Reward-Shifted Speculative Sampling Is An Efficient Test-Time Weak-to-Strong Aligner

Anonymous ACL submission

Abstract

Aligning large language models (LLMs) with human preferences has become a critical step in their development. Recent research has increasingly focused on test-time alignment, where additional compute is allocated during inference to enhance LLM safety and reasoning capabilities. However, these test-time alignment techniques often incur substantial inference costs, limiting their practical application. We are inspired by the speculative sampling acceleration, which leverages a small draft model to efficiently predict future tokens, to address the efficiency bottleneck of test-time alignment. We introduce the reward-Shifted Speculative Sampling (SSS) algorithm, in which the draft model is aligned with human preferences, while the target model remains unchanged. We theoretically demonstrate that the distributional shift between the aligned draft model and the unaligned target model can be exploited to recover the RLHF optimal solution without actually obtaining it, by modifying the acceptance criterion and bonus token distribution. Our algorithm achieves superior gold reward scores at a significantly reduced inference cost in test-time weak-to-strong alignment experiments, thereby validating both its effectiveness and efficiency.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in tasks such as instruction-following (Lou et al., 2024), reasoning (Fei et al., 2024), and coding (Wang et al., 2024a). However, the practical deployment of LLMs is constrained by concerns regarding the safety and helpfulness of their outputs (Bai et al., 2022; Weidinger et al., 2022; Deshpande et al., 2023). As a result, efficiently aligning LLMs with human preferences becomes a critical challenge in LLM development. While classical reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022) has shown great potential, it often suffers from the high computational cost of RL training and is prone to instability (Chen et al., 2024; Mohammadi, 2024).

To mitigate these issues, test-time alignment (Khanov et al., 2024; Li et al., 2024a; Qiu et al., 2024) has emerged as a training-free alternative to RLHF, allocating more compute during inference to improve alignment quality. Nevertheless, most test-time alignment approaches require interaction between the LLM and an external reward model (RM) throughout decoding, resulting in either excessive LLM calls (best-of-N and rejection sampling) or too many RM calls (Deng and Raffel, 2023; Khanov et al., 2024). Although recent studies have recognized these inefficiencies and proposed acceleration techniques (Li et al., 2024a; Qiu et al., 2024; Sun et al., 2024), these methods still rely on external RMs and require numerous calls to large models.

We are inspired by the *speculative sampling* algorithm (Chen et al., 2023; Leviathan et al., 2023), which leverages a small "draft" model to efficiently predict K future tokens and then uses a large "target" model to verify them in parallel. While speculative sampling has been widely used for accelerating LLM inference, its potential for test-time alignment remains under-explored. For example, Nakshatri et al. (2024) applies speculative sampling to accelerate the base model inference within a reward-guided search framework. However, their approach continues to rely on an external reward model during inference, resulting in a pipeline that remains both complex and latency-prone.

We hereby propose a novel combination of test-time alignment and speculative sampling: *aligning the draft model to generate high-reward tokens, while utilizing the target model for verification to ensure fluency*. This approach removes the dependence on external reward models. We introduce the reward-Shifted Speculative Sampling (SSS) algorithm, which employs an aligned draft model

and an unaligned target model. Furthermore, we revise both the acceptance criterion and the residual distribution for bonus tokens, ensuring that SSS recovers the RLHF optimal solution. Extensive experiments on test-time weak-to-strong alignment tasks demonstrate that SSS achieves superior gold reward scores compared to existing baselines at a significantly reduced inference cost.

The main contributions of this paper are summarized as follows:

- We redefine the objective of speculative sampling from recovering the target model distribution to recovering the RLHF optimal solution. This enables test-time alignment with a reward-shifted draft model.
- As a test-time alignment method, SSS eliminates the requirement for external reward models. To the best of our knowledge, this is the first approach that implements test-time alignment using only a draft and a target model.
- We show that SSS is an efficient algorithm for test-time weak-to-strong alignment, which enhances the alignment quality at a much lower computational cost compared to other test-time alignment methods.

2 Preliminaries

2.1 RLHF and Reward-Shifted Decoding

The problem of LLM alignment has been modeled as a KL-constrained reward maximization process (Peters and Schaal, 2007; Korbak et al., 2022; Go et al., 2023; Rafailov et al., 2023), in which the reward r is maximized with the proximity constraint from the reference model π_{ref}:

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{x\in\mathcal{D}_p, y\sim\pi_{\boldsymbol{\theta}}(\cdot|x)} r(x,y) - \lambda \cdot \mathbf{KL}(\pi_{\boldsymbol{\theta}} \| \pi_{\text{ref}}). \tag{1}$$

Here, the training data only contains a prompt set $\mathcal{D}_p$, and the responses are generated by the LLM itself. The above process is the objective of reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022). We demonstrate in Section B.1 that the optimal solution of Eq. (1) is:

$$\pi^\star(y|x) \propto \pi_{\text{ref}}(y|x) \cdot \exp\left(\frac{1}{\beta} r(x,y)\right), \tag{2}$$

where the base model policy π_{ref} is slightly shifted to pursue higher reward.

Built upon the above objective, another direct approach is to sample multiple responses and only keep the responses that satisfy the reward constraint, known as test-time alignment (Khanov et al., 2024; Li et al., 2024a; Qiu et al., 2024). Test-time alignment only modifies the decoding process of base models to pursue higher reward, namely a reward-shifted decoding process. There are two basic approaches to reward-shifted decoding: i) best-of-N, which generates multiple candidates in parallel and selects only the best one: $\max_{y\sim\pi_{\text{ref}}(\cdot|x)} r(x,y)$, and ii) rejection sampling, which continues generating proposals until a reward threshold is met: $y \sim \pi_{\text{ref}}(\cdot|x), \quad s.t.\ r(x,y) \geq \tau_r$. These approaches are often applied to different granularities (Khanov et al., 2024; Li et al., 2024a) and even combined together (Qiu et al., 2024; Sun et al., 2024) for effective and efficient alignment.

2.2 Speculative Sampling

Speculative sampling is an inference acceleration method for autoregressive models (Chen et al., 2023). It leverages a much smaller "draft" model π_{draft} to sequentially sample candidate token sequences: $\hat{y}_{t:t+K} \sim \pi_{\text{draft}}(\cdot|x, y_{<t})$, and requires a larger and more powerful "target" model π_{ref} to verify such candidates. Specifically, speculative sampling accepts a draft token with the probability:

$$p_{\text{accept}}(t) = \min\left(1, \frac{\pi_{\text{ref}}(\hat{y}_t|x, y_{<t})}{\pi_{\text{draft}}(\hat{y}_t|x, y_{<t})}\right), \tag{3}$$

where higher target model likelihoods lead to higher chances of acceptance. The complete procedure of speculative sampling is summarized in Algorithm 2.

Speculative sampling has an acceleration trick to address the efficiency issue caused by a potentially low acceptance rate, called *bonus token*. It additionally accepts one more token from the following residual distribution:

$$\pi_{\text{bonus}}(\cdot|x, y_{<t}) = \left(\pi_{\text{ref}}(\cdot|x, y_{<t}) - \pi_{\text{draft}}(\cdot|x, y_{<t})\right)_+, \tag{4}$$

where $(f(x))_+ = \frac{\max(0, f(x))}{\sum_x \max(0, f(x))}$ is the clamp normalization operator. The bonus token distribution ensures that speculative sampling recovers the target model distribution, as proved in Theorem 1 of Chen et al. (2023). This approach ensures that speculative sampling accepts at least one token per target model call, making it no slower than vanilla decoding in most cases. The procedure of speculative sampling is shown in Fig. 3.

3 Methodology

Most of test-time alignment methods require a reward model as the signal of human preference (Khanov et al., 2024; Li et al., 2024a; Qiu et al., 2024). The additional computation induced by such a two-model decoding process makes test-time alignment not efficient enough for time-intensive LLM serving (Wang et al., 2024b), which calls for new frameworks to further accelerate the reward-guided decoding process. We are inspired by speculative sampling (Chen et al., 2023), which accelerates autoregressive decoding via a draft-then-verify procedure. We propose to shift the small draft model to align with human preferences (computationally cheap) and design a new speculative sampling algorithm to simulate the distribution of an aligned target model without actually obtaining it. We also prove that the proposed algorithm recovers the RLHF optimal solution.

3.1 Shifting Draft Models to Align with Human Preferences

The first step of the proposed framework is obtaining a well-aligned draft model to reflect human preference. Following the settings of Tao and Li (2024), we start from a SFT checkpoint $\pi_{\text{draft}}^{\text{SFT}}$ fine-tuned on the chosen responses of preference data, and align this model via direct preference optimization (DPO) (Rafailov et al., 2023). We assume that the aligned draft model π_{draft}^{r} follows the RLHF optimal solution for $\pi_{\text{draft}}^{\text{SFT}}$:

$$\pi_{\text{draft}}^{r}(y|x) \approx \pi_{\text{draft}}^{\text{SFT}}(y|x) \cdot \exp\left(\frac{1}{\beta} r(x,y)\right). \quad (5)$$

However, shifting the draft model will enlarge its gap from the target model, and consequently lowers the acceptance rate (Hong et al., 2025). As visualized in Fig. 1, aligning draft models to human preference is at the cost of a significant distributional shift. Directly applying the shifted draft model π_{draft}^{r} to standard speculative sampling (Algorithm 2) would result in severe efficiency issues. This is because standard speculative sampling only recovers the unaligned target model's distribution, leading to a low acceptance rate due to the distribution shift between the draft and target models, as shown in Table 1. This drawback motivates us to propose a new speculative sampling algorithm to raise the acceptance rate and ensure that generated responses recover the aligned target model's distribution without actually obtaining it.

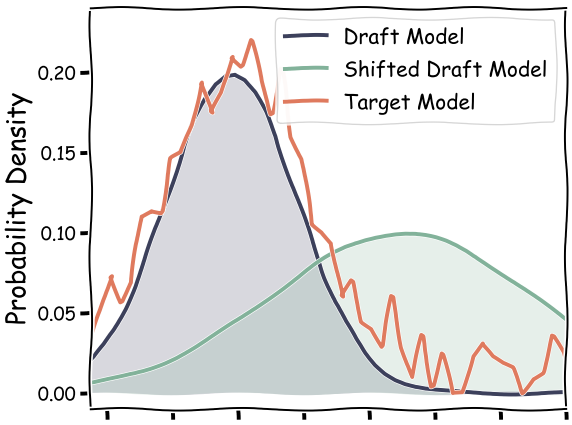

Figure 1: Shifted draft model reflects human preferences, but presents a gap from unaligned target model.

Table 1: Standard speculative sampling with a shifted draft model suffers from low acceptance rate.

Target Model	Draft Model	Acceptance Rate
OPT-6.7B	OPT-125M	0.33
	OPT-125M aligned	0.08 (↓ 76%)
OPT-13B	OPT-350M	0.42
	OPT-350M aligned	0.13 (↓ 69%)

3.2 Speculative Sampling under Draft-Target Distributional Shift

The aforementioned distributional shift caused by aligning draft models to human preferences is a critical problem in reward-shifted speculative sampling. To resolve this, we propose a new speculative sampling algorithm that utilizes such a distributional shift to recover the optimal solution of RLHF, which is typically obtainable by training the target model on preference data (Schulman et al., 2017; Rafailov et al., 2023; Shao et al., 2024). We eliminate the need for training target models and significantly reduce the decoding cost compared with previous test-time alignment methods (Khanov et al., 2024; Li et al., 2024a; Qiu et al., 2024).

As discussed in Section 2.2, the bonus token sampled from the residual distribution π_{bonus} (Eq. (4)) guarantees that standard speculative sampling recovers the distribution of target model π_{ref} (Chen et al., 2023). Intuitively, it compensates for the influence of draft model distribution π_{draft} when draft tokens are rejected. In our proposed new speculative sampling algorithm, the acceptance probability is modified as:

$$p_{\text{accept}}(t) = \min\left(1, \frac{\pi_{\text{ref}}(\hat{y}_t|x, y_{<t})}{\pi_{\text{draft}}^{\text{SFT}}(\hat{y}_t|x, y_{<t})}\right), \quad (6)$$

and the residual distribution for bonus tokens also

has a new form:

$$\pi^r_{\text{bonus}}(\cdot|x, y_{<t+K'}) = \left(\pi^r_{\text{draft}}(\cdot|x, y_{<t+K'}) \left(\frac{\pi_{\text{ref}}(\cdot|x, y_{<t+K'})}{\pi^{\text{SFT}}_{\text{draft}}(\cdot|x, y_{<t+K'})} - 1\right)\right)_+ \tag{7}$$

where K' is the number of actually accepted draft tokens. The entire speculative sampling process is detailed in Algorithm 1. Compared to the standard version (Section 2.2), the condition and reactions for rejecting draft tokens are modified, and no additional token is sampled from the target model once all draft tokens are accepted, since our objective is not the target model distribution.

Our new speculative sampling handles inference acceleration and preference alignment simultaneously. It leverages a shifted draft model π^r_{draft} to generate human-preferred draft tokens, and uses a unaligned target model π_{ref} to ensure the fluency of responses. We also guarantee that *the proposed algorithm recovers the RLHF optimal solution* (Eq. (2)), as discussed in Theorem 1 and proved in Section B.2.

Theorem 1 (SSS recovers RLHF optimal solution) *Shifted speculative sampling (SSS) as demonstrated in Algorithm 1 recovers the RLHF optimal solution in Eq. (2). Specifically, assume a well-aligned draft model* $\pi^r_{draft}(y|x) = \pi^{SFT}_{draft}(y|x) \cdot \exp\left(\frac{1}{\beta} r(x, y)\right)$, *the probability that SSS generates response* y *given prompt* x *is:*

$$\mathbf{P}(Y = y|x) \equiv \pi^\star(y|x).$$

Algorithm 1 Shifted Speculative Sampling

Require: autoregressive target model π_{ref} and autoregressive r-shifted draft model π^r_{draft};
Require: initial prompt x and lookahead $K > 0$.
$y \leftarrow \emptyset$
$t \leftarrow 0$
while $t <$ `max_length` **do**
  for k = 1 : K **do**
    $\hat{y}_{t+k} \sim \pi^r_{\text{draft}}(\cdot|x, y_{<t}, \hat{y}_{t:t+k-1})$
  end for
  $\pi_{\text{ref}}(\hat{y}_{t:t+K}|x, y_{<t})$ ▷ Target Likelihoods
  for k = 1 : K **do**
    $\epsilon \sim \mathcal{U}[0, 1]$
    if $\epsilon < p_{\text{accept}}(t)$ in Eq. (6) **then**
      $y_{t+k} \leftarrow \hat{y}_{t+k}$ ▷ Accept
    else
      $y_{t+k} \leftarrow \pi^r_{\text{bonus}}(\cdot|x, y_{<t+k})$ in Eq. (7)
      $t \leftarrow t + k$ and break
    end if
  end for
  if all draft tokens are accepted **then**
    $t \leftarrow t + K$
  end if
end while

4 Empirical Results

We provide comprehensive experimental results in Section C, to evaluate the alignment quality and efficiency of the propose method. Our method improves the gold reward at a significantly reduced inference cost compared to other test-time alignment methods.

5 Conclusion and Limitations

This paper introduces a novel speculative sampling algorithm designed for efficient test-time alignment. Our approach involves shifting the draft model to align with human preferences, thereby intentionally creating a distributional shift between the draft and target models. This shift is leveraged to simulate the distribution of a well-aligned target model (i.e., the RLHF optimal solution). We modify the standard speculative sampling algorithm by introducing a new acceptance criterion and a new bonus token distribution, ensuring that our algorithm recovers the RLHF optimal solution. Compared to existing test-time weak-to-strong alignment methods, our algorithm achieves superior alignment quality (measured by gold reward) while substantially reducing inference costs.

Despite these promising results, the effectiveness of our algorithm depends on the assumption that the shifted draft model is well-aligned: $\pi^r_{\text{draft}}(y|x) \approx \pi^r_{\text{draft}}(y|x) \cdot \exp\left(\frac{1}{\beta} r(x, y)\right)$. This assumption is sensitive to the post-training process of the draft models. Accurately verifying this assumption is also challenging due to the unknown reward function, making the empirical performance of our algorithm contingent on the tuning of hyperparameters during draft model post-training. Furthermore, employing a small draft model to capture human preferences may lead to generalization issues: a draft model aligned for one task may not perform well on others. We plan to address these limitations in future works.

Statement of Reproducibility

The code used for inference experiments is included in the supplementary material. It will be publicly available upon acceptance.

Statement of AI Assistant Usage

The construction of the codebase partially relied on AI assistant for debugging, and the writing of this paper was polished by AI assistant.

References

argsearch. 2024. Llama 7b reward model (float32). https://huggingface.co/argsearch/llama-7b-rm-float32. Accessed: 2025-05-01.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. In *International Conference on Machine Learning*, pages 5209–5235. PMLR.

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*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024. How is chatgpt's behavior changing over time? *Harvard Data Science Review*, 6(2).

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Haikang Deng and Colin Raffel. 2023. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In *Conference on Empirical Methods in Natural Language Processing*.

Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik R Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. In *Empirical Methods in Natural Language Processing*.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, and 1 others. 2024. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12622–12642.

Hao Fei, Yuan Yao, Zhuosheng Zhang, Fuxiao Liu, Ao Zhang, and Tat-Seng Chua. 2024. From multimodal llm to human-level ai: Modality, instruction, reasoning, efficiency and beyond. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024): Tutorial Summaries*, pages 1–8.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. 2024. Break the sequential dependency of llm inference using lookahead decoding. *arXiv preprint arXiv:2402.02057*.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, and 1 others. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Dongyoung Go, Tomasz Korbak, Germàn Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. 2023. Aligning language models with preferences through f-divergence minimization. In *International Conference on Machine Learning*, pages 11546–11583. PMLR.

Fenglu Hong, Ravi Shanker Raju, Jonathan Lingjie Li, Bo Li, Urmish Thakker, Avinash Ravichandran, Swayambhoo Jain, and Changran Hu. 2025. Training domain draft models for speculative decoding: Best practices and insights. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. 2024. Args: Alignment as reward-guided search. In *The Twelfth International Conference on Learning Representations*.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. 2022. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Bolian Li, Yifan Wang, Anamika Lochab, Ananth Grama, and Ruqi Zhang. 2024a. Cascade reward sampling for efficient decoding-time alignment. *arXiv preprint arXiv:2406.16306*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024b. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*.

Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. 2025. Reward-guided speculative decoding for efficient llm reasoning. *arXiv preprint arXiv:2501.19324*.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. 2024. Online speculative decoding. In *Proceedings of the 41st International Conference on Machine Learning*, pages 31131–31146.

Renze Lou, Kai Zhang, and Wenpeng Yin. 2024. Large language model instruction following: A survey of progresses and challenges. *Computational Linguistics*, 50(3):1053–1095.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, and 1 others. 2024. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 932–949.

Behnam Mohammadi. 2024. Creativity has left the chat: The price of debiasing language models. *Available at SSRN 4858364*.

Nishanth Nakshatri, Shamik Roy, Rajarshi Das, Suthee Chaidaroon, Leonid Boytsov, and Rashmi Gangadharaiah. 2024. Constrained decoding with speculative lookaheads. *arXiv preprint arXiv:2412.10418*.

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. *Advances in neural information processing systems*, 35:27730–27744.

Jan Peters and Stefan Schaal. 2007. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pages 745–750.

Jiahao Qiu, Yifu Lu, Yifan Zeng, Jiacheng Guo, Jiayi Geng, Huazheng Wang, Kaixuan Huang, Yue Wu, and Mengdi Wang. 2024. Treebon: Enhancing inference-time alignment with speculative tree-search and best-of-n sampling. *arXiv preprint arXiv:2410.16033*.

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. *Advances in Neural Information Processing Systems*, 36:53728–53741.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast best-of-n decoding via speculative rejection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Leitian Tao and Yixuan Li. 2024. Your weak llm is secretly a strong teacher for alignment. *arXiv preprint arXiv:2409.08813*.

Lun Wang, Chuanqi Shi, Shaoshuai Du, Yiyi Tao, Yixian Shen, Hang Zheng, and Xinyu Qiu. 2024a. Performance review on llm for solving leetcode problems. In *2024 4th International Symposium on Artificial Intelligence and Intelligent Manufacturing (AIIM)*, pages 1050–1054. IEEE.

Yuxin Wang, Yuhan Chen, Zeyu Li, Zhenheng Tang, Rui Guo, Xin Wang, Qiang Wang, Amelie Chi Zhou, and Xiaowen Chu. 2024b. Towards efficient and reliable llm serving: A real-world workload study. *arXiv e-prints*, pages arXiv–2401.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, and 1 others. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 214–229.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2024. Draft& verify: Lossless large language model acceleration via self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11263–11282.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. 2023. Distillspec: Improving speculative decoding via knowledge distillation. *arXiv preprint arXiv:2310.08461*.

A Related Work

A.1 Test-Time Alignment

Early alignment research focused on post-training LLMs with techniques like RLHF (Ouyang et al., 2022), but the high cost of policy optimization methods like PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) has motivated the test-time (or decoding-time) alignment approaches that leave the LLMs' parameters frozen. For example, best-of-N (BoN) generates multiple complete responses and select the highest-reward one; rejection sampling continues generating proposal responses until a reward threshold is met. Built upon these 2 techniques, test-time alignment methods at varying granularities are developed. For example, token-level reward-guided search (Deng and Raffel, 2023; Khanov et al., 2024), segment-level rejection sampling (Li et al., 2024a), and segment-level MC tree search (Qiu et al., 2024) have all been explored. However, although they improve the efficiency of test-time alignment, they still rely on external reward models to interact with the decoding process, which is the major cause of the inefficiency. Our algorithm (SSS) jumps out of the frameworks with reward models, and leverages a small draft model to reflect human preference. The interaction between draft and target models are far more efficient than the traditional LLM-RM framework.

A.2 Speculative Sampling

Orthogonal to test-time alignment, speculative decoding (Chen et al., 2023; Leviathan et al., 2023) accelerates LLM generation by using a lightweight draft model to guess K future tokens, which is then verified by a large target model in parallel. Speculative sampling recovers the exact distribution of the target model (Chen et al., 2023). Recent work has aimed to further improve the efficiency of this draft–verify framework. One major direction is increasing the number of candidate tokens accepted by the target model. To this end, tree-based methods (Miao et al., 2024; Li et al., 2024b; Fu et al., 2024) generate multiple draft token paths in parallel, increasing the likelihood of acceptance and reducing verification overhead. Other studies enhance draft token quality through knowledge distillation (Zhou et al., 2023; Liu et al., 2024), layer-skipped decoding (Elhoushi et al., 2024; Zhang et al., 2024), or by adding specialized speculative heads such as MEDUSA (Cai et al., 2024) to improve token prediction and verification efficiency. Some recent variants also explore using reward models (RMs) as verifiers instead of target LLMs (Liao et al., 2025). This change allows for higher token acceptance rates by relaxing fluency constraints. Nakshatri et al. (2024) applies speculative sampling to accelerate constrained decoding, still relying on external reward models to enforce the constraints. Despite these advances, existing speculative decoding methods are primarily designed for computational acceleration and assume an *unaligned* draft model. As a result, the question of how to incorporate human preference alignment into the speculative decoding process without sacrificing efficiency remains largely under-explored.

B Proofs

B.1 RLHF Optimal Solution

Starting from the RLHF objective as demonstrated in Eq. (1), the KL-constrained reward maximization can be re-written as:

$$\begin{aligned}
&\frac{1}{\beta}\mathbb{E}_{y\sim\pi_{\boldsymbol{\theta}}(\cdot|x)}r(x,y) - \mathbf{KL}(\pi_{\boldsymbol{\theta}}(\cdot|x)\|\pi_{\text{ref}}(\cdot|x)) \\
&= \sum_y \pi_{\boldsymbol{\theta}}(y|x)\cdot\left(\log\frac{\pi_{\text{ref}}(y|x)}{\pi_{\boldsymbol{\theta}}(y|x)} + \frac{1}{\beta}r(x,y)\right) \\
&= \sum_y \pi_{\boldsymbol{\theta}}(y|x)\cdot\log\frac{\pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)}{\pi_{\boldsymbol{\theta}}(y|x)} \\
&\propto -\mathbf{KL}(\pi_{\boldsymbol{\theta}}(\cdot|x)\|\pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)).
\end{aligned} \tag{8}$$

Optimizing a model $\pi_{\boldsymbol{\theta}}$ with RLHF is equivalent to minimizing its KL-divergence from a new policy: $\pi^\star(y|x) = \pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)$, which we call the RLHF optimal solution.

B.2 Theorem 1

The following proof is inspired by Theorem 1 of Chen et al. (2023). We start by considering the probability of generating a token x, and ignore the prompt and previously generated tokens for simplicity. In our shifted speculative sampling as demonstrated in Algorithm 1, a token can be generated in two cases: i) it is a draft token and is accepted, and i) draft token is rejected and it is sampled as a bonus token. We show these two case in the following formula:

$$\begin{aligned}
&\mathbf{P}(X=x) \\
&= \pi^r_{\text{draft}}(\hat{x}=x)\cdot\mathbf{P}(\hat{x}\text{ accepted}|\hat{x}=x) \\
&\quad + \mathbf{P}(\hat{x}\text{ rejected})\cdot\mathbf{P}(X=x|\hat{x}\text{ rejected}).
\end{aligned} \tag{9}$$

The left half can be re-written as:

$$\begin{aligned}
&\pi^r_{\text{draft}}(\hat{x}=x)\cdot\mathbf{P}(\hat{x}\text{ accepted}|\hat{x}=x) \\
&= \pi^r_{\text{draft}}(\hat{x}=x)\cdot\min\left(1, \frac{\pi_{\text{ref}}(x)}{\pi^{\text{SFT}}_{\text{draft}}(x)}\right) \\
&= \pi^r_{\text{draft}}(\hat{x}=x)\cdot\min\left(1, \frac{\pi^\star(x)}{\pi^r_{\text{draft}}(x)}\right) \\
&= \min\left(\pi^r_{\text{draft}}(x), \pi^\star(x)\right).
\end{aligned} \tag{10}$$

For the right half, we first transform the probability of rejection to be:

$$\begin{aligned}
\mathbf{P}(\hat{x}\text{ rejected}) &= 1 - \mathbf{P}(\hat{x}\text{ accepted}) \\
&= 1 - \sum_{x'}\mathbf{P}(\hat{x}=x', \hat{x}\text{ accepted}) \\
&= 1 - \sum_{x'}\min\left(\pi^r_{\text{draft}}(x'), \pi^\star(x')\right) \\
&= \sum_{x'}\pi^\star(x') - \min\left(\pi^r_{\text{draft}}(x'), \pi^\star(x')\right) \\
&= \sum_{x'}\max\left(0, \pi^\star(x') - \pi^r_{\text{draft}}(x')\right).
\end{aligned} \tag{11}$$

Then, the probability of generating x after draft token rejection is:

$$\begin{aligned}\mathbf{P}(X = x | \hat{x} \text{ rejected}) &= \pi^r_{\text{bonus}}(x) \\ &= \left(\pi^r_{\text{draft}}(x)\left(\frac{\pi_{\text{ref}}(x)}{\pi^{\text{SFT}}_{\text{draft}}(x)} - 1\right)\right)_+ \\ &= (\pi^\star(x) - \pi^r_{\text{draft}}(x))_+ \\ &= \frac{\max\left(0, \pi^\star(x) - \pi^r_{\text{draft}}(x)\right)}{\sum_{x'} \max\left(0, \pi^\star(x') - \pi^r_{\text{draft}}(x')\right)},\end{aligned} \tag{12}$$

where $(f(x))_+ = \max(0, f(x))/\sum_x \max(0, f(x))$ is the clamp normalization operator. Finally, taking all above together, the token generation probability is:

$$\begin{aligned}&\mathbf{P}(X = x) \\ &= \min\left(\pi^r_{\text{draft}}(x), \pi^\star(x)\right) + \max\left(0, \pi^\star(x) - \pi^r_{\text{draft}}(x)\right) \\ &= \pi^\star(x),\end{aligned} \tag{13}$$

always equivalent to the RLHF optimal solution.

C Experiments

In this section, we present the experimental settings and empirical results. We first discuss the configurations of experiments in Section C.1, then show the main empirical results in Section C.2, and finally have ablation studies on the draft model selection and alternative algorithm options in Section C.3.

C.1 Experimental Settings

All experiments in this paper are based on the HH-RLHF dataset (Ganguli et al., 2022), which contains paired conversational data on helpfulness and harmlessness to reflect human preference. To ensure that draft models and target models share the same vocabulary and have similar distributions, we choose the 3 model pairs as shown in Table 2. For the gold reward in response evaluation, we choose a large reward model trained on HH-RLHF with `Llama-7B` backbone (argsearch, 2024).

Table 2: Base model choices for draft and target models.

Draft Model	Target Model
`Qwama-0.5B`	`Llama-3-8B`
`OPT-125M`	`OPT-6.7B`
`OPT-350M`	`OPT-13B`

For post-training draft models, we use the TRL library[1] for supervised fine-tuning (SFT) and direct preference optimization (DPO) (Rafailov et al., 2023) with the hyper-parameters in Table 3. These hyper-parameter choices are based on the settings of Tao and Li (2024) and adjusted for better performance by grid search.

For inference experiments, we use the Transformers library[2] with a temperature of 0.8 and a maximum sequence length of 128 tokens. These settings are standard in previous works (Chen et al., 2023; Khanov et al., 2024; Li et al., 2024a).

Additionally, due to the small size of draft models, all experiments can be conducted on one NVIDIA L40S GPU[3].

[1] https://github.com/huggingface/trl.

[2] https://github.com/huggingface/transformers.

[3] https://www.nvidia.com/en-us/data-center/l40s.

Table 3: Hyper-parameters for fine-tuning and post-training draft models on preference data.

Base Model	Pipeline	Learning Rate	Batch Size	Epochs
`Qwama-0.5B`	SFT	2e-4	32	1.00
	DPO	1e-5	16	0.57
`OPT-125M`	SFT	2e-6	32	1.00
	DPO	5e-5	16	1.00
`OPT-350M`	SFT	2e-6	32	0.57
	DPO	5e-5	16	1.00

Time measurements are obtained from the `time.time()` API in Python[4].

C.2 Alignment Quality and Inference Efficiency

Table 4 demonstrates the performance advantages of our method (SSS) across different model pairs on HH-RLHF for a single run. We compare against a range of test-time alignment baselines, including Best-of-N (BoN), TreeBoN (Qiu et al., 2024), and CARDS (Li et al., 2024a). We keep the reward model sizes used in these baselines the same as draft models for fair comparison. The baseline choices represent a spectrum of test-time alignment strategies balancing alignment quality and computational efficiency. We also include comparisons with a standard speculative sampling (SS) approach.

Our method, SSS, consistently achieves the best trade-off between gold reward and efficiency. For example, on `OPT-6.7B`, SSS achieves the highest gold reward (3.88) while requiring only 115 LLM calls and 7.8 seconds per response, which has a 2.9× speedup over BoN. On `Llama-3-8B`, SSS reduces inference time by over 5× compared to BoN, while maintaining competitive gold reward. Compared to CARDS and TreeBoN, SSS offers significantly lower inference cost and latency, with high alignment quality. Among all baselines, our method achieves the highest gold reward on `OPT-13B`, reaching 4.06 while also delivering a 5.1× speedup over BoN, highlighting its strong alignment quality and inference efficiency.

C.2.1 Why SSS achieves alignment quality and inference efficiency simutenously?

As shown in Table 4, our proposed method (SSS) consistently outperforms existing test-time alignment approaches in both alignment quality and decoding efficiency. This is achieved through two key design choices:

- **Draft model alignment reduces reward evaluation cost.** By aligning a small draft model to human preference using DPO (Rafailov et al., 2023), SSS avoids frequent reward model queries during decoding, significantly lowering computational overhead compared to segment-level or prefix-level alignment methods like CARDS (Li et al., 2024a) and TreeBoN (Qiu et al., 2024).
- **Speculative sampling is adapted for alignment.** Standard speculative decoding accelerates generation but lacks alignment. SSS modifies the acceptance rule and residual distribution to account for the distributional shift caused by draft model alignment, enabling efficient decoding while preserving preference consistency.

[4] https://docs.python.org/3/library/time.html.

Table 4: **Comparison of test-time weak-to-strong alignment methods in terms of gold reward, number of LLM calls, and inference time.** "Gold R" refers to the average score assigned by the gold reward model, "# Calls" indicates the number of forward passes through the target model π_{ref} (a proxy for computational cost), and "Time" is the actual wall-clock inference time per response. "Speedup" is computed relative to the BoN baseline for each method.

	Method	Gold R	# Calls	Time (s)	Speedup
Llama-3-8B	Vanilla	5.63	128.0	6.5	–
	BoN-10	6.37	1280	58.0	1.0×
	TreeBoN	**6.44**	841.4	48.1	1.2×
	CARDS	6.41	790.7	45.7	1.3×
	Vanilla SD	5.74	58.6	7.6	**7.6**×
	SSS (our)	6.14	86.2	10.7	5.4×
OPT-6.7B	Vanilla	3.21	128.0	5.5	–
	BoN-5	3.28	640.0	22.7	1.0×
	TreeBoN	3.27	801.6	25.6	0.9×
	CARDS	2.93	414.1	12.9	1.8×
	Vanilla SD	2.00	41.7	3.1	**7.4**×
	SSS (our)	**3.88**	115	7.8	2.9×
OPT-13B	Vanilla	3.13	128.0	9.5	–
	BoN-5	3.49	640.0	69.4	1.0×
	TreeBoN	3.61	968.6	69.4	1.0×
	CARDS	3.35	741.9	50.1	1.4×
	Vanilla SD	3.85	108.5	14.9	4.7×
	SSS (our)	**4.06**	112.5	13.6	**5.1**×

As a result, SSS achieves fast and well-aligned generation without requiring an aligned target model, and effectively recovers the RLHF optimal solution.

C.3 Ablation Studies

C.3.1 How to select a good draft model?

Table 5 presents the ablation studies comparing the key indicators of draft model training across three models. We evaluate each draft model using chosen/rejected/generated likelihoods, implicit reward accuracy, and gold reward. Pretrained draft models tend to achieve higher implicit reward accuracy, especially on OPT-125M and OPT-350M, but often produce lower gold rewards, indicating misalignment with human preferences despite higher token-level likelihood. SFT models sometimes yield stronger gold rewards (e.g., on OPT-350M), but the results are not always consistent across models. DPO consistently delivers the best trade-off across all checkpoints. This highlights that no single metric fully captures alignment quality, and selecting a good draft model requires balancing alignment quality, generation fluency, and reward supervision.

C.3.2 Practical tricks to handle the gap between DPO draft models and the desired well-aligned draft model

The proposed algorithm relies on the assumption that we have a well-aligned draft model $\pi^r_{\text{draft}}(y|x) \approx \pi^{\text{SFT}}_{\text{draft}}(y|x) \cdot \exp\left(\frac{1}{\beta} r(x,y)\right)$. However, this is often hard to obtain and verify since we do not have access to the reward score $r(x,y)$. We find that, even for imperfect shifted reward models, we can still have outstanding alignment quality by slightly modifying

Table 5: **Performance of different draft model training methods using HH-RLHF.** "Lik(Chosen)", "Lik(Rej)", and "Lik(Gen)" denote the average likelihood of the chosen, rejected, and generated responses, respectively. "Imp. Rw" refers to implicit reward accuracy, and "Gold Rw" denotes the average reward from a gold reward model.

Method	Lik(Chosen)	Lik(Rej)	Lik(Gen)	Imp. Rw	Gold Rw
Qwama-0.5B					
DPO	0.5428	0.4984	0.6809	0.43	3.75
SFT	0.4973	0.4607	0.6314	0.43	3.49
Pretrained	0.3448	0.3314	0.5402	0.42	4.29
OPT-125M					
DPO	0.3944	0.3999	0.6285	0.54	3.18
SFT	0.3542	0.3546	0.6282	0.57	3.55
Pretrained	0.3071	0.3003	0.6156	0.59	3.46
OPT-350M					
DPO	0.3251	0.3139	0.5622	0.62	3.45
SFT	0.3875	0.3881	0.6343	0.55	3.59
Pretrained	0.3307	0.3226	0.6174	0.58	3.35

Eq. (7) to be:

$$\left(\pi^r_{\text{draft}}(\cdot|x, y_{<t+K'})^{\gamma}\left(\frac{\pi_{\text{ref}}(\cdot|x, y_{<t+K'})}{\pi^{\text{SFT}}_{\text{draft}}(\cdot|x, y_{<t+K'})} - 1\right)\right)_+, \tag{14}$$

where $\gamma = 1$ is exactly the original version. We test a set of γ values and find that the optimal γ is below 0.5, as shown in Fig. 2.

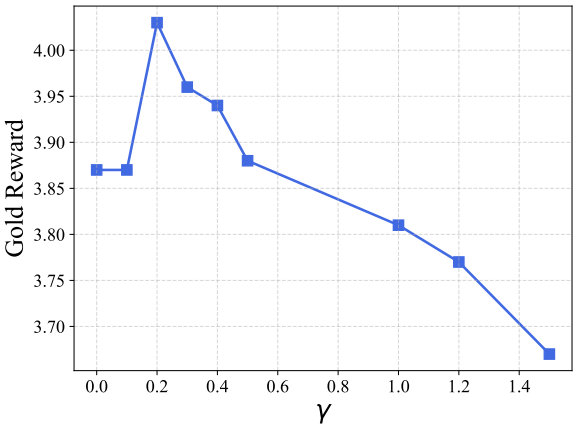

Figure 2: Effect of hyper-parameter $gamma$ to the gold reward on OPT-13B/OPT-350M model. The optimal γ is below 0.5.

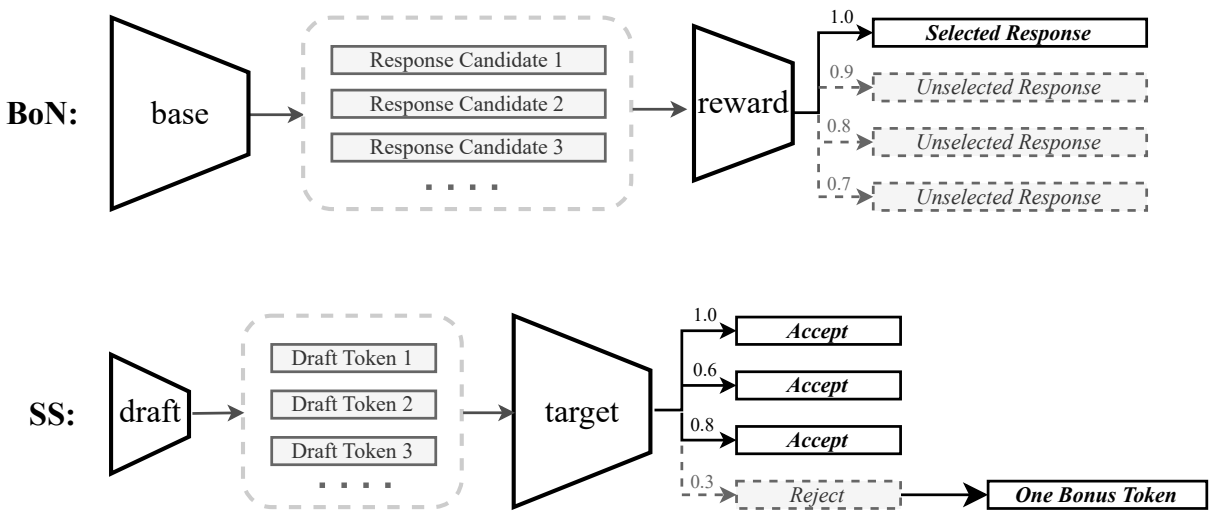

Figure 3: Inference procedures of best-of-N (traditional test-time alignment) and speculative sampling (our new algorithm). Speculative sampling leverages a small draft model to efficiently guess future tokens, which has the potential to be shifted to human preference.

Algorithm 2 Speculative Sampling

Require: autoregressive target model π_{ref} and autoregressive draft model π_{draft};
Require: initial prompt x and lookahead $K > 0$.

```
y ← ∅
t ← 0
while t < max_length do
    for k = 1 : K do
        ŷ_{t+k} ~ π_draft(·|x, y_{<t}, ŷ_{t:t+k−1})
    end for
    π_ref(ŷ_{t:t+K}|x, y_{<t})        ▷ Target Likelihoods
    for k = 1 : K do
        ε ~ U[0, 1]
        if ε < p_accept(t) in Eq. (3) then
            y_{t+k} ← ŷ_{t+k}                ▷ Accept
        else
            y_{t+k} ← π_bonus(·|x, y_{<t+k}) in Eq. (4)
            t ← t + k and break
        end if
    end for
    if all draft tokens are accepted then
        y_{t+K+1} ~ π_ref(·|x, y_{≤t+K})
        t ← t + K + 1
    end if
end while
```