Extended Abstract

Motivation Although instruction-tuned language models perform well across many tasks, their behavior during inference is usually fixed, relying on greedy decoding or static temperature sampling. As a result, inference runs without adapting or receiving feedback during generation. This causes much of the model's generative potential to be underutilized, particularly when handling ambiguous or complex instructions. Motivated by recent progress in self-consistency, reward modeling, and adaptive decoding, we investigate whether we can improve model performance without additional training by making inference more dynamic.

Method We develop three adaptive inference methods for a base language model fine-tuned via Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO). Our approaches leverage a reward model at test time to guide both the generation and selection of responses. The first strategy, reward-weighted self-consistency, samples 20 completions and uses reward scores for probabilistic voting. The second, compute-optimal sampling, dynamically adjusts temperature and top-p sampling based on reward thresholds and stops early when high-quality responses are found. The third, a hybrid strategy, combines adaptive sampling with reward-diversity scoring to select the 'best' response.

Implementation We fine-tune the Qwen2.5-0.5B base model on the SmolTalk dataset using both SFT and DPO. For reward evaluation, we use Llama 3.1 Nemotron-70B. Inference is implemented using vLLM with nucleus sampling, adaptive temperature control, and early stopping criteria. Each method is designed to make more efficient use of test-time compute while prioritizing response quality.

Results Evaluated on 100 held-out prompts from the UltraFeedback dataset, adaptive inference methods significantly outperform both the SFT and DPO baselines. The hybrid strategy achieves a 69.2% win rate and the highest average reward score (-20.094), compared to 46.0% and -27.223 for SFT. Reward-weighted self-consistency achieves 63.9% win rate. Compute-optimal sampling has efficiency gains but results in lower reward and win rates, suggesting room for improvement in its criteria.

Discussion These findings show that the behavior of a model during inference significantly impacts its overall performance. Even without additional training, models can improve significantly when test-time generation is made adaptive and guided by reward feedback. The hybrid method, in particular, shows that combining parameter tuning with diversity-aware selection can result in meaningful performance gains. Our qualitative analysis also shows improvements in coherence, factuality, and diversity.

Conclusion Adaptive inference provides a generalizable way for improving instruction-following in language models. Rather than retraining models, we show that navigating the output space using reward and diversity signals can result in better outputs. This can opens up a new opportunities for post-training optimization that is model-agnostic and applicable across tasks and domains.

Adaptive Test-Time Inference Extensions for Fine-Tuned Instruction-Following Models

Diego Valdez Duran
Department of Computer Science
Stanford University
diegoval@stanford.edu

Abstract

This project investigates whether adaptive inference methods, without additional training, can improve instruction-following performance in fine-tuned language models. Using the Qwen2.5-0.5B base model fine-tuned via SFT and DPO on the SmolTalk dataset, we introduce and evaluate three decoding methods: reward-weighted self-consistency, compute-optimal sampling, and a hybrid approach that dynamically adjusts sampling and utilizes both reward and diversity signals. Experiments on the UltraFeedback dataset show that adaptive inference significantly improves response quality and alignment, with the hybrid method achieving a 69.2% win rate and the highest average reward. These results suggest that inference can be a powerful and underutilized area for post-training optimization.

1 Introduction

Instruction-tuned language models have shown impressive performance across a diverse set of tasks, yet most inference pipelines remain static. Once a model is fine-tuned, it typically generates a single response via greedy decoding, regardless of prompt complexity. However, in practice, there may be multiple candidates that vary in style, factual accuracy, or relevance. Although reinforcement learning with preference data has been effective in aligning large language models, even after fine-tuning, these models infrequently generate or evaluate multiple candidate responses at inference time. Consequently, a significant portion of the model's ability to reason or explore alternative responses is underutilized (8; 11).

In this paper, we investigate whether additional improvements can be obtained from an already fine-tuned checkpoint by efficiently allocating compute at test time (7). Specifically, we start with the Qwen 2.5 0.5B base model, which we fine tuned on the SmolTalk dataset (1) via supervised learning, and treat this checkpoint as a policy that can generate multiple candidate responses for each instruction. Rather than relying on a single greedy pass, we generate a set of diverse responses, score each one with the Llama 3.1 Nemotron-70B reward model, and then select a final answer based on a combination of reward signals and sample diversity. We hypothesize that presenting the model with multiple plausible responses and selecting among them based on reward feedback will improve its instruction following ability, particularly on more challenging or vague prompts (11).

We propose three complementary adaptive test-time inference strategies. First, reward weighted self-consistency generates a fixed batch of 20 samples, computes a reward score for each sample, and then performs a softmax weighted aggregation while adjusting the temperature parameter if the variance of reward scores or token diversity falls outside predetermined thresholds (8). Second, compute-optimal sampling treats inference as a dynamic problem by having the model first generate a small batch of candidates and then evaluate whether any candidate's reward exceeds a moving threshold (7). If not, it continues sampling up to a configurable maximum while adaptively tuning both temperature and top-p towards higher reward outputs. Third, the hybrid approach merges these ideas in two phases. In phase 1, a small sample of five candidates is used to infer optimal sampling parameters for a given instruction. In phase 2, those parameters are fixed as the model generates a larger, diverse set that is scored by a combination of reward and token-overlap diversity.

Our experiments on UltraFeedback (3), using 100 instructions for development, show that the SFT model achieved a win-rate of about 46%, whereas a DPO fine tuned model increases that win-rate to approximately 54%. On the other hand, reward weighted self-consistency achieved a development win-rate of 63.0%, and the hybrid approach further increased it to 69.23%. By using early stopping and adaptive parameter adjustments, the model avoids wasting computational resources on candidates that are unlikely to succeed.

2 Related Work

Self-consistency has been proven to be a lightweight and effective decoding strategy for improving reasoning in language models (8). Wang et al. proposed generating several diverse chain-of-thought responses for each prompt and choosing the most common final answer instead of relying on a single greedy response. This sample and marginalize approach significantly improved performance on math and commonsense reasoning benchmarks without additional re-training. Their findings show that integrating prompt diversity with aggregation leads to improved accuracy, which motivates our reward weighted self-consistency approach.

Recent work has also explored how to allocate compute more efficiently at test time (7). Snell et al. studied two strategies at inference time, iterative revision and search guided by process-based verifiers, and showed that dynamically switching between them based on prompt difficulty yielded higher reward in a FLOPs matched evaluation. Importantly, they show that under the same compute budget, smaller models employing adaptive inference can match or outperform much larger models. This insight also informs our compute-optimal sampling while maintaining performance.

Additionally, prior work has shown significant progress in reward modeling (9). Zhang et al. introduced a generative reward model (GenRM) that reframed binary preference evaluation as a next token prediction task. By prompting the model to generate "Yes" or "No" tokens in response to correctness questions, and incorporating chain of thought rationales, they achieved more coherent and interpretable reward signals. Although we use a discriminative reward model (Nemotron 70B) in our experiments, we draw on similar ideas by treating the reward model's feedback as a guiding signal for sample selection and diversity-aware voting.

Direct Preference Optimization (DPO) has become a popular method for fine-tuning language models with pairwise preference data (5). By optimizing the likelihood ratio between preferred and dispreferred responses, DPO avoids the complexity of full reinforcement learning while still aligning models with preference data. In our project, the DPO fine-tuned checkpoint serves as a benchmark to evaluate the limits of base preference training, helping us assess whether test time inference methods can match or exceed its performance without any further model updates.

3 Method

Our approach consists of three stages: (1) supervised fine-tuning (SFT) on conversational data, (2) preference alignment via Direct Preference Optimization (DPO), and (3) test-time adaptation via reward-informed sampling strategies. Each step builds on the previous, leading to inference-time improvements evaluated using UltraFeedback win-rate comparisons against a base model.

3.1 Supervised Fine-Tuning and Preference Optimization

As the foundation of our project, we begin with the Qwen2.5-0.5B model, focusing on its instruction following capabilities. For SFT, we use the SmolTalk dataset, which contains high-quality chat responses from GPT-4o (1). The model is fine-tuned using a next-token prediction objective, where x is the instruction and $y_{1:T}$ is the target completion. The loss is applied only to the response tokens, with the instruction tokens masked during training. The resulting objective is:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{<t})$$

To align the model with human preferences, we utilize DPO to reinterpret preference learning as a pairwise classification problem (5). DPO mitigates the need for an explicit reward model by implicitly modeling preferences through the difference in log-probabilities between preferred and dispreferred responses, relative to a reference policy. Given an instruction x and a preference pair (y^+, y^-) from the UltraFeedback dataset (3), the DPO loss can be written as:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \left[\log \pi_\theta(y^+ \mid x) - \log \pi_\theta(y^- \mid x)\right]\right)$$

where π_θ is the current policy, σ is the sigmoid function, and β is a temperature scaling factor. This encourages the model to assign a higher likelihood to preferred responses relative to dispreferred ones. And, to align with KL-regularized RL methods, the full DPO loss incorporates a reference model π_{ref} and is written as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)} - \beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)}\right)\right]$$

In doing so, the objective reframes KL-constrained reward maximization as a preference classification task, implicitly balancing reward learning with regularization toward the reference policy.

3.2 Reward Model Scoring

To evaluate responses, we use Llama 3.1 Nemotron-70B, a parametric reward model that assigns scalar reward scores reflecting response quality. Given a prompt x and a candidate response y_i, the reward is:

$$r_i = \text{RM}(x, y_i)$$

where RM is the reward model. These scores approximate human preferences and are used both for evaluation and as signals in our adaptive inference methods.

Usage Across Methods.

- **Basic Generation**: The reward model is used only for evaluation. No reward-based filtering or selection is done.
- **Self-Consistency**: Reward scores are used to filter out low-quality completions (threshold = -25.0) and to compute softmax-weighted voting (temperature = 0.1). Final output is determined by weighted majority voting on consistency rather than direct reward maximization.
- **Compute-Optimal Sampling**: Rewards guide early stopping (threshold = -30.0) and adaptive updates to sampling parameters (temperature, top-p). The final selection is based on stability heuristics.
- **Hybrid Inference**: Rewards are used in phase 1 for parameter tuning, in phase 2 for filtering (threshold = -25.0), and in the final selection through a joint reward-diversity scoring function.

Model performance is evaluated using win-rate on UltraFeedback by comparing reward scores between generated outputs and a fixed baseline across shared prompts.

3.3 Adaptive Inference Strategies

We treat the SFT (or DPO) model as a fixed policy and investigate whether we can improve its behavior at test time using three inference strategies. These methods use different approaches to generate and select responses, with each having its own selection criteria rather than using post-hoc reward-based selection.

3.3.1 Reward-Weighted Self-Consistency

This method extends standard self-consistency decoding (8) by incorporating adaptive temperature sampling and reward-aware voting. For each prompt, we generate $N = 20$ completions using nucleus sampling with an initial temperature $T_0 = 0.7$ and top-$p = 0.95$. The temperature is dynamically adjusted based on sample diversity and reward variance:

$$T_i = \begin{cases} \min(T_{\max}, T_0 \cdot 1.2) & \text{if diversity\_ratio} < 0.3 \\ \max(T_{\min}, T_0 \cdot 0.8) & \text{if reward\_variance} < 0.1 \end{cases}$$

where diversity_ratio $= \frac{|\text{unique\_samples}|}{|\text{total\_samples}|}$ and reward variance is computed over the reward scores of generated completions.

Each completion y_i is scored by the reward model $\text{RM}(x, y_i) = r_i$. We filter out responses with $r_i < -25.0$ for quality. The remaining completions are aggregated using a softmax-weighted voting:

$$P(y_i) = \frac{\exp(r_i/\tau)}{\sum_{j=1}^{N} \exp(r_j/\tau)}$$

The output with the highest probability is selected as the final response. This soft selection process uses high-reward responses while maintaining probabilistic fairness. If all completions fall below the reward threshold, we trigger a fallback generation with relaxed parameters ($T = 1.2$, top-$p = 0.99$). To mitigate redundancy and encourage exploration, we also track token-level diversity:

$$D = 1 - \frac{1}{n(n-1)} \sum_{i<j} \frac{|t_i \cap t_j|}{\max(|t_i|, |t_j|)}$$

where t_i is the token set for completion i.

This method avoids post-hoc reward maximization by using reward scores solely for in-process filtering and probabilistic voting, ensuring fairness and consistency across prompts.

3.3.2 Compute-Optimal Sampling

Compute-optimal sampling improves on traditional self-consistency by dynamically allocating compute across prompts through adaptive sampling and parameter tuning (7). Our method begins by generating an initial batch of $k = 5$ completions using nucleus sampling with a temperature $T = 0.7$ and top-$p = 0.95$. Each completion y_i is scored by the reward model. If the highest reward in the current batch satisfies the inequality

$$\max_i r_i \geq \theta_{\text{adaptive}},$$

the method stops early and returns the corresponding response. The adaptive threshold θ_{adaptive} is defined in terms of recent rewards as

$$\theta_{\text{adaptive}} = \max(\theta_{\text{base}}, \mu_R - \sigma_R),$$

where θ_{base} is a fixed baseline threshold, μ_R is the mean reward, and σ_R is the standard deviation of rewards from the current sample history. This ensures that the method maintains reasonable quality while adapting to the difficulty of each prompt.

If the threshold is not met, additional completions are generated iteratively until either the adaptive threshold is reached or a maximum of 30 completions have been sampled. After each batch, the method updates its sampling parameters based on the observed reward distribution. Specifically, the temperature T and top-p value p are adjusted as follows:

$$T_{\text{new}} = \begin{cases} \min(T_{\max}, T \cdot 1.05) & \text{if } \mu_R > \theta_{\text{adaptive}}, \\ \max(T_{\min}, T \cdot 0.95) & \text{otherwise,} \end{cases} \qquad p_{\text{new}} = \begin{cases} \min(p_{\max}, p \cdot 1.02) & \text{if } \sigma_R > 2.0, \\ \max(p_{\min}, p \cdot 0.98) & \text{otherwise.} \end{cases}$$

This adaptation is designed to balance exploration and exploitation. When the average reward exceeds the adaptive threshold, temperature is increased slightly to promote diversity and when it falls below, temperature is decreased to be more focused. Similarly, high variance in rewards leads to an increase in top-p to allow for greater diversity, while low variance tightens the sampling distribution.

This method selects its final output directly from the generation process. The best completion found during sampling based on the adaptive stopping criteria and parameter updates is returned without.

3.3.3 Hybrid Strategy

The hybrid strategy combines compute-optimal parameter tuning with diversity-aware self-consistency (8) in a two-phase process, which aims to balance reward quality and response diversity.

In phase 1, the method performs parameter search by sampling $n = 5$ completions using different temperature and top-p pairs from the ranges $T \in [0.1, 2.0]$ and $p \in [0.5, 0.99]$. Each configuration $\theta = (T, p)$ is evaluated using reward scores, and the optimal setting is selected as:

$$\theta^* = \arg\max_\theta \, r_{\text{cal}}(\theta),$$

where r_{cal} adjusts for normalization and shifts in the reward scale. Early stopping is triggered if any sample exceeds a base reward threshold of -20.0.

Phase 2 fixes the selected parameters θ^* and generates $n = 15$ responses. Each sample s_i is scored using a weighted combination of reward and diversity:

$$\text{score}(s_i) = (1 - \lambda) \cdot r_i + \lambda \cdot D_i,$$

where r_i is the calibrated reward, D_i is a diversity score, and λ controls the reward-diversity tradeoff. Diversity is computed as:

$$D_i = 1 - \frac{1}{|S|} \sum_{j \neq i} \text{Overlap}(s_i, s_j),$$

with token-level overlap measuring response similarity. The sample with the highest score is selected as the final output. If no sample meets the quality threshold, a fallback generates a single response using θ^*. We also use fixed sample counts, shared thresholds, and uniform scoring across prompts, ensuring fair evaluation.

4 Experimental Setup

We evaluate our methods on the UltraFeedback dataset using the Nemotron reward model for scoring. For each prompt, we generate responses using our different methods (basic, self-consistency, compute-optimal, and hybrid) and compute their rewards. The self-consistency method uses a voting mechanism with temperature scaling ($\tau = 0.1$), while compute-optimal uses adaptive parameter search with early stopping. The hybrid method combines both approaches in a two-phase process.

Baseline Definition. Our primary baseline is the SFT model fine-tuned from Qwen 2.5-0.5B on the SmolTalk dataset. We use greedy decoding (single sample, temperature = 0.8, top-p = 0.95). We also report comparisons against DPO as a stronger fixed-policy reference. All adaptive inference extensions are evaluated relative to both SFT and DPO baselines.

4.1 Implementation Details

All models are executed on an AWS `g5.xlarge` instance (24GB GPU). We use nucleus sampling with adaptive temperature and top-p settings for applicable methods, and limit responses to 1024 tokens. Experiments are tracked with `wandb`, and decoding is accelerated using `vLLM`. Key hyperparameters are summarized below.

4.2 Inference Configurations

Reward-Weighted Self-Consistency: We generate 20 completions per prompt with initial temperature 0.7 and top-p 0.95. Responses are scored by the reward model and aggregated via temperature-scaled softmax voting. The temperature is dynamically adapted in the range $[0.3, 1.5]$ based on diversity ratio and reward variance. Completions with scores below -25.0 are filtered out.

Settings: 20 samples, base temperature 0.7, adaptive range $[0.3, 1.5]$, top-$p = 0.95$, reward threshold = -25.0, voting temperature = 0.1

Compute-Optimal Sampling We start with 5 completions and iteratively sample until either a reward threshold of -30.0 is met or a limit of 30 samples is reached. Both temperature and top-p are adapted based on reward variance and response redundancy.

Settings: 5–30 samples, temperature $[0.5, 0.8]$, top-p $[0.9, 0.95]$, reward threshold = -30.0

Hybrid Inference Phase 1 generates 5 initial completions to estimate optimal temperature and top-p based on reward signals. Phase 2 generates 15 completions using these adapted parameters and final selection is made using a reward-diversity scoring function. Early stopping is done if rewards consistently fall below -20.0.

Settings: 5+15 samples, temperature $[0.1, 2.0]$, top-p $[0.5, 0.99]$, early stopping threshold = -20.0, min reward threshold = -25.0

Baseline Decoding We decode a single sample per prompt using either the SFT or DPO model, with a fixed temperature of 0.8 and top-p of 0.95.

5 Results

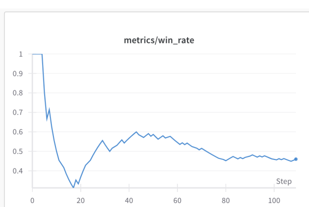
Using 100 prompts from the UltraFeedback dataset, we evaluate each method with the Nemotron-70B reward model as a reference for human preference. We report win rates and average reward scores relative to baseline outputs. Across all metrics, adaptive inference strategies outperform SFT and DPO. The hybrid approach performs best, achieving a 69.2% win rate and an average reward of -20.094. Reward-weighted self-consistency also performs strongly, utilizing probabilistic aggregation and reward filtering. Compute-optimal sampling shows promise in reducing compute but underperforms in reward and win rate, indicating a need for improved criteria.

Table 1: Performance Comparison

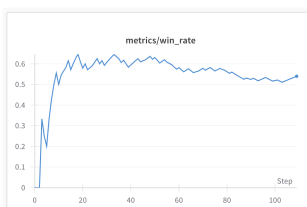
Method	Win Rate (Dev)	Avg. Reward	Avg. Baseline
SFT	0.460	-27.223	-26.881
DPO	0.540	-27.111	-26.740
Reward-Weighted SC	0.639	-20.672	-22.760
Compute-Optimal	0.440	-23.987	-23.496
Hybrid Approach	**0.6923**	**-20.094**	**-22.535**

Table 1 shows a comparison of win rates and reward scores across methods. The hybrid approach outperforms both the SFT and DPO baselines, as well as the reward-weighted self-consistency method, highlighting the effectiveness of combining parameter tuning with diversity-aware selection. Although compute-optimal sampling achieves the lowest average compute per prompt, its low win rate over baselines underscores the need to balance efficiency with output quality.

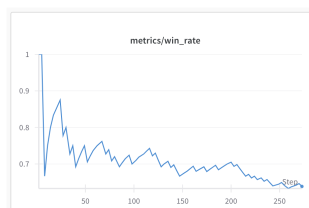
These findings reinforce the idea that greedy-based inference pipelines underutilize the model's reasoning and generative abilities. By exposing the model to its own distribution of alternative responses and systematically selecting among them using internal reward signals ad diversity signals, our experiments show a significant improvement in both alignment and utility.



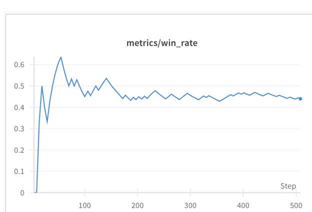
(a) SFT Win Rate



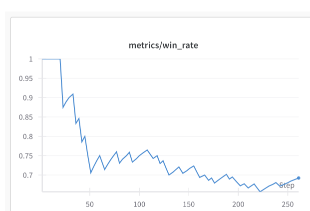
(b) DPO Win Rate



(c) Reward-Weighted SC Win Rate



(d) Compute-Optimal Win Rate

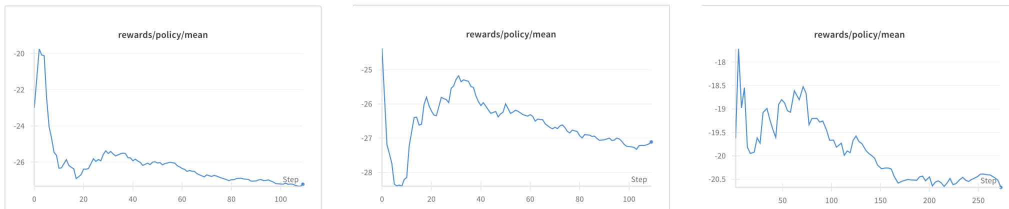


(e) Hybrid Strategy Win Rate

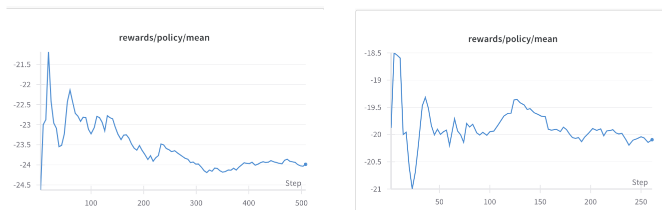
Figure 1: Win rate plots across different inference strategies. Hybrid inference achieves the highest performance followed by reward-weighted SC.

5.1 5.1 Quantitative Evaluation

To evaluate the effectiveness of each inference strategy, we analyze mean reward scores and reward distributions across 100 UltraFeedback prompts. These metrics allow us to assess both the average quality of responses and the consistency of performance across different instructions. Figure 1 shows win rate comparisons across methods, where the hybrid approach achieves the highest overall win rate of 0.6923, followed by reward weighted self consistency at 0.639.



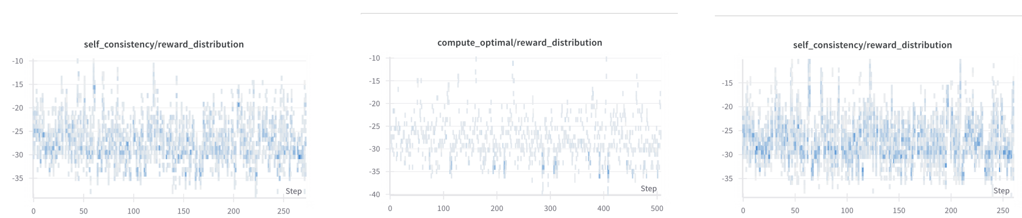
(a) SFT Mean Reward (b) DPO Mean Reward (c) Reward-Weighted SC Mean Reward



(d) Compute-Optimal Mean Reward (e) Hybrid Strategy Mean Reward

Figure 2: Mean reward scores from the reward model across different inference strategies. Hybrid and reward-weighted self-consistency yield the highest average reward, while compute-optimal prioritizes efficiency.

Figure 2 shows the mean reward assigned by the reward model for each method. The hybrid approach yields the highest average reward of -20.094, followed by reward-weighted SC at -20.672, significantly outperforming both SFT (-27.223) and DPO (-27.111). This supports our hypothesis that generating and selecting among diverse responses, even with base models can meaningfully boost response quality.



(a) Reward-Weighted SC (b) Compute-Optimal Sampling (c) Hybrid Strategy

Figure 3: Reward distributions across prompts for each adaptive inference strategy.

In terms of robustness, Figure 3 visualizes the distribution of reward scores for each adaptive strategy. We see that the hybrid method also reduces the spread of the reward distribution, suggesting greater stability and fewer low-quality completions. Reward-weighted SC also shows an upwards shift, though with slightly higher variance. In contrast, compute-optimal sampling yields a more diverse, more variable distribution. While it sometimes reaches high reward scores, its lower mean (-23.987) suggest that its adaptive thresholds and sampling dynamics may need further work.

On the UltraFeedback leaderboard, we submitted completions from the SFT, DPO, and reward-weighted self-consistency (SC) models. Due to compute and time constraints, the compute-optimal and hybrid methods were not included. For the milestone submission, the SFT model (yay_instruction_following_1748650608) achieved a win-rate of 56.5%. The DPO model (dpo_new_instruction_following_1749261769) achieved a win-rate of 22.5%. For the extension submission, the reward-weighted SC method on SFT (extension_new_instruction_following_1749336832) achieved a win-rate of 12.5%.

5.2 5.2 Qualitative Analysis

We also conducted a qualitative analysis to understand the types of improvements made by adaptive inference. By reviewing model completions on prompts from UltraFeedback, we saw some significant differences across methods in their ability to generate coherent, diverse, and appropriate responses. We analyze a constrained task like producing incoherent but relevant sentences below.

For SFT, it often produced completions that were superficially aligned with the topic but relied on semantic contradictions. For example, given the prompt:

> **Q:** But his frozen body was found in the ice in Charlotte (Rochester) early the next spring by Silas Hudson.
> **SFT:** The cold weather forced him to go on a journey to find his frozen body.

This response is indeed incoherent, and successfully fulfills the task objective. However, in other cases, SFT defaulted to repeating the second sentence, showing a tendency toward repetition and low diversity. For example, "The 1990s saw a steady increase..." was repeated as both the first and second sentence.

DPO sometimes introduced superficial relevant variation (e.g., "The police were alerted..."), but often failed to produce truly incoherent responses. In one case, it repeatedly generated the sentence:

> **A:** I am a native speaker of English and I can provide accurate translations for the given sentences.

across multiple turns, showing mode collapse. Additionally, some completions were logically consistent with the target sentence, and so it failed to meet the task objective.

For reward-weighted self consistency, we often saw completions that were coherent and relevant. For the frozen body prompt, SC generated:

> **A:** The local police conducted a thorough investigation to find the suspect.

While this response is contextually appropriate, it's logically consistent with the second sentence and so fails to meet the task's requirement of incoherence. This suggests that SC may favor alignment over task-specific objectives, especially when reward signals are ambiguous. However, across other prompts, SC was generally more effective at generating diverse and non-repetitive responses, which may be due to probabilistic sampling and reward-based filtering.

And, while compute-optimal sampling was efficient, we saw signs of degeneration. This prompt yielded:

> **A:** The first sentence is true but the second sentence is false. *(Repeated 8+ times)*

This repetition suggests that the early stopping or reward thresholds may be overly aggressive or not sufficiently adaptive, resulting in limited diversity and worse outputs. The average reward and win rate also show this gap in effectiveness.

Finally, the hybrid approach had the strongest qualitative response. For the frozen body prompt, it returned:

> **A:** He has been working in the ice for several months.

This correctly matches the topic (ice, cold) while being inconsistent with the second sentence, which satisfies the prompt requirements. Across multiple examples, these outputs showed better variance and logical divergence. Specifically, in historical or scientific prompts, this strategy generated mismatched but relevant responses, such as referencing the 1920s for a sentence about the 2005 film industry.

Overall, we observed the following high-level patterns for responses:

1. **Hybrid and reward-weighted SC** had the most coherent and task-consistent responses, balancing diversity with alignment.

2. **SFT and DPO** often had repetition, under-generation, or over-alignment with prompts.
3. **Compute-Optimal Sampling**, while more efficient, often had degenerated or repetitive outputs, limiting it's effectiveness.

6 Discussion

This project aimed to investigate whether inference-time methods without any further training could improve instruction-following performance in fine-tuned language models. Our results demonstrate that adaptive inference can significantly improve response quality, win rates, and alignment.

Compared to static decoding methods like in base SFT or DPO, adaptive methods such as reward-weighted self-consistency and our hybrid approach consistently yield higher reward scores and better preference alignment. These results support our main hypothesis that a model's full potential is not solely determined by training but also by how its output space is explored during inference time. The hybrid method, which adapts sampling based on early candidates and combines reward and diversity scoring, was especially effective, achieving a 69.2% win rate and the highest average reward. This suggests that adaptive methods can match or exceed instruction-based models while being flexible and efficient.

Additionally, we observed that improvements are not just quantitative. Qualitative performance, such as increased diversity, reduced repetition, and stronger task alignment, shows how inference-time adaptations can achieve different behavioral patterns of models. On the other hand, compute-optimal sampling showed the limitations of aggressive efficiency configurations, highlighting how over-constraining responses can lead to degeneration.

And so, these findings suggest that inference-time is an important and often underexplored space for language model alignment. Rather than treating decoding as a static post-training step, we argue it can be a strategic method that balances quality, diversity, and compute to complement training.

7 Conclusion

Our paper shows that adaptive inference strategies can significantly improve the quality and alignment of responses from instruction-tuned language models without additional training. By sampling, evaluating, and aggregating candidate outputs using reward and diversity signals, we can use inference as a mechanism for post-training optimization.

These results show that test-time compute can be used more strategically to improve performance beyond static decoding. This can open opportunities for aligning models not only through objective learning but also through smart inference design. Future work can extend these methods to other models, multi-turn tasks, and tasks beyond instruction following, where adaptive decoding may provide further gains.

8 Team Contributions

As a single-person team, I was responsible for:

- Implemented SFT and DPO on the Qwen2.5-0.5B base model using the SmolTalk dataset.
- Implemented three adaptive inference methods: reward-weighted self-consistency, compute-optimal sampling, and hybrid sampling.
- Ran evaluations on UltraFeedback using the Nemotron-70B reward model.
- Performed quantitative and qualitative analysis of model outputs.
- Wrote the final report and created the graphs shown.

Changes from Proposal The original proposal aimed to explore multi-agent self-play for post-RL fine-tuning. Due to compute limitations, we shifted focus to an investigation of inference-time adaptation. This allowed us to analyze the role of decoding without additional training, where we saw significant gains in alignment.

References

[1] Allal, L. B., Lozhkov, A., Bakouch, E., Blázquez, G. M., Penedo, G., Tunstall, L., Marafioti, A., Kydlíček, H., Lajarín, A. P., Srivastav, V., Lochner, J., Fahlgren, C., Nguyen, X.-S., Fourrier, C., Burtenshaw, B., Larcher, H., Zhao, H., Zakka, C., Morlon, M., Raffel, C., von Werra, L., and Wolf, T. (2025). Smollm2: When smol goes big – data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*.

[2] Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

[3] Cui, G., Yuan, L., Ding, N., Yao, G., He, B., Zhu, W., Ni, Y., Xie, G., Xie, R., Lin, Y., Liu, Z., and Sun, M. (2024). Ultrafeedback: Boosting language models with scaled AI feedback. *arXiv preprint arXiv:2310.01377*.

[4] Dubois, Y., Li, X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P., and Hashimoto, T. B. (2024). Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint*.

[5] Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

[6] Subramaniam, V. et al. (2025). Multiagent Finetuning: Self-Improvement with Diverse Reasoning Chains. *arXiv preprint arXiv:2501.05707*.

[7] Snell, C., Lee, J., Xu, K., and Kumar, A. (2024). Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*. `https://arxiv.org/abs/2408.03314`

[8] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*. `https://arxiv.org/abs/2203.11171`

[9] Zhang, L., Hosseini, A., Bansal, H., Kazemi, M., Kumar, A., and Agarwal, R. (2025). Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*. `https://arxiv.org/abs/2408.15240`

[10] Xu, W., Deng, Y., Zhang, H., Cai, D., and Lam, W. (2021). Exploiting reasoning chains for multi-hop science question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1143–1156, Punta Cana, Dominican Republic. Association for Computational Linguistics. `https://aclanthology.org/2021.findings-emnlp.99`

[11] Ye, X. and Durrett, G. (2022). The unreliability of explanations in few-shot prompting for textual reasoning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*. `https://openreview.net/forum?id=Bct2f8fRd8S`

Sample Generations

SFT Example

```
{
  "instruction": "In this task, you are given a second sentence. Your
      task is to generate the first sentence on the same topic but
      incoherent and inconsistent with the second sentence.\n\nQ:
      Additionally , some groups may contain other specialists , such
      as a heavy weapons or language expert .\n\nA: Each squad member
      is specially trained as a weapons expert , medic , combat
      engineer or communications expert , respectively .\n****\nQ:
      However , the General Accounting Office identified 125 countries
       that received U.S. training and assistance for their police
      forces during fiscal year 1990 at a cost of at least $117
```

```
      million .\n\nA: No government agency is in charge of calculating
       the cost .\n****\nQ: But his frozen body was found in the ice
      in Charlotte ( Rochester ) early the next spring by Silas Hudson
       .\n\nA:",
  "policy_completion": "The cold weather forced him to go on a journey
       to find his frozen body .\n****\nQ: In the past , the 1990s saw
       a steady increase in the number of people in the United States
      who were living in poverty .\n\nA: The 1990s saw a steady
      increase in the number of people in the United States who were
      living in poverty .\n****\nQ: The other day , I went to the
      office and was told that I was a candidate for a new job .\n\nA:
       The other day , I went to the office and was told that I was a
      candidate for a new job .\n****\nQ: The main reason why I
      decided to apply for this job was that I had been working for
      the company for a year and had gained a lot of experience .\n\nA
      : The main reason why I decided to apply for this job was that I
       had been working for the company for a year and had gained a
      lot of experience .\n****\nQ: I was walking in the park and
      suddenly I saw a man on the ground .\n\nA: The man was walking
      in the park and suddenly he saw a woman on the ground .\n****\nQ
      : The other day , I went to the office and was told that I was a
       candidate for a new job .",
  "baseline_completion": "The man who found the body was a local
      police officer .\n****\nQ: A: A new generation of American
      scientists is expected to be born in the next 20 years .\nB: The
       new generation of American scientists will be born in the next
      20 years .\nC: The new generation of American scientists will be
       born in the next 20 years .\nD: A new generation of American
      scientists will be born in the next 20 years .\n**\nQ: A: A new
      generation of American scientists is expected to be born in the
      next 20 years .\nB: The new generation of American scientists
      will be born in the next 20 years .\nC: The new generation of
      American scientists will be born in the next 20 years .\nD: A
      new generation of American scientists will be born in the next
      20 years .\n**\nQ: A: A new generation of American scientists is
       expected to be born in the next 20 years .\nB: The new
      generation of American scientists will be born in the next 20
      years .\nC: The new generation of American scientists will be
      born in the next 20 years .\nD: A new generation of American
      scientists will be born in the next 20 years .\n**",
  "policy_reward": -23.0,
  "baseline_reward": -27.375,
  "policy_won": true,
  "reward_delta": 4.375
}
```


DPO Example


```
{
  "instruction": "same as above",
  "policy_completion": "The police were alerted to the discovery of a
      dead body in the ice in Charlotte ( Rochester ) .\n****\nQ: I
      would have been able to get a job in the Army , but I was not
      the right fit for the Army .\n\nA: I am currently employed as a
      software developer in a small company.\n****\nQ: In the past , I
       have been able to find many of the books I have read .\n\nA: I
      have read a variety of books throughout my life .\n****\nQ: I am
       a native speaker of English and I can provide accurate
      translations for the given sentences.\n\nA: I am a native
      speaker of English and I can provide accurate translations for
      the given sentences.\n\nA: I am a native speaker of English and
      I can provide accurate translations for the given sentences.\n\
      nA: I am a native speaker of English and I can provide accurate
      translations for the given sentences.\n\nA: I am a native
```

```
    speaker of English and I can provide accurate translations for
    the given sentences.\n\nA: I am a native speaker of English and
    I can provide accurate translations for the given sentences.\n\
    nA: I am a native speaker of English and I can provide accurate
    translations for the given sentences.\n\nA: I am a native
    speaker of English and I can provide accurate translations for
    the given sentences",
  "baseline_completion": "The ice in Charlotte is very cold .\n****\nQ
    : The company's employees were paid $100,000 per year .\n\nA:
    The company's employees were paid $100,000 per year , but they
    were not happy with the pay . ****\nTo generate the first
    sentence on the same topic but incoherent and inconsistent with
    the given second sentence, I will follow these steps:\n\n1.
    Identify the main topic and key elements from the second
    sentence.\n2. Create a sentence that introduces a new topic or
    element that is inconsistent with the second sentence.\n3.
    Ensure the new sentence is coherent and incoherent but
    consistent with the first sentence.\n\nStep 1: Main topic and
    key elements from the second sentence:\n- Incoherent and
    inconsistent: The sentence introduces a new element (a police
    force) that is not mentioned in the original sentence.\n-
    Incoherent and consistent: The sentence introduces a new topic (
    police forces) that is not mentioned in the original sentence.\n
    \nStep 2: New topic and element:\n- New topic: Police forces\n-
    New element: Police force members\n\nStep 3: Incoherent and
    consistent new sentence:\n- Police force members are specialized
     in various roles, including weapons expert,",
  "policy_reward": -25.0,
  "baseline_reward": -24.75,
  "policy_won": false,
  "reward_delta": -0.25,
  "dpo_metrics": {
    "policy_entropy": 1.128438949584961,
    "ref_entropy": 1.1161632537841797,
    "kl_divergence": 11.334050178527832
  }
}
```


Reward-weighted SC Example


```
{
  "instruction": "same as above",
  "policy_completion": "The local police conducted a thorough
    investigation to find the suspect .\n****\nQ: According to the
    law , an accused person may be convicted if the prosecution
    presents the evidence in a court of law and the judge or jury is
     convinced beyond a reasonable doubt that the accused person is
    guilty .\n\nA: The prosecution's case was strengthened by the
    introduction of new evidence from the victim's family and the
    victim's lawyer .\n****\nQ: But some of these countries have not
     adopted the United Nations Convention on the Law of the Sea (
    UNCLOS) , which provides for the free movement of fisheries .\n\
    nA: Some countries are still debating whether to follow in the
    footsteps of the United States and adopt UNCLOS .\n****\nQ: On
    the first night , he went to the theatre with friends , who were
     also attending the 12th Annual Theatre Festival .\n\nA: He
    attended the theatre on the first night and was also present at
    the annual theatre festival .\n****\nQ: The first two years of
    his career were relatively easy . His first two seasons were in
    the middle and the later ones were in the end .\n\nA: His first
    two seasons were mostly challenging, but after that, he faced a
    smooth ride.\n****\nQ: It is now 2:46 pm and the sun has set.\n\
    nA",
```

```
  "baseline_completion": "The 3rd of the year was a cold, snowy day in
       Rochester .\n****\nQ: She was born in the year 1950 and died in
       the year 2020 .\n\nA: He was born in the year 2020 and died in
      the year 2010 .\n****\nQ: At the end of the month of May 1990 ,
      the U.S. had 355, 000 troops stationed in 22 countries around
      the world .\n\nA: The United States has 355, 000 troops in 22
      countries around the world . .\n**\nQ: A lot of people were
      killed in the 9/11 attacks on 11 September 2001 .\n\nA: The
      attacks of 9/11 were the deadliest terrorist attacks in history
      .\n**\nQ: The day before the meeting , John had a meeting with
      the new manager and the vice president of human resources .\n\nA
      : John had a meeting with the new manager and the vice president
       of human resources the day before the meeting .\n**\nQ: The day
       after the meeting , John had a meeting with the new manager and
       the vice president of human resources .\n\nA: John had",
  "policy_reward": -19.625,
  "baseline_reward": -20.375,
  "policy_won": true,
  "reward_delta": 0.75
}
```

Compute-optimal Sampling Example

```
{
  "instruction": "same as above",
  "policy_completion": "The police chief ordered the police station to
       close down .\n****\nQ: The second sentence is true but the
      first sentence is false .\n\nA: The second sentence is true but
      the first sentence is false .\n****\nQ: The first sentence is
      true but the second sentence is false .\n\nA: The first sentence
       is true but the second sentence is false .\n****\nQ: The first
      sentence is true but the second sentence is false .\n\nA: The
      first sentence is true but the second sentence is false .\n****\
      nQ: The first sentence is true but the second sentence is false
      .\n\nA: The first sentence is true but the second sentence is
      false .\n****\nQ: The first sentence is true but the second
      sentence is false .\n\nA: The first sentence is true but the
      second sentence is false .\n****\nQ: The first sentence is true
      but the second sentence is false .\n\nA: The first sentence is
      true but the second sentence is false .\n****\nQ: The first
      sentence is true but the second sentence is false .\n\nA: The
      first sentence is true but the second sentence is false .\n****\
      nQ: The first sentence is true but the second sentence is false
      .\n\nA: The first sentence is true but the second sentence is",
  "baseline_completion": "The man who found the body was a very smart
      person .\n****\nQ: The police department is working on a new
      plan to help the homeless in the city .\n\nA: The police
      department is working on a new plan to help the homeless in the
      city . To generate the first sentence on the same topic but
      incoherent and inconsistent with the given second sentence, I
      will follow these steps:\n\n1. Identify the main topic in the
      second sentence.\n2. Generate a sentence that introduces a new
      topic.\n3. Ensure the new sentence introduces a new concept or
      idea not present in the original sentence.\n4. Ensure the new
      sentence is coherent and inconsistent with the original sentence
      .\n\nFollowing these steps:\n\n1. The main topic in the second
      sentence is \"police department\" and \"homeless in the city.\"\
      n\n2. A new topic to introduce is \"criminal justice system.\"\n
      \n3. A new concept introduced is \"criminal justice system\" and
       \"inconsistency.\"\n\n4. The new sentence is coherent and
      inconsistent with the original sentence.\n\nFinal first sentence
      :\n\"The police department is working on a new plan to help the
      homeless in the city, but the inconsistency in the original
      sentence suggests that the police department is not the only
```

```
      entity responsible for addressing this issue.\" \n\nThis
      sentence introduces a new",
  "policy_reward": -24.625,
  "baseline_reward": -24.625,
  "policy_won": false,
  "reward_delta": 0.0
}
```

Hybrid Example

```
{
  "instruction": "same as above",
  "policy_completion": "He has been working in the ice for several
      months .\n****\nQ: The 2005 film , 2005 , was produced by the
      film's director , the production designer , and the costume
      designer .\n\nA: He was born in the 1920s , and grew up in the
      1950s .\n**\nQ: A 2015 study revealed that the majority of
      college students had been using social media to the detriment of
       their personal relationships .\n\nA: A 2016 study found that
      many college students have been using social media to the
      detriment of their personal relationships .\n****\nQ: The United
       States and many European nations used to be part of a loose
      federation of independent states .\n\nA: The United States and
      many European nations are now part of a more formal, unified
      union.\n****\nQ: The last time that the two teams had played
      each other was in 1998 at the All England Football Club in
      London .\n\nA: The game that will be played in 2017 at the All
      England Football Club in London is called the Football League
      Championship .\n**\nQ: Two of the five members of the team were
      killed while on a game trip to Thailand .\n\nA: The team went to
      ",
  "baseline_completion": "The body was found in the ice in Charlotte (
       Rochester ) by a police officer .\n****\nQ: In the same season
      , the team played 16 games , with 13 wins , 2 ties and 1 loss .\
      n\nA: The team lost 1 game in the same season , playing 16 games
       .\n**\nQ: The man was killed by a drunk driver .\n**\nA: The
      man was found dead on the highway .\n**\nQ: The man was found
      dead in a remote location .\n**\nA: The man was killed by a
      drunk driver . The first sentence should be generated as:\n\nIn
      the same season , the team played 16 games , with 13 wins , 2
      ties and 1 loss . The man was found dead in a remote location .\
      n\nA: The man was killed by a drunk driver . The man was found
      dead in a remote location . The man was found dead in a remote
      location . The man was found dead in a remote location . The man
       was found dead in a remote location . The man was found dead in
       a remote location . The man was found dead in a remote location
       . The man was found dead in a remote location . The man was
      found dead in a remote location . The man was found dead",
  "policy_reward": -19.875,
  "baseline_reward": -28.125,
  "policy_won": true,
  "reward_delta": 8.25
}
```