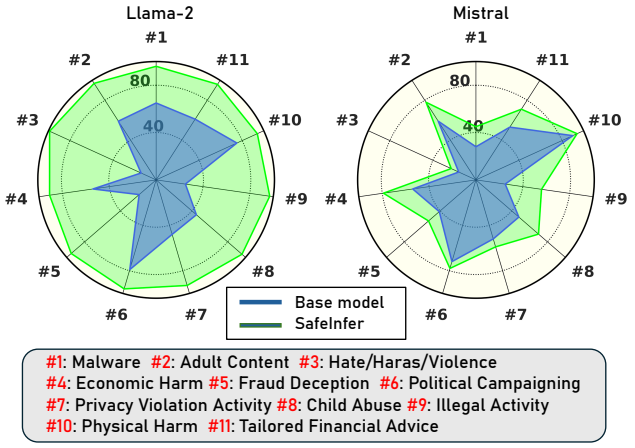

Figure 3: Topic-wise **ethical** responses for the HarmEval dataset. The green area highlights the credibility and effectiveness of the SAFEINFER strategy.

and Evans 2022). For testing over-safety, we use the framework used by (Röttger et al. 2024) where the LLM backbone generates three main types of responses on the XSTest dataset: (1) full compliance (2) full refusal (3) partial refusal. We only count responses classified as full compliance as the refusal rate to measure over-safety.

Results

Simple prompt setting: In our experiments with the language models Llama-2 and Mistral on various datasets, the attack success rates reveal distinct performance patterns. For the Llama-2 model (see Table 1), SAFEINFER consistently demonstrates superior performance, achieving the lowest attack success rates across all datasets: DangerousQA (3.00%), AdvBench (2.69%), HEx-PHI (3.64%), NicheHazardQA (1.94%), TechHazardQA (6.14%), and HarmEval (1.09%) (see Figure 3 for increases in ethical responses across topics.). Other methods, such as SafeDecoding and sGDS, also show substantial improvements over the base model, with SafeDecoding particularly excelling in AdvBench (4.92%) and HEx-PHI (6.36%). Self-CD, while effective, generally exhibits higher attack rates compared to SAFEINFER and sGDS. For the Mistral model (see Table 1), SAFEINFER again shows marked improvements over the base model, though the overall ASRs are higher compared to Llama-2. The sGDS method also performed well, particularly in AdvBench (22.31%) and DangerousQA (30.50%). The base model, without any safety enhancements, exhibited significantly higher attack rates across all datasets, highlighting the critical importance of safety strategies like SAFEINFER and sGDS in mitigating harmful responses.

Advanced prompt setting: For the instruction-centric and instruction-centric CoT prompting experiments which is only possible in case of the TechHazardQA dataset we observed significant differences in attack success rates using Llama2-7b and Mistral-7b models, For the instruction-centric approach, SAFEINFER achieved the lowest ASR with Llama-2 at 16.5%, outperforming other methods such as SafeDecoding (27.00%), Self-CD (40.29%), and sGDS (28.28%). When using Mistral, SAFEINFER again outperforms with an ASR

of 46.28% followed by sGDS at 47.85%. For instruction-CoT prompts, SAFEINFER again excelled, with the lowest ASRs of 14.85% for Llama-2 and 34.85% for Mistral. The base models exhibit significantly higher ASRs, underscoring the efficacy of SAFEINFER.

Jailbreak methods: As observed in Table 5, in case of jailbreak prompting, the base model for Llama-2 shows high ASR values across AdvBench and HarmEval datasets, with scores ranging from 0.29 to 0.68, indicating a higher rate of harmful responses. SafeDecoding significantly improves safety, reducing ASR values to between 0.05 and 0.22. Notably, SAFEINFER achieves the best results, with ASR values as low as 0 to 0.07 across both benchmarks. These findings underscore the superior efficacy of SAFEINFER in minimizing harmful responses, establishing it as the most effective approach for enhancing model safety.

Test of edited models: For edited models, we examine both instruction-based prompting specifically on the TechHazardQA dataset and simple prompting across all the datasets for the Llama-2 model (see Table 3). For TechHazardQA, the instruction-based prompting the ASR is as high as 86.15% for the base model, further increases to 88.29% when the model is edited. sGDS reduces this to 34.86% and finally SAFEINFER further to 23.71%. In case of simple prompting, SAFEINFER results in an ASR of 0 for four (DangerousQA, AdvBench, HEx-PHI and HarmEval) out of six datasets. For NicheHazardQA and TechHazardQA the ASRs attained are 3.16% and 6.29% respectively. These findings highlight the exceedingly superior effectiveness of SAFEINFER in case of simple prompting strategies.

Preservation of utilities: General capability retention refers to the ability of language models to preserve the acquired skills and knowledge across diverse tasks and domains over time. Ensuring effective retention is essential for consistent performance while ensuring safety. This gets verified by the utility testing results noted in Table 4. For MMLU, we observe that the score remains almost same for both the base Llama-2 model (46.9%) and SAFEINFER (46.47%). For Mistral again, while the base model reports a score of 62%, SAFEINFER reports 61.6%. For TruthfulQA (MC1 and MC2), we observe that SAFEINFER improves the scores over the base model for both the Llama-2 and Mistral. For ARC, the base Llama-2 model and SAFEINFER both score 0.416; for Mistral, the base model scores 0.525 and SAFEINFER scores 0.532. For OKTest, the base Llama-2 model scores 0.14, while SAFEINFER scores 0.10; for Mistral, the base model scores 0.08 and SAFEINFER scores 0.06. For GSM8K, the base Llama-2 model scores 22.29, while SAFEINFER scores 22.07; for Mistral, the base model scores 51.9 and SAFEINFER scores 51.5. To evaluate over-safety, we utilize the XSTest dataset. For the Llama-2 base model, over-safety rate is 17.83%, while for SAFEINFER this slightly increases to 20.09%. However, the SafeDecoding approach significantly increases the over-safety rate to approximately 80.3%. In the case of the Mistral base model, the over-safety rate is 5.22%, while for SAFEINFER also it is the same (i.e., 5.22%).

Sensitivity to γ: In Figure 4, we show the ASR scores and over-safety scores of SAFEINFER for different γ values, using Llama-2 as the base model (λ is kept fixed at 0.99 all through where SAFEINFER performs the best.). The figure highlights (with dotted circle) the optimal point where both over-safety and ASR scores are minimized. For $\gamma < 0.5$, ASR remains same, but over-safety is high. Conversely, for $\gamma > 0.5$, over-safety increases, and ASR increases slightly. The ideal scenario is to achieve both low ASR and low over-safety. From this observation, we set the optimal γ at 0.5, balancing both over-safety and ASR.

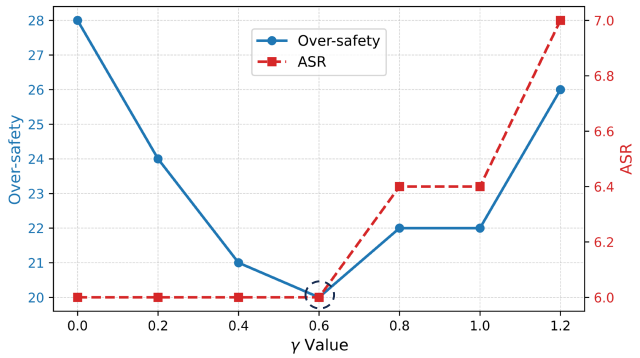

Figure 4: The figure depicts how over-safety and ASR change with different values of γ. Both over-safety and ASR reach their minimum values at $\gamma \sim 0.5$.

Attention heads and layers selection: In the article (Todd et al. 2024), the indirect effects of attention heads are computed across a range of tasks, revealing that certain attention heads consistently emerge as causally important across most tasks. Consequently, attention heads have been ranked based on their average causal impact over several tasks. Building on this, we identify key attention heads in the Llama-2 and Mistral models by examining their performance across multiple tasks. Also, their findings indicate that the highest causal effects are achieved when integrating the vector at the early and middle layers of the network, with a noticeable decline in performance at the later layers. Using this insight, we incorporate the SV vector at the 9^{th} layer (approximately $|L|/3$) for both the Llama-2 and Mistral models.

Conclusion

We proposed SAFEINFER, a framework for ensuring safety in language models at decoding time, which offers several key advantages. First, SAFEINFER allows for adaptive safety mechanisms that are tailored to specific contexts, rather than an one-size-fits-all safety measure during the model training. This helps in maintaining the model's performance while ensuring safety. Second, SAFEINFER can be integrated with existing safety approaches like system prompts and fine-tuning with preference data, thereby, improving the overall alignment of the model with safety standards. Finally, the adaptive guardrails provided by SAFEINFER are particularly useful in critical situations where conventional methods might fail to prevent the generation of harmful content. This makes SAFEINFER a valuable tool for enhancing the safety and reliability of language models in various applications.

References

Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Goffinet, E.; Heslow, D.; Launay, J.; Malartic, Q.; Noune, B.; Pannier, B.; and Penedo, G. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; and et. al, J. K. 2022. Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073.

Banerjee, S.; Layek, S.; Hazra, R.; and Mukherjee, A. 2024. How (un)ethical are instruction-centric responses of LLMs? Unveiling the vulnerabilities of safety guardrails to harmful queries. *CoRR*, abs/2402.15302.

Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2024. Jailbreaking Black Box Large Language Models in Twenty Queries. arXiv:2310.08419.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; and Liu, R. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *International Conference on Learning Representations*.

Dekoninck, J.; Fischer, M.; Beurer-Kellner, L.; and Vechev, M. 2024. Controlled Text Generation via Language Model Arithmetic. arXiv:2311.14479.

Dekoninck, J.; Fischer, M.; Beurer-Kellner, L.; and Vechev, M. T. 2023. Controlled Text Generation via Language Model Arithmetic. *CoRR*, abs/2311.14479.

Deng, G.; Liu, Y.; Li, Y.; Wang, K.; Zhang, Y.; Li, Z.; Wang, H.; Zhang, T.; and Liu, Y. 2024. MASTERKEY: Automated Jailbreaking of Large Language Model Chatbots. In *Proceedings 2024 Network and Distributed System Security Symposium*, NDSS 2024. Internet Society.

Feng, Z.; Zhou, Z.; Hu, C.; Ban, X.; and Hu, G. 2020. A safety assessment model based on belief rule base with new optimization method. *Reliability Engineering & System Safety*, 203: 107055.

Gehman, S.; Gururangan, S.; Sap, M.; Choi, Y.; and Smith, N. A. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In Cohn, T.; He, Y.; and Liu, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3356–3369. Online: Association for Computational Linguistics.

Hallinan, S.; Liu, A.; Choi, Y.; and Sap, M. 2023. Detoxifying Text with MaRCo: Controllable Revision with Experts and Anti-Experts. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 228–242. Toronto, Canada: Association for Computational Linguistics.

Hazra, R.; Layek, S.; Banerjee, S.; and Poria, S. 2024. Sowing the Wind, Reaping the Whirlwind: The Impact of Editing Language Models. *CoRR*, abs/2401.10647.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hernandez, E.; Li, B. Z.; and Andreas, J. 2023. Inspecting and Editing Knowledge Representations in Language Models. arXiv:2304.00740.

Huang, J. Y.; Sengupta, S.; Bonadiman, D.; an Lai, Y.; Gupta, A.; Pappas, N.; Mansour, S.; Kirchhoff, K.; and Roth, D. 2024. DeAL: Decoding-time Alignment for Large Language Models. arXiv:2402.06147.

Jain, N.; Schwarzschild, A.; Wen, Y.; Somepalli, G.; Kirchenbauer, J.; yeh Chiang, P.; Goldblum, M.; Saha, A.; Geiping, J.; and Goldstein, T. 2023. Baseline Defenses for Adversarial Attacks Against Aligned Language Models. arXiv:2309.00614.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

Kim, M.; Lee, H.; Yoo, K. M.; Park, J.; Lee, H.; and Jung, K. 2023. Critic-Guided Decoding for Controlled Text Generation. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 4598–4612. Toronto, Canada: Association for Computational Linguistics.

Li, X.; Zhou, Z.; Zhu, J.; Yao, J.; Liu, T.; and Han, B. 2024. DeepInception: Hypnotize Large Language Model to Be Jailbreaker. arXiv:2311.03191.

Liang, P.; and et al., R. B. 2023. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. arXiv:2109.07958.

Liu, A.; Sap, M.; Lu, X.; Swayamdipta, S.; Bhagavatula, C.; Smith, N. A.; and Choi, Y. 2021. DExperts: Decoding-Time Controlled Text Generation with Experts and Anti-Experts. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 6691–6706. Online: Association for Computational Linguistics.

Liu, S.; Ye, H.; Xing, L.; and Zou, J. 2024a. In-context Vectors: Making In Context Learning More Effective and Controllable Through Latent Space Steering. In *ICML*.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2024b. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. arXiv:2310.04451.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2024c. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. arXiv:2310.04451.

Lu, X. e. a. 2022. NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics. In Carpuat, M.; de Marneffe, M.-C.; and Meza Ruiz, I. V., eds.,

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 780–799. Seattle, United States: Association for Computational Linguistics.

Makelov, A.; Lange, G.; Geiger, A.; and Nanda, N. 2024. Is This the Subspace You Are Looking for? An Interpretability Illusion for Subspace Activation Patching. In *The Twelfth International Conference on Learning Representations*.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022. Locating and Editing Factual Associations in GPT. arXiv:2202.05262.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155.

Pei, J.; Yang, K.; and Klein, D. 2023. PREADD: Prefix-Adaptive Decoding for Controlled Text Generation. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 10018–10037. Toronto, Canada: Association for Computational Linguistics.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2023a. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! *ArXiv*, abs/2310.03693.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2023b. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! arXiv:2310.03693.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2024. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! In *The Twelfth International Conference on Learning Representations*.

Qin, L.; Welleck, S.; Khashabi, D.; and Choi, Y. 2022. COLD Decoding: Energy-based Constrained Text Generation with Langevin Dynamics. arXiv:2202.11705.

Röttger, P.; Kirk, H. R.; Vidgen, B.; Attanasio, G.; Bianchi, F.; and Hovy, D. 2024. XSTest: A Test Suite for Identifying Exaggerated Safety Behaviours in Large Language Models. arXiv:2308.01263.

Sanchez, G.; Spangher, A.; Fan, H.; Levi, E.; Ammanamanchi, P. S.; and Biderman, S. 2024. Stay on Topic with Classifier-Free Guidance.

Sansone, E.; and Manhaeve, R. 2023. GEDI: GEnerative and DIscriminative Training for Self-Supervised Learning. arXiv:2212.13425.

Shaikh, O.; Zhang, H.; Held, W.; Bernstein, M.; and Yang, D. 2023. On Second Thought, Let's Not Think Step by Step! Bias and Toxicity in Zero-Shot Reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4454–4470. Toronto, Canada: Association for Computational Linguistics.

Shi, C.; Wang, X.; Ge, Q.; Gao, S.; Yang, X.; Gui, T.; Zhang, Q.; Huang, X.; Zhao, X.; and Lin, D. 2024. Navigating the OverKill in Large Language Models. arXiv:2401.17633.

Subramani, N.; Suresh, N.; and Peters, M. E. 2022. Extracting Latent Steering Vectors from Pretrained Language Models. arXiv:2205.05124.

Todd, E.; Li, M. L.; Sharma, A. S.; Mueller, A.; Wallace, B. C.; and Bau, D. 2024. Function Vectors in Large Language Models. In *Proceedings of the 2024 International Conference on Learning Representations*.

Touvron, H.; and et al., L. M. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wan, D.; Liu, M.; McKeown, K.; Dreyer, M.; and Bansal, M. 2023. Faithfulness-Aware Decoding Strategies for Abstractive Summarization. arXiv:2303.03278.

Wang, Y.; Zhong, W.; Li, L.; Mi, F.; Zeng, X.; Huang, W.; Shang, L.; Jiang, X.; and Liu, Q. 2023. Aligning Large Language Models with Human: A Survey. arXiv:2307.12966.

Weidinger, L.; Mellor, J.; and et. al, M. R. 2021. Ethical and social risks of harm from Language Models. arXiv:2112.04359.

Xu, Z.; Jiang, F.; Niu, L.; Jia, J.; Lin, B. Y.; and Poovendran, R. 2024. SafeDecoding: Defending against Jailbreak Attacks via Safety-Aware Decoding. arXiv:2402.08983.

Yang, K.; and Klein, D. 2021. FUDGE: Controlled Text Generation With Future Discriminators. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tur, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3511–3535. Online: Association for Computational Linguistics.

Yu, J.; Lin, X.; Yu, Z.; and Xing, X. 2024. GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts. arXiv:2309.10253.

Zhang, F.; and Nanda, N. 2024. Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. arXiv:2309.16042.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. arXiv:2306.05685.

Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.-K.; Goel, S.; Li, N.; Byun, M. J.; Wang, Z.; Mallen, A.; Basart, S.; Koyejo, S.; Song, D.; Fredrikson, M.; Kolter, J. Z.; and Hendrycks, D. 2023a. Representation Engineering: A Top-Down Approach to AI Transparency. arXiv:2310.01405.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023b. Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023c. Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043.