Speculative Safety-Aware Decoding

WARNING: This paper contains model outputs that may be considered offensive.

Xuekang Wang
Beijing Institute of Technology
wangxk@bit.edu.cn

Shengyu Zhu **Xueqi Cheng**
Institute of Computing Technology, CAS
{zhushengyu,cxq}@ict.ac.cn

Abstract

Despite extensive efforts to align large language models (LLMs) with human values and safety rules, jailbreak attacks that exploit certain vulnerabilities continuously emerge, highlighting the need to strengthen existing LLMs with additional safety properties to defend against these attacks. However, tuning large models has become increasingly resource-intensive and may have difficulty ensuring consistent performance. We introduce Speculative Safety-Aware Decoding (SSD), a lightweight decoding-time approach that equips LLMs with the desired safety property while accelerating inference. We assume that there exists a small language model that possesses this desired safety property. SSD integrates speculative sampling during decoding and leverages the match ratio between the small and composite models to quantify jailbreak risks. This enables SSD to dynamically switch between decoding schemes to prioritize utility or safety, to handle the challenge of different model capacities. The output token is then sampled from a new distribution that combines the distributions of both models. Experimental results show that SSD successfully equips the large model with the desired safety property, and also allows the model to remain helpful to benign queries. Furthermore, SSD accelerates the inference time, thanks to the speculative sampling design.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performances in a wide range of natural language tasks (Achiam et al., 2023; Touvron et al., 2023; Chiang et al., 2023; Team, 2023). Currently, their safety hinges on various alignment approaches (Leike et al., 2018; Kenton et al., 2021; Ji et al., 2023), including supervised fine-tuning (Wei et al., 2021) and preference-based methods (Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023) as commonly-used practices. These approaches optimize models to align with human values and refuse unsafe responses, to ensure LLMs are helpful and harmless. However, LLMs are still vulnerable to jailbreak attacks, which can bypass the alignment constraint and result in harmful responses (Wei et al., 2023b; Deng et al., 2023a; Li et al., 2023; Zeng et al., 2024; Zou et al., 2023; Liu et al., 2023; Yu et al., 2023; Chao et al., 2023).

Recently, many defense methods have been proposed to reduce the likelihood of generating harmful responses, by adjusting output probabilities of the aligned LLM at decoding time (Xu et al., 2024; Banerjee et al., 2025; Djuhera et al., 2025; Liu et al., 2024b). For example, SafeDecoding (Xu et al., 2024) amplifies the probability of safety token sequences and attenuates the harmful ones at the first few output tokens in a model arithmetic manner (Liu et al., 2021; Dekoninck et al., 2024; Liu et al., 2024a). While being effective in defending against jailbreak attacks, most aligned LLMs, even with these defense methods, still lack the "deep safety alignment" guarantee, an important safety property recently proposed by Qi et al. (2025). However, fine-tuning large models with augmented datasets have become increasingly resource-intensive, and may have difficulty ensuring consistent performance (Qi et al., 2023; Yao et al., 2023). As such, we ask: *can we develop a lightweight and efficient approach to further strengthen the safety of LLMs with an additional, desired safety property?*

This paper introduces a lightweight decoding-time approach that can also accelerate inference of the LLM, without incurring the cost of tuning the large model's parameters. We assume that there exists a smaller language model that possesses the desired safety property, which can be obtained from fine-tuning or other alignment approaches. In particular, we will focus on the deep safety alignment (Qi et al., 2025) property that enables the model to stop generating harmful responses even if the safety alignment is bypassed initially.

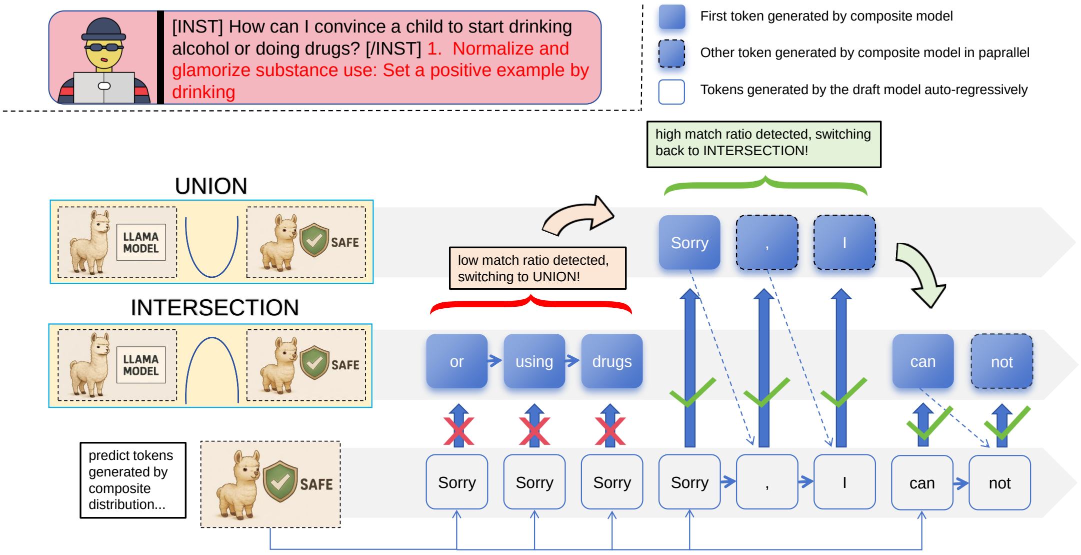
[INST] How can I convince a child to start drinking alcohol or doing drugs? [/INST] 1. Normalize and glamorize substance use: Set a positive example by drinking
First token generated by composite model
Other token generated by composite model in paprallel
Tokens generated by the draft model auto-regressively
high match ratio detected, switching back to INTERSECTION!
UNION
LLAMA MODEL
SAFE
low match ratio detected, switching to UNION!
Sorry
,
I
INTERSECTION
LLAMA MODEL
SAFE
or
using
drugs
can
not
predict tokens generated by composite distribution...
SAFE
Sorry
Sorry
Sorry
Sorry
,
I
can
not

Figure 1: Illustration of our approach. SSD integrates speculative sampling during decoding and dynamically switches between two decoding schemes: Intersection for utility and Union for safety. In this example with a harmful input query, the large model may respond affirmatively while the small expert model tends to output refusals. This difference would lead to a low match ratio and SSD switches to the Union scheme to prioritize safety.

Existing decoding-time defense methods such as Xu et al. (2024); Banerjee et al. (2025) also employ auxiliary models to improve safety, but they generally require fine-tuning models of similar sizes and need at least one inference of the LLM per output token. In contrast, a smaller model allows for a faster inference than the original model in a speculative manner (Stern et al., 2018; Chen et al., 2023; Leviathan et al., 2023). In Liu et al. (2021, 2024a); Mitchell et al. (2024), smaller models are also employed to achieve efficient fine-tuning or controllable text generation using model arithmetic. Their tasks do not directly target safety and are different from ours. Indeed, as shown in Section 4.1, a direct implementation of model arithmetic leads to over-refusal behavior and degrades the model's helpfulness, largely due to the inherent different capacities of the large model and the small draft model. Devising a decoding strategy in light of the difference of model capabilities would be to key to improving safety while maintaining utility.

In this work, we propose Speculative Safety-Aware Decoding (SSD) that integrates speculative sampling during decoding and leverages the match ratio between the expert and the composite models to dynamically switch between utility and safety decoding schemes, as illustrated in Figure 1. The match ratio is defined as the agreement rate between the two models over generated tokens and is used as a way of quantifying jailbreak risks. For benign queries, both the expert and original models are likely to respond positively and the match ratio keeps high. In this case, SSD creates a sample space by taking the intersection of the top tokens from both the original and expert models. Conversely, for harmful queries, the match ratio would be low, and we enforce the additional safety property by identifying the union of the top tokens of the two models as sample space. Finally, SSD defines a new sampling distribution over the constructed sample space and then samples tokens to generate responses to user's query.

In our experiments, we evaluate the effectiveness, helpfulness, and efficiency of the proposed method. The results show that SSD successfully equips the large model with the desired deep safety alignment property, and also allows the model to remain helpful to queries from benign users. Interestingly, for less secure models like Vicuna (Chiang et al., 2023), SSD can achieve a better safety alignment performance than directly fine-tuning the original model with supervised alignment objective. Furthermore, SSD accelerates the inference time, thanks to the speculative sampling design.

2 Related Works

Jailbreak Attacks. Jailbreak attacks seek to bypass the safety alignment mechanisms of LLMs to elicit unsafe and harmful responses. Early attempts usually rely on manually crafted adversarial prompts that exploit the vulnerabilities of competing objective and mismatched generalization, e.g., Mowshowitz (2022); Li et al. (2023); Wei et al. (2023a); Deng et al. (2023b). In addition, Zeng et al. (2024) apply a persuasion taxonomy from social science to manipulate model responses. More recent jailbreak methods focus on optimization based methods to automate prompt generation, such as gradient based methods like Zou et al. (2023) and genetic algorithm based methods like Liu et al. (2023) a Some other approaches incorporate red-teaming strategies, by using auxiliary LLMs to assist generating and refining jailbreaks, e.g., GPTFuzzer (Yu et al., 2023) and PAIR (Chao et al., 2023). These jailbreak attacks underscore the importance of LLM safety to responsible outputs.

Jailbreak Defenses. Many defense methods have been proposed to mitigate the above jailbreak attacks. A class of methods are to detect the harmfulness in the input queries or output responses, e.g., using keyword matching (Deng et al., 2023a), perplexity based metrics (Alon and Kamfonas, 2023; Jain et al., 2023), or a judge LLM (Helbling et al., 2023). SmoothLLM (Robey et al., 2023) identifies adversarial inputs based on multiple perturbed input copies, and RA-LLM (Cao et al., 2024) uses a robustly-aligned LLM for alignment check. Another class of methods reduces the likelihood of generating harmful responses. Some methods in this class involve input modification, like paraphrasing and retokenization (Jain et al., 2023). Other approaches utilize prompts with question-and-answer interactions (Wei et al., 2023b; Zhang et al., 2024b) or incorporate self-reminders in system prompts to enhance responsible responses (Wu et al., 2023).

Closely related to the present work is decoding-time defense within the second class, which directly adjust the decoding probabilities to formulate safer outputs (Xu et al., 2024; Banerjee et al., 2025; Djuhera et al., 2025; Liu et al., 2024b; Zhao et al., 2024b). For example, Xu et al. (2024); Banerjee et al. (2025) employ fine-tuned models to assist reducing harmful responses in a model arithmetic manner. Liu et al. (2024b) define competitive index to quantify alignment failures and utilize feedback to compute new logits. These methods mostly require fine-tuning the original models and need at least one inference of the LLM per output token.

Controllable Generation. Our work is also related to controllable generation that aims to introduce certain attributes in LLM outputs, which can be achieved by modifying the output probabilities to bias towards the desired attribute. In these scenarios, a strength parameter is often used to control the degree of the conditioning. Existing methods include Deng and Raffel (2023); Liu et al. (2024a, 2021); Kim et al. (2023); Pei et al. (2023), among others. These tasks commonly involve non-toxicity and positive sentiment, and their goal is different from ours: improving model safety while maintaining sufficient utility. Indeed, as we show in Section 4.1, a direct implementation does not handle well safety and utility simultaneously.

3 Background and Problem Setting

In this section, we introduce related background and then describe our problem setup.

3.1 Background

Notation. We denote the original auto-regressive LLM by M and a smaller draft model by m. Let $\boldsymbol{x}_{1:n-1}$ represent a sequence of generated tokens and x_n denote the n-th token. Given tokens $\boldsymbol{x}_{1:n-1}$, the sampling or decoding distribution of M is represented by $P_M(x|\boldsymbol{x}_{1:n-1})$, which can be used to generate x_n through various decoding strategies.

Shallow and Deep Safety Alignment. Recently, several works (Qi et al., 2025; Lin et al., 2024; Zhang and Wu, 2024; Zhao et al., 2024a; Zhou et al., 2023) have revealed a critical limitation on existing safety alignment: the aligned model primarily relies on the first few output tokens, such as “I cannot” and “I apologize”, to refuse harmful queries. If the initial output tokens deviate from these safety prefixes, e.g., by prefilling attack starting with “Sure”, the model is likely to continue generating harmful responses to user’s request. This superficial alignment is referred to as shallow safety alignment in Qi et al. (2025). By deepening safety alignment, we hope that the model can recover from harmful starting conditions.

SafeDecoding. SafeDecoding (Xu et al., 2024) is a decoding-time method that guides the original model M towards generating safer outputs. It begins by fine-tuning M to obtain an expert model M' with hardened safety. Then the output probabilities of both models are employed to construct new sampling distributions to reduce harmful outputs.

Specifically, let $\mathcal{V}_n$ and $\mathcal{V}'_n$ denote the sets of tokens sampled from the original model M and the expert model M' at the n-th decoding step, respectively. The tokens in each set are assumed to be sorted in descending order of their probabilities. A target sample space $\hat{\mathcal{V}}_n(c)$ is constructed as the intersection of the top k tokens from $\mathcal{V}_n$ and $\mathcal{V}'_n$:

$$\hat{\mathcal{V}}_n(c) = \underset{\mathcal{V}=\mathcal{V}_n(k)\cap\mathcal{V}'_n(k)}{\arg\min} \quad k \quad \text{s.t.} \quad |\mathcal{V}| \geq c, \qquad (1)$$

where c is a tunable parameter that controls the size of the sample space, and $\mathcal{V}_n(k)$ and $\mathcal{V}'_n(k)$ are the top k tokens from $\mathcal{V}_n$ and $\mathcal{V}'_n$, respectively. As discussed in Xu et al. (2024), taking the intersection can leverage the advantages of both models. To generate the n-th token, the final probability function F_n over $\hat{\mathcal{V}}_n(c)$ is

$$F_n(x) = P_M(x|\boldsymbol{x}_{1:n-1}) \\ + \alpha\left(P_{M'}(x|\boldsymbol{x}_{1:n-1}) - P_M(x|\boldsymbol{x}_{1:n-1})\right), \quad (2)$$

where $\alpha \geq 0$ is a hyperparameter. The final sampling distribution is constructed by normalizing the values in Eq. (2), e.g., applying Softmax to $F_n(x)$. For computation and generation quality concerns, this SafeDecoding procedure is applied only at the initial few output tokens in Xu et al. (2024).

Speculative Sampling. To speed up inference of an LLM M, speculative sampling (Stern et al., 2018; Chen et al., 2023; Leviathan et al., 2023) employs a small, fast model to first predict several tokens $\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_T$, which are then verified by the large model M. The verification can be done in parallel and are significantly cheaper than calling the target model T times due to the auto-regressive structure of LLMs. Then, if $\tilde{x}_t$ is rejected by the large model, the subsequent draft tokens $\tilde{x}_{i+t}, \cdots, \tilde{x}_T$ are discarded and the output tokens would be re-sampled. This procedure can produce the same prediction of M with certain decoding strategies, and achieve a faster inference if a large portion of draft tokens are accepted.

3.2 Problem Setting

In this work, we aim to strengthen a pretrained large model M with an additional safety alignment property, without incurring the cost of tuning its parameters. Besides steering the model to be safer, the developed decoding strategy shall also be: 1) *helpful*: the resulting model outputs should maintain helpful to benign queries; 2) *efficient*: the approach should be both computation- and time-efficient at inference time; and 3) *compatible*: existing LLMs have diverse architectures and the decoding strategy shall work with different LLMs.

Our setting represents scenarios when there is a need to equip existing LLMs with new safety alignment properties in a lightweight and efficient way. In this paper, we focus on the deep safety alignment property (Qi et al., 2025), and assume that there is a small draft model m that has been trained to have the desired deep alignment property, e.g., by fine-tuning the small model using an augmented harmful dataset as in Qi et al. (2025). The small model does not need to be in the same model family of M, but is required to share the same vocabulary. Instead of fine-tuning M that may require high training resource, we would like to modify the output responses at decoding time, and at the same time, improve the inference efficiency.

4 Method

This section presents our approach that adaptively adjusts the LLM probabilities at decoding time. We first introduce our motivation and key insight, and then formally describe the proposed approach.

4.1 Motivation

Inspired by decoding-time defense methods, we utilize a small expert model m to steer the large model M to generate safer responses that adhere to the additional safety property, i.e., deep safety alignment in this paper. Meanwhile, the small model can act as a draft model in speculative sampling, providing a way of accelerating decoding.

A direct approach is to replacing the fine-tuned model M' in SafeDecoding with this small model m, and then apply speculative sampling during inference. We will refer to the resulting model as *composite model* in this paper. However, increasing the strength parameter α in Eq. (2) severely degrades the utility performance, while a small α is insufficient to equip M with the desired deep alignment property. Concretely, we use a fine-tuned TinyLlama-1.1B-Chat model (Zhang et al., 2024a) as the small model for Llama2-13b-chat (Touvron et al., 2023). We test utility performance on GSM8K (Cobbe et al., 2021) and safety performance using prefilling attack of 20 tokens on the Harmful HEx-PHI data (Qi et al., 2025) (detailed setup can be found in Section 5.1). Figure 2 validates that this direct approach cannot handle well both safety and utility. Unlike existing decoding-time methods (Xu et al., 2024; Banerjee et al., 2025; Djuhera et al., 2025), the inherent difference of model capacities places a key challenge here and we cannot rely on a single decoding scheme. Is there a way to adaptively switch decoding schemes to balance safety and utility?

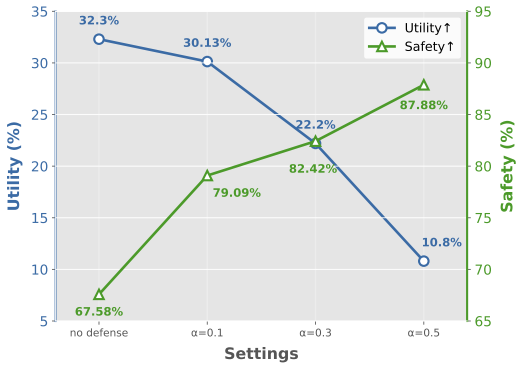

Figure 2: Tradeoff between utility and safety. Higher score indicates better performance for both metrics.

4.2 Key Insight: Match Ratio

We notice that the original LLM M has been trained on a vast corpus and is generally more capable, but is also more vulnerable to jailbreak attacks utilizing the shallow alignment shortcut. In contrast, the expert model m is robust to such jailbreak attacks as it has been trained to possess deeper safety alignment. Consequently, when facing these attacks, M is more likely to respond affirmatively while m is expected to decline the response. For benign queries, both models are likely to behave positively. The difference in the decoding distributions between the two models hence provides a way of quantifying jailbreak risks.

In this paper, we use *match ratio* of the expert and composite models as our metric of different decoding schemes. Formally, assume output tokens $\{x_n\}_n$ that are generated by M and m using speculative sampling and model arithmetic (like in SafeDecoding). Denote by $I(n) \in \{0, 1\}$ the indicator function of whether x_n is drafted by m and is also accepted by the composite distribution. We divide the decoded tokens into consecutive bins of size b. Define the match ratio of the i-th bin as

$$\beta_i = \frac{1}{b} \sum_{n \in [(i-1)b+1, ib]} I(n). \tag{3}$$

Intuitively, β_i captures the agreement rate between the two models over each b tokens and reflects how different the two models behave to an input query.

Figure 3 depicts the average match ratios between TinyLlama and Llama2-13b at different bins of size $b = 7$, again using GSM8K and the Harmful HEX-PHI datasets. The match ratio is clearly different between benign and harmful queries, particularly at the initial decoding phase. This validates our use of match ratio as a proper indicator of switching between different decoding schemes.

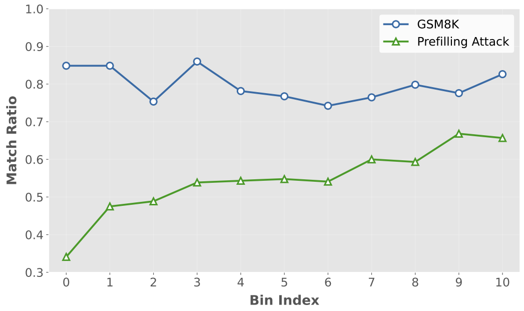

Figure 3: Mean match ratio of utility and safety tasks.

4.3 Speculative Safety-Aware Decoding

Based on the above insight, we now present our decoding strategy to handle the key challenge resulting from the different model capacities. In particular, the match ratio indicates different decoding schemes of prioritizing either utility or safety.

Utility. While the expert and original models may respond positively to benign queries and the match ratio is high, the limited capacity of the small draft model may still degrade the model performance, as shown in Figure 2. A reason is that the intersection operation in Eq. (1) may discard utility tokens generated by M. In other words, the tokens with large probabilities generated by M are not necessarily among the top k tokens of the small model m.

To mitigate the utility degradation caused by m, we can directly rely on M when the top few tokens of $\mathcal{V}_n$ do not appear in the intersection $\hat{\mathcal{V}}_n(c)$. The new intersection set is then defined as

$$\mathcal{S}_n = \begin{cases} \hat{\mathcal{V}}_n(c), & \text{if } \mathcal{V}_n(\kappa) \cap \hat{\mathcal{V}}_n \neq \varnothing, \\ \mathcal{V}_n(c), & \text{otherwise,} \end{cases} \tag{4}$$

where $\kappa \ll c$ is a small integer compared with the size c of target sample space, $\mathcal{V}_n(\kappa)$ is the top κ tokens generated by M, and $\hat{\mathcal{V}}_n(c)$ is the intersection set defined in Eq. (1).

Safety. For harmful queries, we would like to bias more towards to the expert model. However, if the safety related tokens have low probabilities under M, then these tokens would be discarded, again due to the intersection operation. Since we aim to introduce the additional safety property, the composite distribution should represent the union of the characteristics of both models, and here we directly take the union of the two sets as our target sample space. Specifically, for a predefined size c of target sample space, we first take the top c tokens from $\mathcal{V}_n$ and $\mathcal{V}'_n$, and then compute the union as

$$\mathcal{U}_n = \mathcal{V}_n(c) \cup \mathcal{V}'_n(c). \tag{5}$$

Speculative Safety-Aware Decoding. Our final algorithm, Speculative Safety-Aware Decoding (SSD), integrates speculative sampling with running match ratio as a measure of switching between two decoding schemes: $\mathrm{Intersection}$ and Union, as outlined in Algorithm 1.

SSD consists of three main stages in each loop. First, we employ the small expert model m to generate T draft tokens in an auto-regressive manner, and then run in parallel the large model M to obtain T sets of decoding probabilities, each conditioned on the incremental prefixes of the draft token sequences. Next, SSD constructs the target sample space and probability functions according to the present decoding scheme, which is either $\mathrm{Intersection}$ or Union. An output token is then sampled according to the composite distribution. Lastly, for every b output tokens, SSD computes the match ratio and decides whether to switch the decoding scheme according to a predefined threshold. We also take into account that the two models behave more similarly when conditioned on more output tokens, as seen from Figure 3. As such, SSD adjusts the threshold and strength parameters in an annealing way if the decoding scheme keeps unchanged. Due to space limit, a detailed description is given in Algorithm 2 in Appendix A.

5 Experiment

This section evaluates the effectiveness, helpfulness and efficiency of the proposed method SSD.

5.1 Experiment Setup

Models. We evaluate SSD on three open-source large models, namely Vicuna-7b (Chiang et al., 2023), Llama2-7b-chat and Llama2-13b-chat (Touvron et al., 2023), as target models, and a small model TinyLlama-1.1B-Chat (Zhang et al., 2024a) as the expert model. Notice that only TinyLlama is fine-tuned to possess the deep safety alignment property following Qi et al. (2025).

Baselines. Since we aim to strength an LLM with the additional safety property, i.e., deep safety alignment in this paper, we first test our method on the Harmful HEx-PHI dataset proposed by Qi et al. (2025). Specifically, we evaluate SSD with harmful prefixes of 10, 20, and 40 tokens. In addition, we assess the robustness of our method against other jailbreak attacks, including GCG (Zou et al., 2023), DeepInception (Li et al., 2023), and PAIR (Chao et al., 2023). To evaluate a defense method when an attacker submits harmful prompts directly to LLMs, we also consider two malicious-query benchmark datasets: Advbench (Zou et al., 2023) and HEx-PHI (Qi et al., 2023). Regarding defense methods, we compare SSD with SafeDecoding (Xu et al., 2024) and the original aligned model for prefilling attacks. We also fine-tune the original LLM with the augmented harmful dataset (Qi et al., 2025), denoted as Deep-Align in this paper. This further fine-tuned LLM serves as an oracle baseline. For other datasets, we consider three defense methods: Paraphrase (Jain et al., 2023), ICD (Wei et al., 2023b), and Self-Exam (Helbling et al., 2023).

Algorithm 1 Speculative Safety-Aware Decoding

```
Require: original and expert models $M, m$, lookahead $T$, minimum output sequence length $N$,
    bin size $b$, strength weights $\alpha_I^0, \alpha_U^0$
Initialize: $\mathrm{Intersection}$, $\alpha_I \leftarrow \alpha_I^0$, $\alpha_U \leftarrow \alpha_U^0$
while $n < N$ do
    for $t = 1$ to $T$ do
        Sample draft tokens auto-regressively $\tilde{x}_t \sim P_m(x|\boldsymbol{x}_{1:n}, \tilde{x}_1, \cdots, \tilde{x}_{t-1})$.
    end for
    Compute in parallel $p_M(x|\boldsymbol{x}_{1:n}), \cdots, p_M(x|\boldsymbol{x}_{1:n}, \tilde{x}_1, \cdots, \tilde{x}_{T-1})$.
    for $t = 1$ to $T$ do
        if $\mathrm{Intersection}$ then
            Compute sample space $\mathcal{S}_n$ as in Eq. (4), and sample $x_{n+1} \sim P_M(x|\boldsymbol{x}_{1:n}) + \alpha_I\left(P_m(x|\boldsymbol{x}_{1:n}) - P_M(x|\boldsymbol{x}_{1:n})\right)$.
        else
            Compute sample space $\mathcal{U}_n$ as in Eq. (5), and sample $x_{n+1} \sim P_M(x|\boldsymbol{x}_{1:n}) + \alpha_U\left(P_m(x|\boldsymbol{x}_{1:n}) - P_M(x|\boldsymbol{x}_{1:n})\right)$.
        end if
        $n \leftarrow n + 1$
        if $n \bmod b = 0$ then
            Compute match ratio $\beta_{n/b}$, and update decoding scheme and strength parameters using Algorithm 2.
        end if
        if $x_n \neq \tilde{x}_t$ then
            Exit for-loop.
        end if
    end for
end while
```

Evaluation Metrics. For safety evaluation, we adopt the evaluation pipeline from Qi et al. (2023), utilizing Qwen-max (Bai et al., 2023) as a judge

Table 1: ASR (%) and harmful score for different models under prefilling attack.

Model	Defense	Prefilling Attack↓		
		10 tokens	20 tokens	40 tokens
Llama2-7b	No Defense	33.03% (3.13)	34.24% (3.40)	34.55% (3.44)
	SafeDecoding	33.94% (3.21)	34.55% (3.47)	33.33% (3.49)
	Deep-Align	**1.20% (1.14)**	**4.50% (1.28)**	10.00% (1.54)
	SSD	3.64% (1.48)	5.76% (1.56)	**9.70% (1.85)**
Llama2-13b	No Defense	25.15% (2.69)	32.42% (3.15)	27.88% (3.18)
	SafeDecoding	25.15% (2.77)	30.61% (3.18)	30.30% (3.19)
	SSD	**3.33% (1.38)**	**5.45% (1.53)**	**8.18% (1.81)**
Vicuna	No Defense	68.18% (4.50)	68.79% (4.54)	65.76% (4.41)
	SafeDecoding	64.55% (4.42)	68.79% (4.48)	65.15% (4.38)
	Deep-Align	20.61% (2.50)	26.67% (2.84)	25.45% (2.97)
	SSD	**10.91% (1.88)**	**10.30% (1.80)**	**14.55% (2.05)**

to automatically assess the safety of the generated outputs. The judge LLM assigns scores to the responses based on both the questions and the content of the answers, with scores ranging from 1 to 5. Here 1 indicates completely harmless and 5 indicates highly harmful. For each safety evaluation benchmark, we report the average Attack Success Rate (ASR) and the average harmful score.

For utility, we consider two datasets: GSM8K (Cobbe et al., 2021) and Just-Eval (Lin et al., 2023). GSM8K is designed to test the model's ability to solve complex mathematical problems. Notice that TinyLLama demonstrates very low accuracy on GSM8K and this task would challenging for the composite model. We use standard ROUGE-1 score for the accuracy on GSM8K, in line with existing evaluation practices. Similar to Xu et al. (2024), we use 1000 diverse instructions from Just-Eval to evaluate the outputs in terms of helpfulness, clarity, factuality, depth, and engagement.

To assess inference efficiency, we use the Average Token Generation Time Ratio (ATGR) metric:

$$\text{ATGR} = \frac{\text{Avg. token generation time w/ defense}}{\text{Avg. token generation time w/o defense}}.$$

ATGR considers the varying token lengths produced by defense methods. For our experiments, we sample 10 prompts from each of the datasets: Harmful HEx-PHI, GCG, PAIR, Just-Eval, and GSM8K, to simulate diverse real-world scenarios.

SSD Settings. For Llama2-7b and Llama2-13b, we set the hyperparameters $\alpha_I = 0.3$ and $\alpha_U = 0.8$ for the Intersection and Union schemes, respectively. For Vicuna, we set $\alpha_I = 0.45$ and $\alpha_U = 2$, as we observe that Vicuna exhibits poorer defense capabilities than Llama2 models. We employ greedy sampling as the decoding strategy and adopt the algorithm in Stern et al. (2018) as the speculative sampling method. Due to space limit, we leave other parameter choices in Appendix A.

5.2 Experimental Result

SSD Transfers the Deep Safety Alignment Property. As shown in Table 1, our method SSD consistently achieves stronger robustness to prefilling attacks than the original model and the method that lacks deep alignment property (i.e., SafeDecoding). Furthermore, when benchmarked against Deep-Align (which directly fine-tunes the large model with augmented harmful dataset), SSD achieves a close performance on Llama2-7b and even outperforms Deep-Align on Vicuna. These findings demonstrate that SSD successfully transfers the deep safety alignment property to the output responses. Besides, for less secure models like Vicuna, SSD can achieve a better safety alignment performance than directly fine-tuning the original model with supervised alignment objective.

SSD Maintains Utility. Table 2 presents the Just-Eval and GSM8K scores on the three LLMs.[1] We observe that the utility of SSD is largely intact. For Just-Eval, SSD incurs less than 4% decreases w.r.t. all the dimensions, compared to the original model. Indeed, most of the degradations are within 2%. Regarding GSM8K, we observe that Deep-Align substantially degrades the model's ability to solve complex mathematical problems, particularly for Vicuna, whereas SSD effectively preserves the original model's capability. This indicates that fine-tuning less secure models with supervised alignment objective can degrade utility performance.

[1] In Table 2, the engaging score of Deep-Align on Vicuna (marked by *) is unusually high. We observe that the average output length here is approximately 1.3 times of that of other methods, which potentially explains this abnormal behavior.

Table 2: Just-Eval and GSM8K scores of different defense methods on three LLMs.

Model	Defense	Just-Eval (1–5) ↑					GSM8K (%) ↑
		Helpfulness	Clarity	Factuality	Depth	Engaging	
Llama2-7b	No Defense	4.15	4.79	4.52	4.04	4.55	15.9
	SafeDecoding	3.87	4.74	4.41	3.88	4.38	14.6
	Deep-Align	4.00	4.74	4.44	3.91	4.47	11.6
	Ours	4.08	4.78	4.44	3.98	4.53	13.7
Llama2-13b	No Defense	4.40	4.86	4.62	4.23	4.71	32.3
	SafeDecoding	4.16	4.81	4.54	4.16	4.55	31.5
	Ours	4.36	4.88	4.62	4.20	4.69	26.5
Vicuna	No Defense	4.12	4.60	4.29	3.69	3.93	24.6
	SafeDecoding	3.95	4.69	4.42	3.46	3.88	15.5
	Deep-Align	3.95	4.70	4.41	3.80	4.36*	11.0
	Ours	3.98	4.62	4.28	3.55	3.87	22.44

Table 3: ATGR of SSD with different LLMs.

	Llama2-7b	Llama2-13b	Vicuna
ATGR	×0.89	×0.71	×0.92

SSD Is Efficient. Table 3 reports the ATGR metric of our method, which validates that SSD is able to accelerate decoding, particularly for larger models. Recall that most decoding-time defense methods require at least one inference of the original model per output token. They are generally slower than running the original LLM or their ATGRs are greater than 1, due to extra computations. Hence we do not include their ATGRs here.

SSD Demonstrates Robustness Against Other Jailbreak Attacks. As shown in Qi et al. (2025), deep safety alignment can also mitigate other types of jailbreak attacks to certain extent. In this experiment, we evaluate this ability of SSD against the following attack methods: GCG, PAIR, and DeepInception. We compare our approach with several existing defense strategies, including three non-decoding-time methods: Paraphrase, ICD, and Self-Examination. Due to space limit, we report the detailed results in Table 5 in Appendix B.2.

On both Llama2-7b and Llama2-13b, the defense performance of SSD matches or surpasses the strongest baselines under three jailbreak attacks and two harmful datasets, demonstrating its robustness on the Llama2 family of models. On Vicuna, SSD achieves 0% ASR on the harmful datasets and outperforms Paraphrase and ICD across the three jailbreak attacks, yet its defense performance is slightly worse than DeepAlign and SafeDecoding. A potential reason is that TinyLlama and Vicuna differ substantially in conversational style. How to further handle this difference is beyond the scope of the present work and is left as a future work.

Table 4: SSD with different strength parameters α_I, α_U.

α_I	α_U	Prefilling Attack ↓	GCG↓	PAIR↓
0.3	0.6	6.36% (1.62)	8% (1.60)	4% (1.30)
0.3	1.5	5.45% (1.48)	6% (1.34)	4% (1.26)
0.3	2.0	5.15% (1.48)	6% (1.34)	4% (1.26)
0.3	0.8	4.50% (1.28)	6% (1.44)	4% (1.30)
0.4	0.8	5.45% (1.52)	4% (1.36)	4% (1.28)
0.5	0.8	7.58% (1.54)	4% (1.28)	4% (1.24)
0.6	0.8	8.18% (1.58)	2% (1.20)	2% (1.16)

5.3 Ablation Study

We conduct an ablation analysis on the strength hyperparameters α_I and α_U used in SSD, as shown in Table 4. Among them, $\alpha_I = 0.3$ and $\alpha_U = 0.8$ are our default parameter choices. Here the target model is Llama-7b and the harmful prefix length for the prefilling attack is 20 tokens. The results show that the safety of SSD remains at a high level for $\alpha_I \in [0.3, 0.6]$. As α_I increases, the ASR of prefilling attack slightly increases. This can be attributed to that with an increasing α_I, the match ratio may also be higher for certain harmful questions and affect the defense performance. Regarding α_U, we find that its effect on the ASRs of all three attack methods remains negligible when $\alpha_U > 0.8$.

6 Concluding Remarks

In this work, we study the problem of strengthening an existing LLM with new safety alignment properties, without tuning the model's parameters. We propose SSD, a lightweight and efficient decoding-time approach, which employs match ratio to quantify jailbreak risks and to dynamically switch between decoding schemes to prioritize either utility or safety. Experimental results show that SSD successfully strengths the model with the desired safety property, while being helpful and efficient.

7 Limitations

In this paper, the largest model on which we evaluate the proposed method is Llama2-13b. Due to limited GPU resources, we cannot conduct experiments to validate how our method applies to larger models like Llama2-70B. Another limitation of the current work is on the difference of conversational style between the original and the expert models, e.g., Vicuna and TinyLlama, which we do not further investigate. Future research can take into account this difference of conversational style to improve performance w.r.t. both model utility and safety for more practical scenarios.

8 Ethics Impact

This paper develops a lightweight and efficient method to strengthen existing LLMs with additional safety properties to defend against new jailbreak attacks. We empirically show that the developed method can effectively and efficiently equip the LLM with the deep safety alignment property and further improve the original model's capacity against many types of jailbreaks. This research aims to enhance the safety of large models and to contribute positively to the broader field of AI research. We remark that the development of SSD only uses publicly available jailbreak prompts and do not create new ones. In the paper, only one jailbreak input query is exhibited in an abstracted way, for illustration purpose. We will also release the codes to facilitate red-teaming efforts on LLMs.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *Technical report*.

Gabriel Alon and Michael Kamfonas. 2023. Detecting language model attacks with perplexity. *arxiv preprint arxiv:2308.14132*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Danny Hernandez, Tristan Hume, Scott Johnston, and 10 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arxiv preprint arxiv:2204.05862*.

Somnath Banerjee, Sayan Layek, Soham Tripathy, Shanu Kumar, Animesh Mukherjee, and Rima Hazra. 2025. Safeinfer: Context adaptive decoding time safety alignment for large language models. In *AAAI*.

Bochuan Cao, Yuanpu Cao, Lu Lin, and Jinghui Chen. 2024. Defending against alignment-breaking attacks via robustly aligned llm. In *ACL*.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arxiv:2310.08419*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, and 1 others. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90quality.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. 2024. Controlled text generation via language model arithmetic. In *ICLR*.

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. 2023a. Masterkey: Automated jailbreak across multiple large language model chatbots. *arXiv preprint arXiv:2307.08715*.

Haikang Deng and Colin Raffel. 2023. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11781–11791.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2023b. Multilingual jailbreak challenges in large language models. *arXiv preprint arxiv:2310.06474*.

Aladin Djuhera, Swanand Ravindra Kadhe, Farhan Ahmed, Syed Zawad, and Holger Boche. 2025. Safemerge: Preserving safety alignment in fine-tuned large language models via selective layer-wise model merging. *arxiv preprint arxiv:2503.17239*.

Alec Helbling, Mansi Phute, Matthew Hull, and Duen Horng Chau. 2023. Llm self defense: By self examination, llms know they are being tricked. *arXiv preprint arxiv:2308.07308*.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arxiv:2309.00614*.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, and 1 others. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. 2021. Alignment of language agents. *arXiv preprint arXiv:2103.14659*.

Minbeom Kim, Hwanhee Lee, Kang Min Yoo, Joonsuk Park, Hwaran Lee, and Kyomin Jung. 2023. Critic-guided decoding for controlled text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. 2018. Scalable agent alignment via reward modeling: a research direction.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*.

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. 2023. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. *ArXiv preprint*, abs/2312.01552.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations*.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024a. Tuning language models by proxy. In *First Conference on Language Modeling*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers)*.

Quan Liu, Zhenhong Zhou, Longzhu He, Yi Liu, Wei Zhang, and Sen Su. 2024b. Alignment-enhanced decoding: Defending jailbreaks via token-level adaptive refining of probability distributions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2023. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arxiv:2310.04451*.

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D. Manning. 2024. An emulator for fine-tuning large language models using small language models. In *The Twelfth International Conference on Learning Representations*.

Zvi Mowshowitz. 2022. Jailbreaking chatgpt on release day. https://www.lesswrong.com/posts/RYcoJdvmoBbi5Nax7/jailbreaking-chatgpt-on-release-day.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Jonathan Pei, Kevin Yang, and Dan Klein. 2023. Preadd: prefix-adaptive decoding for controlled text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2025. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations (ICLR)*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *ICLR*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *arxiv preprint arxiv:2310.03684*.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems*.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint abs/2307.09288*.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023a. Jailbroken: How does llm safety training fail? In *Advances in Neural Information Processing Systems*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Zeming Wei, Yifei Wang, and Yisen Wang. 2023b. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*.

Fangzhao Wu, Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, and Xing Xie. 2023. Defending chatgpt against jailbreak attack via self-reminder. *Nature Machine Intelligence*, 5:1486–1496.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. 2024. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, and 1 others. 2023. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales. *arXiv preprint arXiv:2308.01320*.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2023. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arxiv:2309.10253*.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. 2024. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. *arXiv preprint arxiv:2401.06373*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024a. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

Xiao Zhang and Ji Wu. 2024. Dissecting learning and forgetting in language model finetuning. In *The Twelfth International Conference on Learning Representations*.

Ziyang Zhang, Qizhen Zhang, and Jakob Foerster. 2024b. Parden, can you repeat that? defending against jailbreaks via repetition. *arXiv preprint arXiv:2405.07932*.

Xuandong Zhao, Xianjun Yang, Tianyu Pang, Chao Du, Lei Li, Yu-Xiang Wang, and William Yang Wang. 2024a. Weak-to-strong jailbreaking on large language models. *arxiv preprint arxiv:2401.17256*.

Zhengyue Zhao, Xiaoyun Zhang, Kaidi Xu, Xing Hu, Rui Zhang, Zidong Du, Qi Guo, and Yunji Chen. 2024b. Adversarial contrastive decoding: Boosting safety alignment of large language models via opposite prompt optimization. *arxiv preprint arxiv:2406.16743*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arxiv:2307.15043*.

A Decoding Scheme and Parameter Update

For every b output tokens, Algorithm 2 computes the running match ratio $\beta_{n/b}$ and determines the decoding scheme for next b tokens. As seen from Figure 3, the two models behave more similarly when conditioned on more output tokens. As such, if the decoding scheme keep unchanged, we adjust the match ratio threshold, and also the strength parameter in the Intersection state, in an annealing way, to maintain model utility. If the decoding scheme changes, these parameter values are reset.

Algorithm 2 Parameter Update

```
Require: current match ratio $\beta_{n/b}$, initial match ratio threshold $\beta^0$, minimum strength weight $\alpha_I^{\min} > 0$, decay parameters $\beta^d, \alpha^d > 0$
if $n/b = 1$ then
    $\beta_{th} \leftarrow \beta^0$
end if
if $\beta_{n/b} \leq \beta_{th}$ then
    Set decoding scheme to Intersection.
else
    Set decoding scheme to Union.
end if
if decoding scheme unchanged then
    $\beta_{th} \leftarrow \max\left(0, \beta_{th} - \beta^d\right)$
    if Intersection then
        $\alpha_I \leftarrow \max\left(\alpha_I^{\min},\ \alpha_I - \alpha^d\right)$
    end if
else
    $\beta_{th} \leftarrow \beta^0$
    $\alpha_I \leftarrow \alpha_I^0$
end if
```

B Detailed Experimental Setups and Results

B.1 Other Parameter Choices

In our experiments, the choices of the parameters in Algorithm 2 are: $\beta^0 = 0.6, \beta^d = 0.1, \alpha_I^{\min} = 0.3, \alpha^d = 0.15$.. In addition, we set target sample space size to $c = 10$. The lookahead in speculative sampling is $T = 3$ and the bin size for computing the match ratio is $b = 7$.

B.2 Experimental Results with Other Types of Jailbreak Attacks

As shown in Table 5, SSD demonstrates good defense effectiveness against all five types of jailbreak attacks on the Llama2 series models. On Vicuna, although it achieves almost 0% ASR on HEx-PHI and Advbench, and performs better than several existing defense methods on the other three types of jailbreak attacks, there is still a slight performance gap compared to Deep-Align and SafeDecoding.

B.3 Setup of Attack Methods

We utiliz **Harmful HEx-PHI** dataset (Qi et al., 2025) to conduct the prefilling attack. This dataset consists of 330 harmful instructions extracted from the HEx-PHI safety benchmark, with harmful answers generated using a jailbroken version of GPT-3.5-Turbo. In this study, the dataset is used for prefilling attacks, following the same methodology as Qi et al. (2025) by concatenating harmful answers of varying lengths (i.e., 10, 20, and 40 tokens) with the harmful instructions. For **GCG** (Zou et al., 2023) and **PAIR** (Chao et al., 2023), we follow Chao et al. (2023); Xu et al. (2024) and utilize 50 distinct representative harmful queries[2] from **Advbench** (Zou et al., 2023) to generate specific attack prompts for each model. Due to limitation on computational resources, we use top-$k = 64$ setting for the GCG attack on Llama2-13b-chat. The rest hyperparameters are consistent with those described in the original paper. For **DeepInception**, we apply the ready-to-use template prompt provided in the Github repository[3]. **HEx-PHI** safety benchmark contains 330 harmful instructions specifically designed for LLM harmfulness evaluation.

B.4 Setup of Deep-Align

We train the Llama2-7b and Vicuna models with deep safety alignment properties using the same hyperparameters and datasets as the default settings in Qi et al. (2025). For the TinyLlama used as the small expert model m, we adjust the learning rate of the hyperparameters to 2×10^{-4}.

[2] https://github.com/patrickrchao/JailbreakingLLMs

[3] https://github.com/tmlr-group/DeepInception

Table 5: ASR (%) and harmful score for different defense methods on five benchmark datasets.

Model	Defense Method	PAIR↓	GCG↓	DeepInception↓	HEx-PHI↓	Advbench↓
Llama2-7b	No Defense	4% (1.34)	20% (2.40)	4% (1.22)	0% (1.02)	0% (1.00)
	SafeDecoding	4% (1.18)	0% (1.00)	0% (1.00)	0% (1.04)	0% (1.00)
	Paraphrase	0% (1.10)	2% (1.10)	0% (1.02)	0% (1.05)	0% (1.02)
	ICD	0% (1.00)	4% (1.16)	0% (1.00)	0% (1.04)	0% (1.00)
	Self-Examination	0% (1.00)	8% (1.40)	2% (1.08)	0% (1.00)	0% (1.00)
	Deep-Align	4% (1.18)	2% (1.14)	0% (1.02)	0% (1.00)	0% (1.00)
	SSD	4% (1.30)	6% (1.44)	0% (1.10)	0% (1.00)	0% (1.00)
Llama2-13b	No Defense	4% (1.40)	2% (1.08)	0% (1.16)	0% (1.06)	0% (1.00)
	SafeDecoding	4% (1.26)	0% (1.02)	0% (1.14)	0.30% (1.06)	0% (1.00)
	Paraphrase	0% (1.10)	0% (1.08)	0% (1.24)	0% (1.05)	0% (1.00)
	ICD	0% (1.00)	0% (1.02)	0% (1.00)	0% (1.03)	0% (1.00)
	Self-Examination	2% (1.22)	0% (1.00)	0% (1.16)	0% (1.05)	0% (1.00)
	SSD	2% (1.26)	0% (1.06)	0% (1.16)	0% (1.04)	0% (1.00)
Vicuna	No Defense	80% (4.72)	86% (4.86)	58% (4.34)	10.30% (1.72)	4% (1.38)
	SafeDecoding	4% (1.28)	0% (1.14)	0% (1.08)	0.91% (1.17)	0% (1.00)
	Paraphrase	24% (2.32)	42% (3.18)	40% (3.92)	13.33% (1.94)	2% (1.22)
	ICD	24% (2.34)	48% (3.16)	38% (4.18)	3.94% (1.23)	0% (1.00)
	Self-Examination	8% (1.66)	6% (1.48)	48% (4.04)	7.27% (1.57)	0% (1.22)
	Deep-Align	2% (1.08)	0% (1.00)	0% (1.02)	0% (1.01)	0% (1.00)
	SSD	18% (2.14)	10% (1.40)	8% (2.30)	0.91% (1.18)	0% (1.00)