

Learning Diverse Responses with Prefix-Conditioned Supervised Fine-Tuning

Zhiyuan Fan¹ Guanqiao Chen² Yanyi Huang¹
Mingkuan Zhao¹ Dadi Guo¹ Yi R. Fung^{1*}

¹The Hong Kong University of Science and Technology

²University of Science and Technology of China

zhiyuan.fan@connect.ust.hk yrfung@ust.hk

Abstract

Large language models (LLMs) have shown strong performance on hard reasoning and general instruction-following tasks. However, when sampling multiple outputs for the same prompt, they often produce highly homogeneous, repetitive responses, resulting in inefficient exploration. This limits the gains from test-time scaling and constrains the upper bound of reinforcement learning (RL) training. We attribute this issue in part to supervised fine-tuning (SFT): when a single prompt is paired with multiple reference responses, the model is trained to generate diverse outputs under the same prior condition, which induces optimization interference and can lead to diversity collapse. To address this, we propose Prefix-Conditioned SFT (P-SFT), a simple yet effective method that constructs semantically consistent yet distributionally distinct prior contents to different responses, thereby projecting the instruction into distinct latent regions to establish diverse prior distributions and decouple the one-to-many mapping. Experiments on large reasoning language models show that our approach improves absolute performance by 5.3% on reasoning benchmarks and increases generation diversity by 198.3% on average, while substantially enhancing output diversity and test-time scaling. Notably, even without any additional training, our prefixing strategy can be applied at inference time alone and still yields significant gains in both diversity and reasoning performance for instruction-tuned LLMs and reasoning-enhanced models.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success in following complex instructions and performing reasoning, as evidenced by their performance on benchmarks such as IF-Eval (Zhou et al., 2023), Arena Hard (Li et al., 2024), and competitive programming tasks such as

LiveCodeBench (Jain et al., 2024). Despite these advancements, a critical phenomenon of diversity collapse, or mode collapse, persists (O’Mahony et al., 2024; Kirk et al., 2024; Zhang et al., 2025a). Even when capable of correctness, LLMs often exhibit highly repetitive styles and similar solution paths, struggling to generate novel ideas in creative writing or diverse reasoning chains for mathematical problems. The lack of diversity not only limits the model’s creative utility but also severely constrains test-time scaling (Muennighoff et al., 2025b). Since the sampling distribution is overly concentrated, increasing the number of trials (e.g., Best-of- N) fails to yield significantly new or effective solutions.

Recent literature has primarily attributed this collapse to reward model bias or entropy collapse during RL training (Gao et al., 2022; Yu et al., 2025a). However, we argue that the root cause lies earlier in the fundamental mismatch between traditional Supervised Fine-Tuning (SFT) objectives and the structure of modern synthetic datasets. Current open-source datasets (e.g., OpenThoughts (Guha et al., 2025), Nemotron (Bercovich et al., 2025)) for general instruction following or specialized reasoning for math, code, and STEM are largely distilled from proprietary teacher models (Shumailov et al., 2024), often adopting a single-instruction to multiple-responses (1-to-M) format. Standard SFT expects the model to fit these diverse responses under an identical conditioning context, which induces optimization interference, where the gradient directions for different responses conflict within the same prior distribution, forcing the model to converge to a single mode and ignore the remaining diverse response patterns.

To resolve this, we propose Prefix-conditioned SFT (P-SFT), a simple yet effective approach to the instruction tuning paradigm. As illustrated in Figure 1, P-SFT decouples the learning of multiple responses by constructing semantically consis-

*Correspondence Author

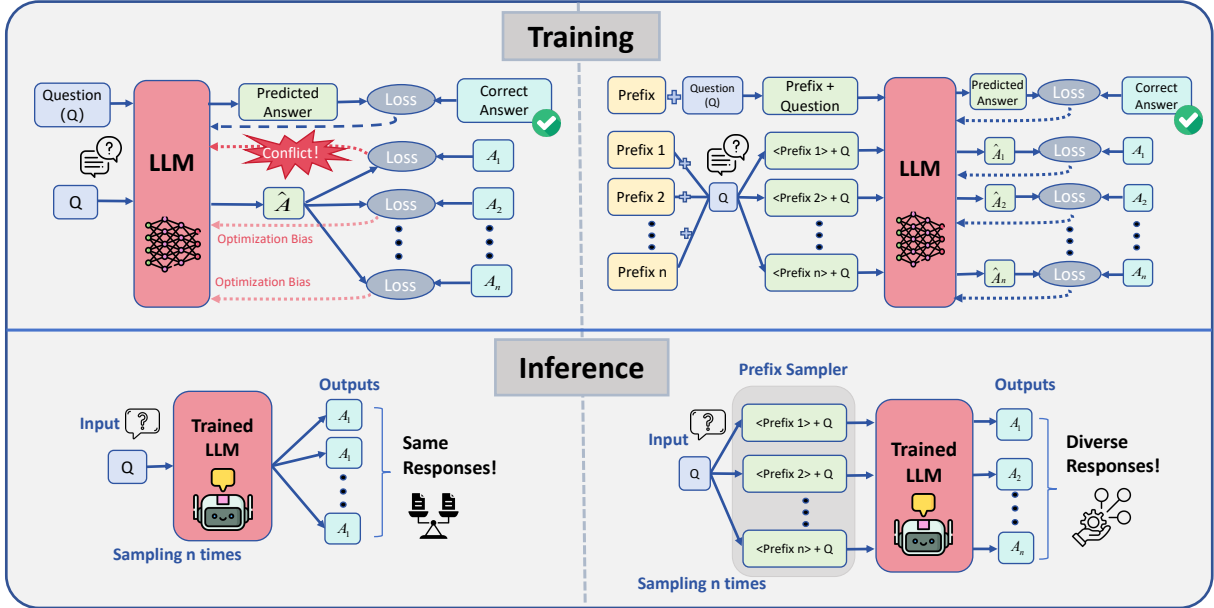


Figure 1: **Training:** Traditional SFT forces the model to generate different responses under the same prompt (i.e., the same prior condition), which creates conflicts and leads to diversity collapse. Our method decouples this one-to-many mapping by attaching different prefixes to the prompt, thereby alleviating the conflict. **Inference:** We prepend n different prefixes to the prompt to create conditions that differ while preserving the same semantics, encouraging the model to sample more diverse responses, which contrasts with conventional inference that samples n times under the same condition.

tent but distributionally distinct prior distributions. By prepending unique, semantically neutral prefixes to the instruction, we project the same semantic instruction into different regions of the high-dimensional prior latent space, which increases the orthogonality of learning signals and mitigates gradient conflicts, allowing the model to fully absorb the diversity inherent in 1-to-M datasets. Furthermore, we demonstrate that this mechanism naturally extends to inference as a novel test-time scaling strategy that perturbs the input latent space rather than output logits (e.g., temperature, top-p (Holtzman et al., 2020)), facilitating a deeper exploration of the model’s generative priors.

Our contributions are summarized as follows:

- 1) We identify a fundamental mismatch between traditional SFT and the 1-to-M format of modern distilled datasets, characterizing it as an optimization interference problem where identical prior conditions lead to gradient conflicts.
- 2) We propose P-SFT to decouple response learning via semantically-neutral prefixes, which mitigates mode collapse during training and enables a novel input-side perturbation for enhanced test-time scaling during inference.
- 3) We demonstrate that P-SFT significantly outperforms standard SFT in diversity and reasoning

benchmarks. Furthermore, P-SFT provides a superior initialization for Reinforcement Fine-Tuning (RFT) by substantially improving exploration efficiency and rollout quality.

2 Related work

Test-Time Scaling. Test-time scaling (TTS) has emerged as a pivotal paradigm for augmenting the reasoning capabilities of large language models (LLMs) through increased inference-stage computation. This paradigm is typically categorized into parallel and sequential allocation strategies (Muenighoff et al., 2025a). Parallel scaling, or Best-of- N sampling, leverages the principle that solution coverage improves monotonically with the number of generated candidates (Chollet, 2019; Brown et al., 2024). However, the effectiveness of this paradigm is fundamentally bottlenecked by the *selection gap*, where standard discriminative mechanisms such as majority voting (Wang et al., 2023b) or reward models (Lightman et al., 2023; Wu et al., 2024; Pan et al., 2025) frequently fail to isolate optimal solutions from increasingly large and noisy candidate pools (Hassid et al., 2024; Stroebel et al., 2024). Conversely, sequential scaling emphasizes deepening internal reasoning trajectories through Chain-of-Thought (CoT) (Wei et al., 2022) and it-

erative self-correction (Madaan et al., 2023; Hou et al., 2025), a strategy successfully operationalized in frontier models such as OpenAI o1, DeepSeek-R1 (Guo et al., 2025), and Qwen-QwQ (Team). While these methods have been specialized for code generation (Li et al., 2025a; Yu et al., 2025b) and multimodal reasoning (Xin et al., 2025), they often rely on heuristic compute allocation that encounters performance plateaus as inference compute scales (Ghosal et al., 2025). Crucially, these plateaus are often symptomatic of a lack of *solution diversity*, where additional computation fails to explore novel reasoning paths. This underscores the necessity of identifying *thinking-optimal* allocation curves that balance reasoning depth against search breadth and diversity, ensuring that scaled compute translates into meaningful exploration.

Diversity Collapse. During the training process, LLM outputs tend toward homogenization, converging to local optima and becoming unable to generate alternative correct solutions with different reasoning approaches (Zhang et al., 2025a). This phenomenon emerges at the supervised fine-tuning stage (O’Mahony et al., 2024; Li et al., 2025b), where model-generated responses gradually converge to a fixed style or reasoning pattern (Yun et al., 2025), reducing diversity compared to the pre-training phase. During the reinforcement learning (RL) training stage, particularly when using algorithms such as GRPO (Shao et al., 2024), output diversity further deteriorates due to entropy collapse caused by high-advantage positive samples in the RL training process (Casper et al., 2023; Kirk et al., 2024). The model increasingly exploits already-discovered correct paths rather than exploring new potential reasoning trajectories. To mitigate this issue, several methods have been proposed, including reward shaping (Wang et al., 2026), pass@k optimization (Chen et al., 2025), DAPO (Yu et al., 2025a), and Adaptive Entropy-Controlled Policy Optimization (Wan et al., 2025; Shen et al., 2025). However, these approaches primarily focus on the RL stage while overlooking diversity issues during supervised finetuning (SFT), which occurs earlier and affects RL initialization performance. The gap is particularly critical in contemporary LLM development scenarios, where the majority of instruction tuning datasets are distilled from proprietary LLMs (Lee et al., 2025), typically structured with multiple responses per question. While the literature identifies SFT as a source of

stylistic convergence (Li et al., 2025b), the intrinsic conflict between 1-to-M data distributions and identical prior conditioning remains under-explored. In this paper, we present the first systematic investigation of this problem and propose a solution to address it.

3 Approach

3.1 Preliminary

In supervised fine-tuning (SFT), an LLM is trained to follow instructions and generate responses that align with human preferences by maximizing the conditional likelihood:

$$\mathcal{L}_{\text{SFT}}(\theta) = - \sum_{(I,r) \in \mathcal{D}} \log p_{\theta}(r | I), \quad (1)$$

where I denotes the instruction and r is the corresponding response provided by human annotation or distilled from a teacher LLM.

When the dataset contains multiple responses for the same instruction (*i.e.*, 1-to-M), the objective becomes

$$\mathcal{L}_{\text{SFT}}^{1\text{-to-M}}(\theta) = - \sum_{(I, \{r^{(m)}\}_{m=1}^{M_I}) \in \mathcal{D}} \sum_{m=1}^{M_I} \log p_{\theta}(r^{(m)} | I). \quad (2)$$

In practice, the model does not learn these responses equally under the same conditioning context due to differences in data ordering and gradient magnitudes that can induce optimization interference. Specifically, the model is forced to fit multiple distinct outputs $\{r^{(m)}\}$ under the identical hidden state $h = \text{Encoder}(I)$, leading to gradient conflicts where $\nabla_{\theta} \log p_{\theta}(r^{(1)} | I) \cdot \nabla_{\theta} \log p_{\theta}(r^{(2)} | I) < 0$, see Appendix K. This phenomenon leads the model to fit a subset of responses and ignore others, which reduces response diversity and can manifest as response collapse.

3.2 Prefix-conditioned SFT

We leverage the autoregressive nature of LLMs and create distinct conditioning contexts by prepending a semantic-irrelevant prefix to the instruction. For each instruction I with responses $\{r^{(m)}\}_{m=1}^{M_I}$, we assign a unique prefix $z_{I,m}$ to each response $r^{(m)}$, in order to increase the separation between conditioning contexts, making the learning signals for different responses less coupled and reducing gradient interference. By projecting the same

instruction I into different starting hidden states $h_m = f_\theta(z_{I,m} \oplus I)$ in a high-dimensional space via prefixes, the model can seek update directions with higher orthogonality in the parameter space.

We optimize the following objective:

$$\mathcal{L}_{\text{PreSFT}}^{1\text{-to-}M}(\theta) = - \sum_{(I, \{r^{(m)}\}_{m=1}^{M_I}) \in \mathcal{D}} \sum_{m=1}^{M_I} \log p_\theta(r^{(m)} | z_{I,m} \oplus I), \quad (3)$$

where $z_{I,m}$ is designed to be semantically unrelated to I and \oplus denotes concatenation. The training objective reduces to

$$\mathcal{L}_{\text{PreSFT}}(\theta) = - \sum_{(I,r) \in \mathcal{D}} \log p_\theta(r | z_I \oplus I), \quad (4)$$

with a unique prefix z_I for each training pair.

In our implementation, we instantiate the prefix using the template “random state: { } \n\n”, where the placeholder { } is filled with a randomly sampled integer in $[0, 99999]$ (zero-padded to a fixed width). To prevent the model from establishing a rigid dependency or *hard-indexing* between specific random IDs and semantic content, we re-sample the integer for each prefix at every training iteration:

$$z_{I,m}^{(t)} \sim \mathcal{Z}, \quad \forall (I, r^{(m)}) \in \mathcal{D}, \quad (5)$$

where t denotes the iteration index and \mathcal{Z} represents the discrete uniform space of prefixes. This ensures that the model treats the prefix as a signal for context separation rather than a static identifier, thereby maintaining its generalization ability. To prove that prefixing creates distinct conditioning contexts while preserving the semantics needed for correct instruction following, we provide a theoretical analysis in Appendix C.

3.3 Prefix-conditioned Inference

Since the prefix can directly modify the prior distribution, P-SFT can be applied at inference time for test-time scaling without updating any parameters. Unlike standard inference which draws n samples under a single conditioning context, we approximate the posterior response distribution by marginalizing over the prefix space \mathcal{Z} :

$$p_{\text{scaled}}(r | I) \approx \frac{1}{N} \sum_{n=1}^N p_\theta(r | z_n \oplus I), \quad z_n \sim \mathcal{Z}. \quad (6)$$

While traditional decoding (e.g., Top- p sampling) introduces randomness at the output layer $p_\theta(y_t | y_{<t}, I)$, prefix-conditioning perturbs the hidden representations at the input level. This facilitates latent-side exploration, encouraging the model to traverse diverse reasoning trajectories that are semantically consistent yet structurally varied. Increasing the diversity of the input-side conditioning allows for a more comprehensive coverage of the full probability distribution $p(r|I)$, which can be combined with existing techniques such as Best-of- N or Self-consistency (Wang et al., 2023a) to further enhance performance. The proposed method for both training and inference is illustrated in Figure 1.

4 Experiments

4.1 Experimental Setup

Benchmarks for Diversity: We follow the DivPo (Lanchantin et al., 2025) task setup to evaluate the diversity of LLM-generated responses, focusing on three types of tasks: persona generation, keyword-based story generation, and full story generation. NoveltyBench (Zhang et al., 2025b) is also included to assess the novelty of generated content, which is directly influenced by diversity. Beyond open-ended generation, we also evaluate diversity in mathematical reasoning tasks, where we measure solution-level diversity using an LLM-as-a-judge (Zheng et al., 2023) protocol with Claude Sonnet 4.5 as the judge, using 5-sample self-consistency majority voting and calibration against 100 human-annotated instances (Pearson $r = 0.87$).

Reasoning Capability: We use HMMT (hmm, 2025) and AIME 2024 & 2025 (Mathematical Association of America, 2025) for evaluating mathematical reasoning, LiveCodeBench (Jain et al., 2024) for coding, and GPQA-Diamond (Rein et al., 2023) for general reasoning.

Model Backbone: Our experiments include instruction-tuned LLMs (Qwen2.5-7B-Instruct (Yang et al., 2025), Llama-3.1-8B-Instruct (Touvron et al., 2023)) and a reasoning-enhanced model (Qwen3-8B). For additional implementation details, please refer to the Appendix B.

Method	Persona			Keyword		Full Story		NoveltyBench
	Name	City	Job	1-Gram	Entropy	1-Gram	Entropy	Distinct
Qwen3-8B								
B-Inference	6.32	9.58	5.94	6.83	2.84	286.08	37.57	1.04
P-Inference	<u>43.37</u>	<u>56.71</u>	<u>53.84</u>	<u>10.27</u>	<u>2.97</u>	<u>1645.87</u>	<u>80.99</u>	<u>3.41</u>
Qwen2.5-7B-Instruct								
B-Inference	6.91	8.34	6.74	6.41	2.80	265.32	35.81	1.02
P-Inference	<u>38.23</u>	<u>51.96</u>	<u>42.39</u>	<u>9.54</u>	<u>2.92</u>	<u>1523.65</u>	<u>74.56</u>	<u>2.97</u>
Llama3.1-8B-Instruct								
B-Inference	30.45	13.82	27.93	6.57	2.80	264.33	35.38	1.02
P-Inference	<u>46.37</u>	<u>52.74</u>	<u>48.32</u>	<u>9.81</u>	<u>2.91</u>	<u>1521.98</u>	<u>72.31</u>	<u>2.83</u>
DivPO	43.82	42.34	47.01	8.34	2.75	1332.97	56.49	2.37

Table 1: Performance comparison on four diversity benchmarks under different training and inference methods. B-Inference denotes conventional sampling-based inference, while P-Inference denotes prefix-conditioned inference. Higher values indicate better performance for all metrics.

4.2 Prior Condition Difference

We quantitatively analyze how prefixing affects the prior conditioning context. Our evaluation focuses on two metrics: (1) the differences in the logits of the first generated token under different prefixes; and (2) the diversity of the sampled responses when different prefixes are prepended.

We consider three categories of prompts: mathematical reasoning, factual questions, and story-style descriptive prompts. We find that after adding prefixes, the next-token logits change substantially, leading to a 36.67% change in the greedy-decoding outputs. In addition, the cosine similarity between the prior conditioning contexts before and after prefixing drops to 0.558–0.736, indicating that prefixing has a notable impact on the model’s decoding behavior. From the perspective of KL divergence, the average change is 0.42, suggesting a significant shift in the logit distribution, which implies that our prefixing strategy induces distinct prior distributions. For concrete case studies, please refer to Appendix D.

Table 1 summarizes the response-diversity results. Across both instruction-tuned LLMs and reasoning-enhanced LLMs, prefix-based inference yields a substantial and consistent increase in output diversity, and notably outperforms DivPO (Lanchantin et al., 2025), a training-time method explicitly designed to promote diversity. A potential concern is that directly perturbing the next-token logit distribution, and thus altering the decoding dynamics, might inadvertently shift the semantic

conditioning of the prompt and lead to instruction misalignment. To assess this risk, we randomly sample 100 instances for human assessment and additionally apply rule-based automated checks to verify format compliance. In all inspected cases, the generated outputs strictly follow the instructions and satisfy the required formatting constraints, suggesting that prefix conditioning can meaningfully diversify the effective prior context without compromising prompt semantics.

Moreover, the diversity gains translate into improved test-time scaling for reasoning-enhanced LLMs. As shown in Figure 2, $\text{pass}@k$ on AIME 2024/2025 and HMMT improves markedly over the traditional parallel sampling method, indicating that prefix-conditioned inference provides a simple yet effective mechanism to strengthen test-time scaling performance further.

4.3 Prefix-conditioned SFT

We follow the OpenThought (Guha et al., 2025) setting and train on the same dataset, OpenThought3-1.2M, where each question is paired with 16 responses. We fine-tune Qwen2.5-7B-Instruct using both traditional SFT and our prefix-conditioned SFT; all other optimizer hyperparameters are kept identical. A relatively large learning rate is adopted, motivated by the substantial distributional mismatch between instruction-tuned and reasoning-oriented model parameters. During training, instead of fixing seeds to 0–15, we sample seeds uniformly from 0 to 99999 without replacement to avoid repeated randomness across examples. At

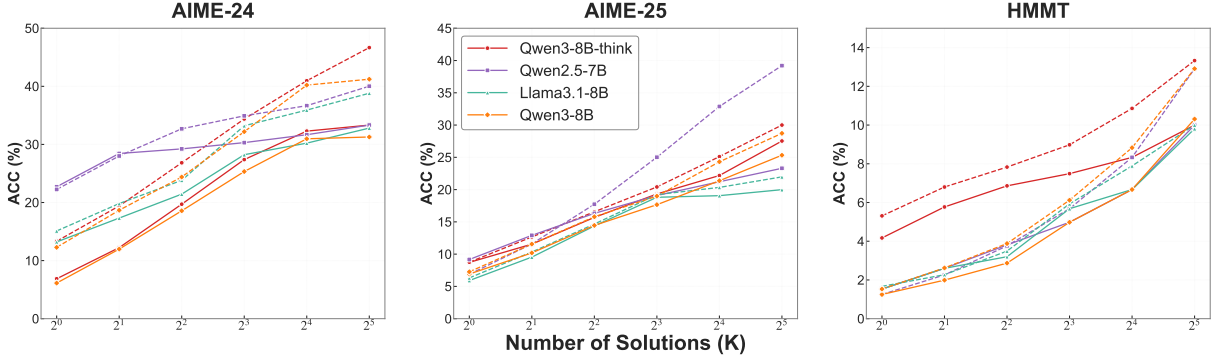


Figure 2: Performance of test-time scaling with increasing number of samples on hard reasoning benchmarks. Solid lines represent the traditional inference method, while dashed lines represent prefix-conditioned inference.

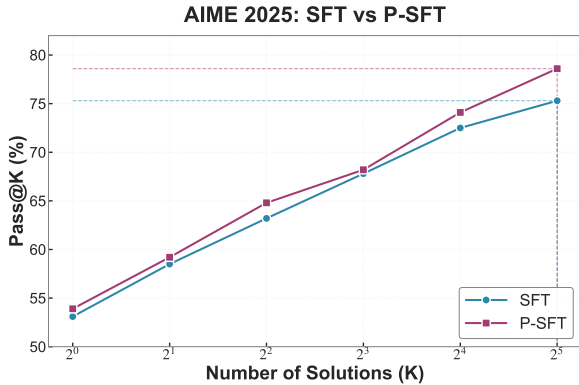


Figure 3: Performance comparison of SFT and P-SFT trained models on AIME 2025 with increasing number of samples (K). P-SFT achieves 4.38% relative improvement over SFT at $K=32$.

inference time, we evaluate two decoding protocols: (1) Standard sampling (no prefix): sample n responses under an identical prior conditioning context, where the temperature is set to 0.6, following the previous works. (2) Prefix-conditioned sampling: prepend a distinct prefix for each sample, thereby inducing different prior conditioning contexts across responses.

Table 2 reports results on hard reasoning benchmarks, and Figure 3 shows the performance along the increasing sampling number. Prefix-conditioned SFT yields a substantial improvement in $\text{pass}@k$, while maintaining $\text{pass}@1$ and in some cases even slightly improving it, suggesting that ours indeed enhanced diversity. Moreover, applying prefix-conditioned inference on top of P-SFT provides additional gain in $\text{pass}@k$. The gap between standard inference and prefix-conditioned inference on the P-SFT model suggests that training has meaningfully reshaped the model’s learned prior. Even without prefixes, the P-SFT checkpoint

Model	HMMT	AIME25	LCB	GPQA-D
Qwen2.5-7B-Instruct	2.0	8.0	16.3	24.6
DS-R1-Distill-7B	25.0	38.0	34.5	33.2
OpenR1-Distill-7B	25.7	39.7	30.7	58.9
AceReason	33.3	50.7	44.3	52.9
Nemotron-Nano	26.7	48.0	50.9	52.9
OpenThinker3-7B (SFT)	41.3	53.1	52.1	51.7
OpenThinker3-7B (P-SFT)	42.5	53.9	53.3	51.8
OpenThinker3-7B (P-SFT*)	42.5	53.9	53.4	52.1

Table 2: Performance comparison across reasoning benchmarks. SFT denotes our reproduced results using traditional SFT, while P-SFT is trained with our prefix-conditioned SFT method, where the only difference is the training method and all other settings are kept identical. P-SFT* denotes results obtained using prefix-conditioned inference.

produces more diverse outputs than the SFT baseline, which implies that decoupling the 1-to- M supervision signal helps avoid collapsing different solution modes into a single overfitted trajectory, and instead keeps multiple distinct solution paths in the weights. Further improvements appear when we enable prefix-conditioned inference. In this case, the random prefix acts like a latent switch: it steers the initial prior distribution away from the over-stable region induced by the vanilla prompt, making it easier to follow alternative reasoning routes. As shown in the scaling curves (Figure 3), standard SFT gradually hits a diversity ceiling, while P-SFT continues to improve more steadily in $\text{pass}@k$. This pattern is consistent with P-SFT turning the landscape from one dominant mode into a more multi-modal surface where different local optima are easier to reach. Each prefix provides a small but useful push in latent space that helps the model move off the main solution mode and explore other valid trajectories. As a result, inference shifts from just sampling different tokens to a more structured form of exploration through conditioning, so extra test-time

Type	KL	Persona	Keyword	Full Story	NoveltyBench	IF ACC	Format ACC
Prefix Types							
Corpus Span	0.63	<u>51.83</u>	6.21	759.32	<u>2.84</u>	0.92	<u>0.95</u>
Soft Embedding	<u>0.61</u>	47.19	<u>6.17</u>	678.43	2.01	<u>0.97</u>	0.93
Template (Ours)	0.58	44.19	6.23	799.11	2.97	1.00	1.00
Insertion Position							
After	<u>0.47</u>	<u>45.08</u>	<u>6.19</u>	<u>748.21</u>	<u>2.83</u>	<u>0.93</u>	<u>0.97</u>
Before (Ours)	0.58	44.19	6.23	799.11	2.97	1.00	1.00

Table 3: Ablation study on prefix types and insertion position. For Persona, Keyword, and Full Story tasks, we report the average results across their respective metrics. IF ACC denotes instruction following accuracy.

compute more reliably translates into broader solution coverage. We hypothesize that exposure to a broader distribution of prefix-conditioned contexts during training reduces coupling among alternative solution trajectories, so that different prefixes at inference can more reliably elicit distinct yet valid solutions.

Beyond $\text{pass}@k$, we additionally report $\text{cons}@k$ (majority-vote accuracy over k samples) with 95% bootstrap confidence intervals on AIME25 and HMMT25. On AIME25, $\text{cons}@32$ improves by +2.70%, with the 95% CI shifting from [6.83, 17.71] to [7.15, 18.32]; on HMMT25, $\text{cons}@32$ improves by +2.09%, with the 95% CI shifting from [6.57, 17.56] to [7.27, 17.75]. These variance bounds indicate that the diversity gain is achieved without compromising decision stability. To further disentangle useful diversity from diffuse incorrect modes, we measure Answer Entropy, Majority Margin, and Modal Answer Flip Rate. Answer entropy rises from 1.31 \rightarrow 1.52 on AIME25 and 1.28 \rightarrow 1.47 on HMMT, while the majority margin drops only moderately (71.3% \rightarrow 64.8% on AIME25) and the modal-answer flip rate stays nearly unchanged (12.5% \rightarrow 13.2%), which indicates the added diversity is primarily at the solution-strategy level rather than at the final-answer level.

4.4 Ablation Study

We ablate both the prefix type and the insertion position to examine their effects on performance and to understand the underlying mechanism of latent perturbation.

Prefix type Three types of prefixes are considered:

- (1) Ours: a templated natural-language prefix that includes a random seed;
- (2) Corpus span: randomly sampled spans from

a pretraining corpus used as perturbations;

- (3) Soft prefix: randomly initialized soft-embedding tokens sampled from the existing language-embedding distribution.

Prefix-conditioned inference is used to compare these variants, with results reported in Table 3. Our design achieves the best trade-off between output diversity and instruction-following. While random corpus spans induce larger distributional shifts in next-token logits, they often introduce uncontrollable semantic noise that competes with the original instruction, occasionally leading the model to hallucinate or deviate from the intended task. Soft-embedding prefixes, despite their theoretical flexibility in continuous latent space, exhibit high instability in zero-shot settings, frequently resulting in collapsed attention patterns and premature sequence termination. These observations suggest that semantically-neutral natural language serves as the most effective medium for latent perturbation. It leverages the model’s pre-trained linguistic priors to shift the context without breaking the internal coherence of the transformer’s hidden representations, ensuring that diversity is gained without sacrificing logical consistency.

Insertion position Table 3 compares placing the prefix before versus after the prompt. While diversity metrics remain comparable in both settings, the impact on instruction alignment is drastically different. Placing the prefix after the prompt frequently yields off-target responses, where the model either attempts to incorporate the random prefix into the task logic or fails to recognize the prompt’s instructions entirely.

We attribute this failure to the interruption of the causal dependency chain within the autoregressive transformer. Inserting a semantically irrelevant block between the instruction and the response

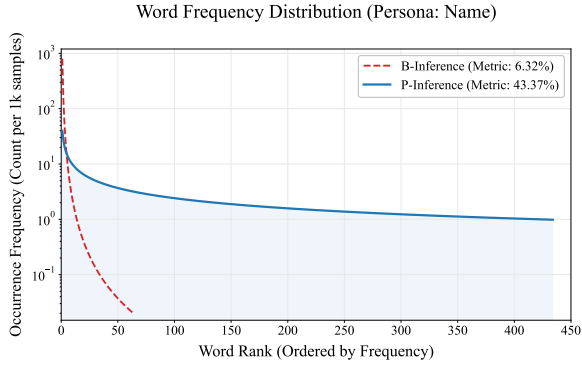


Figure 4: Word frequency distribution analysis for Persona Generation.

manifold disrupts the continuity of the KV-cache, forcing the model to re-contextualize the sequence attention from a noisy span. In contrast, placing the prefix at the very beginning allows the instruction I to be contextualized by z through self-attention from the initial layers, effectively biasing the latent manifold while keeping the primary semantic path from instruction to response intact. This confirms that the prefix acts as a global conditioning variable rather than a local noise source, necessitating its position at the root of the sequence to maximize its influence on the prior distribution while preserving strict adherence to the prompt.

4.5 Case Analysis

We conduct a detailed case analysis to investigate the specific improvements brought by our approach. In the persona generation task, as illustrated in Figure 4, it is most evident that the sampled responses from P-Inference tend toward a uniform distribution as the word rank increases, whereas the B-Inference results exhibit a severe and biased long-tail distribution where the generated responses are concentrated on a few modes rather than being sampled evenly across the potential token space. Similarly, for mathematical reasoning, we find that the additional performance gain stems from the enhanced diversity induced by the prefix. Specifically, the combination of P-Inference and P-SFT elicits entirely new solution strategies rather than merely exhibiting low token-level variance within a single, homogeneous reasoning path as seen in traditional SFT, thereby enabling a deeper level of exploration at the solution level.

4.6 Reinforcement Fine-Tuning

To examine whether the diversity gains induced by P-SFT translate into downstream benefits for

Model	HMMT	AIME25	LCB	GPQA-D
OpenThinker3-7B (SFT)				
Baseline	41.3	53.1	52.1	51.7
+ Iteration	43.3	53.2	52.7	51.8
+ Iteration 2	43.1	53.7	52.7	51.8
+ Iteration 3	43.2	54.1	53.1	51.7
OpenThinker3-7B (P-SFT)				
Baseline	42.5	53.9	53.7	52.1
+ Iteration 1	43.1	54.1	55.2	52.9
+ Iteration 2	43.7	54.3	55.1	52.7
+ Iteration 3	43.6	54.3	55.2	53.1

Table 4: Impact of RFT on reasoning performance. Baseline refers to the model after supervised fine-tuning, and iteration represents the round of RFT applied to each baseline model.

RL-style optimization, we perform Reinforcement Fine-Tuning (RFT) (Yuan et al., 2023) starting from both the SFT and P-SFT checkpoints. The fundamental motivation is that the performance ceiling of RFT is strictly governed by the agent’s exploration capability. A policy that can generate a wider variety of valid trajectories is more likely to discover global optima and avoid local mode saturation during the iterative self-boosting process. Concretely, for each prompt, 32 responses are sampled, and the model is iteratively optimized on the top-performing candidates for a total of three rounds.

Table 4 reports the comparative results. Under the RFT framework, P-SFT consistently maintains a performance margin over standard SFT across all iterations. Notably, as the number of self-boosting rounds increases, the performance of P-SFT saturates significantly more gradually than its SFT counterpart. We attribute this result to the enhanced exploration profile provided by P-SFT’s decoupled prior distribution. Standard SFT models, due to their inherent mode collapse, often produce highly homogeneous rollouts that lead to a rich-get-richer feedback loop. The policy repeatedly reinforces a narrow set of already-discovered correct paths, leading to premature convergence and diminishing returns in later iterations.

In contrast, P-SFT facilitates a more exhaustive exploration of the solution space by maintaining distinct response modes even as the policy parameters are updated. This characteristic ensures that the on-policy distribution remains sufficiently broad to capture alternative reasoning strategies that a collapsed policy would otherwise overlook.

By striking a superior balance between exploiting high-reward samples and preserving diversity, P-SFT serves as a more robust initialization for RL-based optimization. This property is particularly critical for hard reasoning tasks where the reward landscape is sparse, as it ensures that the model continues to synthesize novel, correct trajectories across successive rounds, pushing the ultimate performance ceiling higher than traditional SFT baselines.

5 Conclusion

This paper introduces Prefix-conditioned SFT (P-SFT), a simple yet effective approach to learning from instruction-following data while preserving response diversity. By assigning distinct, semantically neutral prefixes to decouple multiple valid responses for the same prompt, P-SFT effectively mitigates optimization interference and alleviates the mode collapse inherent in traditional supervised fine-tuning. Our experiments demonstrate that P-SFT substantially enhances the diversity of large language model (LLM) outputs across both creativity and hard reasoning benchmarks. More importantly, we show that preserving such diversity at the training stage is a prerequisite for effective test-time scaling and provides a superior exploration space for reinforcement learning, thereby raising the performance ceiling for complex reasoning. P-SFT demonstrates the broader potential of diversity-preserving fine-tuning methods in establishing a more robust foundation model for scalable reasoning and iterative LLM improvement.

Limitations

The primary focus of this work is addressing diversity collapse during the supervised finetuning (SFT) stage through prefix-conditioned SFT, which mitigates the issue of reduced output diversity when training on one-problem-multiple-answer pairs. However, diversity degradation can also occur during the reinforcement learning (RL) stage. Future work will explore applying this method to the RL phase.

Ethics Statements

This paper investigates methods to mitigate response collapse in large language models and to increase response diversity for identical prompts. Improving diversity can enhance sampling efficiency

by reducing redundant generations. Moreover, diversified outputs help capture a wider range of values and geographic perspectives, rather than exhibiting bias toward a narrow, overrepresented response pattern.

Acknowledgement

The research in this article is supported in part by WeBank (Grant No. WEB24EG01-L).

References

2025. Harvard-MIT Mathematics Tournament (HMMT). <https://www.hmmt.org/>.
- Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, Yian Zhang, Tugrul Konuk, and 117 others. 2025. *Llama-nemotron: Efficient reasoning models*. *Preprint*, arXiv:2505.00949.
- Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.
- Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, and 13 others. 2023. *Open problems and fundamental limitations of reinforcement learning from human feedback*. *Preprint*, arXiv:2307.15217.
- Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. 2025. *Pass@k training for adaptively balancing exploration and exploitation of large reasoning models*. *Preprint*, arXiv:2508.10751.
- François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.
- Leo Gao, John Schulman, and Jacob Hilton. 2022. *Scaling laws for reward model overoptimization*. *Preprint*, arXiv:2210.10760.
- Soumya Suvra Ghosal, Souradip Chakraborty, Avinash Reddy, Yifu Lu, Mengdi Wang, Dinesh Manocha, Furong Huang, Mohammad Ghavamzadeh, and Amrit Singh Bedi. 2025. Does thinking more always help? understanding test-time scaling in reasoning models. *arXiv preprint arXiv:2506.04210*.

- Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, and 31 others. 2025. [Openthoughts: Data recipes for reasoning models](#). *Preprint*, arXiv:2506.04178.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Michael Hassid, Tal Remez, Jonas Gehring, Roy Schwartz, and Yossi Adi. 2024. [The larger the better? improved LLM code-generation via budget reallocation](#). In *First Conference on Language Modeling*.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text de-generation](#). In *International Conference on Learning Representations*.
- Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. 2025. [T1: Advancing language model reasoning through reinforcement learning and inference scaling](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 23976–24003. PMLR.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. [Live-codebench: Holistic and contamination free evaluation of large language models for code](#). *Preprint*, arXiv:2403.07974.
- Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. [Understanding the effects of r1hf on llm generalisation and diversity](#). *Preprint*, arXiv:2310.06452.
- Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilija Kulikov. 2025. [Diverse preference optimization](#). *Preprint*, arXiv:2501.18101.
- Sunbowen Lee, Junting Zhou, Chang Ao, Kaige Li, Xinrun Du, Sirui He, Haihong Wu, Tianci Liu, Jiaheng Liu, Hamid Alinejad-Rokny, Min Yang, Yitao Liang, Zhofutu Wen, and Shiwen Ni. 2025. [Quantification of large language model distillation](#). *Preprint*, arXiv:2501.12619.
- Dacheng Li, Shiyi Cao, Chengkun Cao, Xiuyu Li, Shangyin Tan, Kurt Keutzer, Jiarong Xing, Joseph E. Gonzalez, and Ion Stoica. 2025a. [S*: Test time scaling for code generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15964–15978, Suzhou, China. Association for Computational Linguistics.
- Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. [From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline](#). *Preprint*, arXiv:2406.11939.
- Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. 2025b. [Preserving diversity in supervised fine-tuning of large language models](#). *Preprint*, arXiv:2408.16673.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.
- Mathematical Association of America. 2025. American Invitational Mathematics Examination (AIME). <https://maa.org/maa-invitational-competitions/>.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025a. [s1: Simple test-time scaling](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20275–20321, Suzhou, China. Association for Computational Linguistics.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025b. [s1: Simple test-time scaling](#). *Preprint*, arXiv:2501.19393.
- Laura O’Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. 2024. [Attributing mode collapse in the fine-tuning of large language models](#). In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.
- Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. 2025. [Training software engineering agents and verifiers with SWE-gym](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 47717–47737. PMLR.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. [Gpqa: A graduate-level google-proof qa benchmark](#). *Preprint*, arXiv:2311.12022.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan

- Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Weizhou Shen, Ziyi Yang, Chenliang Li, Zhiyuan Lu, Miao Peng, Huashan Sun, Yingcheng Shi, Shengyi Liao, Shaopeng Lai, Bo Zhang, Dayiheng Liu, Fei Huang, Jingren Zhou, and Ming Yan. 2025. [Qwenlong-11.5: Post-training recipe for long-context reasoning and memory management](#). *Preprint*, arXiv:2512.12967.
- Iliia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2024. [The curse of recursion: Training on generated data makes models forget](#). *Preprint*, arXiv:2305.17493.
- Benedikt Stroebel, Sayash Kapoor, and Arvind Narayanan. 2024. Inference scaling flaws: The limits of llm resampling with imperfect verifiers. *arXiv preprint arXiv:2411.17501*.
- Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, 2024. URL <https://qwenlm.github.io/blog/qwq-32b-preview>.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Fanqi Wan, Weizhou Shen, Shengyi Liao, Yingcheng Shi, Chenliang Li, Ziyi Yang, Ji Zhang, Fei Huang, Jingren Zhou, and Ming Yan. 2025. [Qwenlong-11: Towards long-context large reasoning models with reinforcement learning](#). *Preprint*, arXiv:2505.17667.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. [Self-consistency improves chain of thought reasoning in language models](#). *Preprint*, arXiv:2203.11171.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Yumeng Wang, Zhiyuan Fan, Jiayu Liu, Jen tse Huang, and Yi R. Fung. 2026. [Diversity-enhanced reasoning for subjective questions](#). *Preprint*, arXiv:2507.20187.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*.
- Yi Xin, Siqi Luo, Qi Qin, Haoxing Chen, Kaiwen Zhu, Zhiwei Zhang, Yangfan He, Rongchao Zhang, Jinbin Bai, Shuo Cao, and 1 others. 2025. [dmllm-tts: Self-verified and efficient test-time scaling for diffusion multi-modal large language models](#). *arXiv preprint arXiv:2512.19433*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025a. [Dapo: An open-source llm reinforcement learning system at scale](#). *Preprint*, arXiv:2503.14476.
- Zhaojian Yu, Yinghao Wu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2025b. [Z1: Efficient test-time scaling with code](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2688–2712, Suzhou (China). Association for Computational Linguistics.
- Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. [Scaling relationship on learning mathematical reasoning with large language models](#). *Preprint*, arXiv:2308.01825.
- Longfei Yun, Chenyang An, Zilong Wang, Letian Peng, and Jingbo Shang. 2025. [The price of format: Diversity collapse in llms](#). *Preprint*, arXiv:2505.18949.
- Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyang Shi. 2025a. [Verbalized sampling: How to mitigate mode collapse and unlock llm diversity](#). *Preprint*, arXiv:2510.01171.
- Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. 2025b. [Noveltybench: Evaluating language models for humanlike diversity](#). *Preprint*, arXiv:2504.05228.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). *Preprint*, arXiv:2306.05685.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.

A Author Contributions

Zhiyuan Fan conceived the original idea, led the project, and provided the computing resources and API credits that supported this work. Zhiyuan Fan and Guanqiao Chen designed and conducted the experiments and co-wrote the paper. Yanyi Huang and Zhiyuan Fan developed the theoretical analysis. Yanyi Huang prepared the figures. Dadi Guo and Mingkuan Zhao contributed insightful discussions and constructive suggestions.

B Implementation details

All training was conducted on 32 H200 GPUs using the LLaMA-Factory library. For inference, we use vLLM for parallel sampling. For the learning rate, we adopt a relatively large value of $1e-4$ due to differences in the parameter distributions between the instruction model and the reasoning-enhanced model. During sampling, we use the recommended temperature = 0.6 for all conventional parallel sampling. For prefix-conditioned inference, we fix temperature = 0 for each prefix to ensure reproducibility. This choice is deliberate and serves as a controlled experimental design: standard sampling ($T = 0.6$) derives diversity solely from decoding stochasticity, whereas prefix-conditioned inference derives diversity from the prefix-induced prior shift alone; using $T > 0$ for prefix-conditioned inference would confound these two sources, making it unclear whether gains come from the prefix mechanism or from sampling noise. Note that although the temperature is 0 for P-Inference, we can still obtain diverse responses, as the prefixes already shift the prior distribution.

RFT setup. For the reinforcement fine-tuning experiments in Section 4.6, rollouts use temperature = 0.6 and maximum response length = 8192, with 32 responses sampled per prompt. RFT training uses batch size = 512 and learning rate = $1e-4$. The prefix is applied only during rollout to elicit diverse trajectories and is removed during the RFT gradient update, since its role is limited to inducing exploration rather than being part of the learned policy.

C Theoretical proof

C.1 Semantic Consistency

We consider the following spaces in our analysis: Input Space \mathcal{X} , Semantic Space \mathcal{S} , Prefix Space \mathcal{C} , Output Space \mathcal{Y} , $\mathcal{D} = \{(x_k, y_k, c_k)\}_{k=1}^K$ training dataset.

Target: Given Input $x \in \mathcal{X}$, $y \in \mathcal{C}$, we aim to prove that the model’s posterior distribution over Semantics $P(S|x, c)$ remains concentrated around S_{true} .

Standard SFT:

$$P_{\text{base}}(S|x) = \sum_{c \in \mathcal{C}} P(c|x)P(S|x, c)$$

Using Bayes’ theorem to derive this equality, we need to make certain assumptions, namely that prefix c and semantics S should be decoupled. This requires that the reverse distribution of c does not disturb the true semantics:

$$P(S|x, c) = \frac{P(c|S, x)P(S|x)}{P(c|x)}$$

where $P(c|S, x)$ is the prefix likelihood.

Let $c \perp S|x$, i.e., $P(c|S, x) = P(c|x)$. Under this conditional independence assumption:

$$\Rightarrow P(S|x, c) = \frac{P(c|x)P(S|x)}{P(c|x)} = P(S|x).$$

This shows that prefix c is designed to be independent of S , preserving the semantic distribution.

Model: We can interpret the output as $y = \phi(S, z)$ where S represents semantics and z represents form/style.

$$P(S|x, c) = P(S|x) \quad (\text{semantics decoupled})$$

$$P(z|x, c) \neq P(z|x) \quad (\text{form/style perturbed})$$

$$P(y|x, c) \propto P(S|x) \cdot P(z|c)$$

Multi-response scenario: Let S_{true} denote correct semantics and S_{false} denote incorrect semantics.

$$P(S|x, c) = \frac{P(x, c|S)P(S)}{P(x, c)}$$

The posterior odds ratio is:

$$\frac{P(S_{\text{true}}|x, c)}{P(S_{\text{false}}|x, c)} = \frac{P(x, c|S_{\text{true}})P(S_{\text{true}})}{P(x, c|S_{\text{false}})P(S_{\text{false}})} = \frac{P(x|S_{\text{true}})P(S_{\text{true}})}{P(x|S_{\text{false}})P(S_{\text{false}})} \cdot \frac{P(c|S_{\text{true}})}{P(c|S_{\text{false}})}$$

Assuming that given S , x and c are conditionally independent (we construct c such that x and prefix c are independent given semantics):

$$\frac{P(S_{\text{true}}|x, c)}{P(S_{\text{false}}|x, c)} = \underbrace{\frac{P(x|S_{\text{true}})}{P(x|S_{\text{false}})}}_{\text{Input strength}} \cdot \underbrace{\frac{P(c|S_{\text{true}})}{P(c|S_{\text{false}})}}_{\text{Prefix Bias}} \cdot \underbrace{\frac{P(S_{\text{true}})}{P(S_{\text{false}})}}_{\text{Prior odds}}$$

When the input is sufficiently informative and the prefix has no semantic influence (i.e., $P(c|S_{\text{true}}) \approx P(c|S_{\text{false}})$), the posterior odds are primarily determined by the input strength, ensuring semantic consistency in standard SFT without prefix conditioning.

C.2 Variance Analysis via Law of Total Variance

We now analyze how prefix conditioning affects the variance of model outputs using the law of total variance.

Law of Iterated Expectations:

$$\mathbb{E}[Y] = \mathbb{E}_c (\mathbb{E}[Y|c])$$

Definition of Conditional Variance:

$$\begin{aligned} \text{Var}(Y|c) &= \mathbb{E}[Y^2|c] - [\mathbb{E}(Y|c)]^2 \\ \Rightarrow \mathbb{E}[Y^2|c] &= \text{Var}(Y|c) + [\mathbb{E}(Y|c)]^2 \end{aligned}$$

Variance decomposition:

$$\text{Var}(Y) = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2$$

By the law of iterated expectations:

$$\begin{aligned} \mathbb{E}[Y^2] &= \mathbb{E}_c (\mathbb{E}[Y^2|c]) \\ &= \mathbb{E}_c (\text{Var}(Y|c) + [\mathbb{E}[Y|c]]^2) \\ &= \mathbb{E}_c (\text{Var}(Y|c)) + \mathbb{E}_c [\mathbb{E}[Y|c]]^2 \end{aligned}$$

And:

$$(\mathbb{E}[Y])^2 = (\mathbb{E}_c [\mathbb{E}[Y|c]])^2$$

Therefore:

$$\begin{aligned} \Rightarrow \text{Var}(Y) &= \mathbb{E}_c [\text{Var}(Y|c)] + \underbrace{\mathbb{E}_c (\mathbb{E}[Y|c])^2 - (\mathbb{E}_c [\mathbb{E}[Y|c]])^2}_{\text{Prefix}^2} \\ &= \underbrace{\mathbb{E}_c [\text{Var}(Y|c)]}_{\text{Within-group variance due to } c} + \underbrace{\text{Var}_c (\mathbb{E}[Y|c])}_{\text{Between-group variance. Group variance } \geq 0} \end{aligned}$$

Baseline comparison:

$$\mathbb{E}[Y_{\text{base}}] \approx \mathbb{E}_c [\mathbb{E}[Y|c]] \quad (\text{No significant bias introduced.})$$

$$\text{Var}(Y_{\text{base}}) \approx \mathbb{E}_c (\text{Var}(Y|c)) \quad (\text{Bias term } \approx 0)$$

$$\Rightarrow \text{Var}(Y) > \text{Var}(Y_{\text{base}}) \Rightarrow \text{Diversity increases}$$

C.3 Training Stability Analysis

We analyze the training stability when using prefix conditioning by examining the distance between hidden representations.

Let input x be the question and c be the prefix. We use h to denote the embedding or hidden representation at a certain layer. Given: $h_{x,c}$: hidden representation when both x and c are present; h_x : hidden representation when only x is present.

Since prefix is relatively short and does not significantly change the meaning of input x , we expect a small distance between representations, i.e.,

$$\|h_{x,c} - h_{\text{base}}\|_2 \leq b$$

By the log-Lipschitz property of neural networks, the log-likelihood function $f(h) = \log \pi_\theta(y|h)$ is Lipschitz continuous in the hidden space. That is, given h_1, h_2 , there exists K (Lipschitz constant) such that:

$$|f(h_1) - f(h_2)| \leq K \|h_1 - h_2\|$$

Target: The SFT loss $\mathcal{L}_{\text{SFT}}(x) = -\log \pi_\theta(y|x)$ should be stable.

If y is sampled from the distribution $\pi_\theta(y|x, c)$ during training, then without prefix conditioning, the model tries to fit x in the "wrong" way.

That is, at inference time: $\log \pi_\theta(y|x, c) \gg M \rightarrow -\infty$

Therefore:

$$|\log \pi_\theta(y|x, c) - \log \pi_\theta(y|x)| \leq K \|h_{x,c} - h_{\text{base}}\|_2$$

$$\begin{aligned} \Rightarrow \log \pi_\theta(y|x) &\geq \log \pi_\theta(y|x, c) - K \|h_{x,c} - h_{\text{base}}\|_2 \\ &\geq \log \pi_\theta(y|x, c) - Kb \\ &\geq M - Kb \end{aligned}$$

$$\Rightarrow \mathcal{L}_{\text{SFT}}(x) \leq Kb - M \quad (\text{bounded})$$

As long as b is sufficiently small and K is not too large (when c has minimal impact on the representation), K will not be too large (the model is not too sensitive). Thus the loss remains bounded.

Therefore, even in the special case of standard SFT (because loss is bounded), training will converge to a reasonable point. (Due to bounded loss) Training will not diverge. As long as Kb is sufficiently small \Rightarrow training will stabilize and converge.

D Case Analysis

As shown in Table 5, for prompts with strong factual determinism, adding a semantically irrelevant prefix has a much smaller impact on the next-token logits than for open-ended prompts without a specific ground truth. This suggests that introducing prefixes can influence the LLM's decoding behavior to produce more diverse outputs, while not overly degrading its reasoning ability or factual knowledge. Our experiments on the reasoning benchmarks in the main text further support this observation.

E Detail Studies of NoveltyBench

Novelty Bench

Question: What is the best book of all time?

Traditional Inference: War and Peace by Leo Tolstoy is often regarded as the best book of all time due to its epic scope and profound exploration of human life during the Napoleonic Wars. The novel follows several aristocratic families in Russia, including the Rostovs, Bolkonskys, and Bezukhovs,

Prompt	Cosine Sim	L2 Dist	KL Div	Top-10 Overlap
The capital of France is	0.973	475.9	0.557	0.433
2 + 2 equals	0.986	404.9	0.416	0.567
The color of the sky is	0.965	515.2	1.648	0.400
Machine learning is a	0.964	423.6	0.791	0.700
In the beginning, there was	0.986	329.6	0.283	0.867
Python is a programming	0.984	405.8	0.006	0.800
The sun rises in the	0.991	236.9	0.056	0.567
Water boils at	0.559	1307.1	0.150	0.700
She walked into the	0.990	198.4	0.063	0.800
The answer to life is	0.981	878.6	0.231	0.500

Table 5: Comparison of similarity metrics for next-token logit distributions under different prefixes across prompt categories.

as they navigate love, loss, and societal changes amid historical events like the invasion of Russia by Napoleon in 1812. Tolstoy masterfully weaves together personal stories with broader historical narratives, showing how individual lives are shaped by larger forces. For instance, Pierre Bezukhov’s journey from a naive heir to a man seeking deeper meaning through Freemasonry and imprisonment reflects themes of self-discovery and redemption. Prince Andrei Bolkonsky’s experiences on the battlefield highlight the futility of war and the search for purpose, while Natasha Rostova’s youthful enthusiasm and growth represent the joys and pains of coming of age. The book’s length, over 1,200 pages, allows for detailed character development and intricate plotlines, making readers feel immersed in 19th-century Russian society. Tolstoy draws from real historical figures and events, blending fiction with nonfiction to question how history is made—not by great leaders alone, but by the collective actions of ordinary people. This challenges traditional views of heroism and power. Philosophically, the novel delves into questions of fate versus free will, with Tolstoy arguing that events unfold through a complex web of causes rather than singular decisions. The prose is straightforward yet descriptive, capturing everyday details like family dinners, balls, and military strategies. Critics praise its realism and psychological insight, influencing countless writers since its publication in 1869. *War and Peace* has been adapted into films, operas, and TV series, proving its enduring appeal. It offers lessons on resilience, morality, and the human condition, reminding readers that even in chaos, personal growth is possible. Ultimately, its combination of history, drama, and philosophy makes it a cornerstone of world literature, appealing to those who appreciate deep, thoughtful storytelling over quick reads. Many lists, like those from *Time Magazine* or *The Guardian*, rank it at the top for its ambition and impact. Reading it requires commitment, but the rewards are immense, providing a panoramic view of life that few books can match.

Prefix Inference: Gabriel García Márquez’s *One Hundred Years of Solitude* emerges as the unparalleled masterpiece of world literature, a swirling tapestry of magical realism that encapsulates the boundless absurdity, beauty, and tragedy of human existence across generations in the fictional town of Macondo. Through the Buendía family’s epic saga, Márquez weaves a narrative that defies linear time, blending the mundane with the miraculous—rains that last for years, levitating priests, and gypsies bearing ice as a wonder—to mirror Latin America’s turbulent history of colonialism, civil wars, and forgotten revolutions. The novel’s genius lies in its profound exploration of solitude as an inescapable human curse; each Buendía member, from the visionary founder José Arcadio Buendía, who ties himself to a tree in madness, to the resilient Ursula Iguarán, who outlives eras with unyielding matriarchal strength, grapples with isolation amid familial bonds that both nurture

and destroy. Márquez infuses the story with lush sensory immersion: the metallic tang of alchemy experiments, the oppressive humidity of endless downpours, or the ethereal ascent of Remedios the Beauty into the sky amid fluttering bedsheets. Philosophically, it critiques the cyclical nature of history and progress; Macondo's rise and fall symbolize how societies repeat mistakes, from exploitative banana massacres echoing real events to the amnesia induced by insomnia plagues, erasing collective memory. As a stylistic triumph, the book employs a polyphonic voice that shifts seamlessly between omniscient narration and intimate revelations, drawing on oral storytelling traditions to create a mythic quality that feels both ancient and urgently modern. With over 100 characters spanning seven generations, it delves into themes of fate versus agency, the illusion of progress, and the interplay of reality and myth, challenging readers to question their own perceptions of time and legacy. Márquez's prose is a hypnotic blend of poetic lyricism and raw irony, where butterflies trail a doomed lover or a blood trail snakes through town like a living vein, evoking a visceral sense of wonder and despair. Culturally, it birthed magical realism as a global phenomenon, influencing writers worldwide and capturing the essence of postcolonial identity struggles. Yet, at its core, *One Hundred Years of Solitude* is a profound meditation on love's redemptive power amid inevitable decay, offering a kaleidoscopic vision of life that affirms humanity's resilience even as prophecies unfold to Macondo's apocalyptic end, scripted on ancient parchments. This novel doesn't just recount a story; it conjures a universe where the extraordinary is ordinary, reminding us that in the solitude of existence, our shared dreams and follies connect us across time, making it an eternal beacon for understanding the human soul's intricate labyrinth.

Diversity Analysis Prefix inference chooses a completely different book, Gabriel García Márquez's *One Hundred Years of Solitude*, instead of the traditional *War and Peace*, and delivers a richly diversified set of perspectives: magical realism that bends time and reality, the inescapable solitude that haunts every generation of the Buendía family, vivid sensory details (endless rains, levitating priests, metallic alchemy smells), historical allegory of Latin America's cycles of violence and colonialism, philosophical reflections on fate, memory, and progress, polyphonic storytelling rooted in oral tradition, cultural birth of a global literary movement, and profound existential meditations on love, decay, and human resilience. In contrast, the traditional inference stays narrowly focused on just one book—*War and Peace* by Leo Tolstoy—repeating the same handful of angles: epic historical scope, aristocratic family stories, Napoleonic Wars, character development, and broad philosophical ideas about fate and free will. The result is a linear, factual summary that feels like a standard encyclopedia entry, with only 3–4 core dimensions and high overlap with almost any other generic answer. This single-book, single-track approach makes traditional inference predictable and repetitive, while prefix inference opens up a much wider, fresher space of ideas and language.

F Detail Studies of Full Story Generation

Full story

Seed Word: Fires Queen

Traditional Inference: In the ancient kingdom of Eldoria, the Fires Queen ruled with an iron fist from her throne of smoldering embers. Her name was whispered in fear: Queen Ignara, the one who commanded flames like loyal soldiers. Born from the ashes of a volcanic eruption, she had risen to power by taming the wild fires that plagued the land. Her palace was a fortress of blackened stone, where torches burned eternally, casting flickering shadows on walls etched with tales of her conquests. One fateful night, a rebellion stirred in the villages below. Peasants, tired of the queen's relentless demands for tribute in the form of firewood and coal, gathered under the cover of darkness. Led by a blacksmith named Thorne, they plotted to douse the eternal flames that fueled her magic.

Thorne had forged a secret weapon: a blade quenched in the icy waters of the northern springs, said to extinguish any fire it touched. As the rebels marched toward the palace, Queen Ignara sensed the disturbance. She stood atop her tower, her crimson gown billowing like flames in the wind. With a wave of her hand, she summoned walls of fire that encircled the intruders. "Fools," she hissed, her voice crackling like burning wood. "You dare challenge the Fires Queen?" The flames roared higher, consuming the forest paths and forcing the rebels to retreat. Thorne, undeterred, pressed on alone, his blade gleaming in the firelight. In the heart of the palace, the confrontation unfolded. Queen Ignara unleashed torrents of fire, but Thorne dodged and parried, his weapon slicing through the inferno. Sparks flew as blade met flame, and for a moment, the queen's power wavered. She laughed maniacally, her eyes glowing like coals. "You cannot quench what is eternal!" But Thorne struck true, piercing her fiery core. The flames sputtered and died, leaving only smoke and silence. With the Fires Queen defeated, the kingdom breathed freely. Thorne became the new guardian, ensuring fires served the people rather than enslaved them. Yet, in the embers of her throne, a faint glow remained—a reminder that fire, like power, could always reignite. The land prospered under balanced rule, but whispers of her return lingered in every hearth. The story of the Fires Queen became legend, taught to children as a cautionary tale. Fire was both destroyer and creator, and those who wielded it must do so wisely. Eldoria entered an era of peace, where flames warmed homes instead of burning them. Thorne's blade was enshrined, a symbol of courage against tyranny. And so, the cycle of fire continued, tempered by the lessons of the past.

Prefix Inference: Amid the swirling mists of a forgotten realm where elements danced in eternal harmony, the Fires Queen emerged not as a tyrant, but as a multifaceted force of transformation. Known as Queen Aelara, she embodied the dual nature of fire: the scorching blaze that razed forests and the gentle hearth that nurtured life. Her origins were shrouded in myth—some said she was forged in the heart of a dying star, others claimed she rose from the ritual pyres of ancient shamans. Her court was a vibrant tapestry, blending volcanic forges with lush gardens where fire lilies bloomed under controlled embers. In this world, fire was more than destruction; it was alchemy, passion, and renewal. Queen Aelara's reign began with a cataclysm: a great inferno that threatened to engulf the continents. But she intervened, channeling the flames into rivers of molten gold that enriched the soil. Villages flourished, artisans crafted wonders from tempered metals, and healers used her controlled burns to purify plagues. Yet, shadows loomed—jealous rivals from the Water Realms and Earth Clans sought to undermine her, fearing her unpredictable power. One epochal dawn, an alliance formed against her. A water sorceress named Lirra, an earth guardian called Terron, and a wind nomad, Zephyr, united to steal the Eternal Spark, the crystal that anchored Aelara's essence. They infiltrated her palace, a labyrinth of glowing caverns and sky-high spires, where flames whispered secrets in multiple tongues. Aelara, ever perceptive, anticipated their moves. She didn't merely defend; she adapted, weaving fire with elements of her foes—creating steam veils from water, forging unbreakable alloys with earth, and fueling gusts into firestorms with wind. The battle was a symphony of chaos and creation. Lirra's waves clashed with Aelara's infernos, birthing rainbows of vapor. Terron's boulders melted into lava sculptures, alive with inner glow. Zephyr's tempests fanned flames into protective barriers. Amid the fray, Aelara revealed her true philosophy: "Fire is not singular; it evolves through union." She offered alliance instead of annihilation, transforming enemies into allies. Together, they quelled a greater threat—a void entity devouring realms. Under the Fires Queen's guidance, the alliance birthed a new era. Fire integrated with water for irrigation miracles, earth for fertile rebirths after burns, and wind for spreading seeds across lands. Aelara's court became a hub of innovation: inventors harnessed geothermal energy for flying machines, artists painted with living flames, and scholars decoded fire's language in stars. Challenges persisted—droughts tested balance, rebellions arose from those fearing change—but Aelara's approach fostered growth. Legends of the Fires Queen spread across dimensions, inspiring beings to embrace multiplicity. Fire symbolized not just heat, but the spark of ideas, the forge of bonds, and the light of discovery. In her realm, every ember told a story of

Question: Mark writes the expression \sqrt{abcd} on the board, where $abcd$ is a four-digit number and $a \neq 0$. Derek, a toddler, decides to move the a , changing Mark's expression to $a\sqrt{bcd}$. Surprisingly, these two expressions are equal. Compute the only possible four-digit number $abcd$.

Traditional inference

Let $y = 100b + 10c + d$. From $\sqrt{1000a + y} = a\sqrt{y}$, squaring both sides yields:

$$1000a + y = a^2y \quad \Rightarrow \quad y = \frac{1000a}{a^2 - 1}$$

Enumerate $a = 2, 3, \dots, 9$ to find integer $y \in [100, 999]$:

- $a = 2$: $y = 2000/3 \approx 666.67$ (not an integer)
- $a = 3$: $y = 3000/8 = 375$ ✓

Solution: $abcd = \boxed{3375}$

Characteristics: Single algebraic starting point; sequential enumeration; no back-substitution verification.

Prefix Id 2

Let $y = 100b + 10c + d$. From $\sqrt{1000a + y} = a\sqrt{y}$, squaring both sides yields:

$$y = \frac{1000a}{a^2 - 1}$$

Enumerate $a = 1, 2, \dots, 9$ to find integer $y \in [100, 999]$:

- $a = 1$: undefined
- $a = 2$: $y = 2000/3 \approx 666.67$ (not an integer)
- $a = 3$: $y = 3000/8 = 375$ ✓

Verification by exact simplification:

$$\sqrt{3375} = 15\sqrt{15}, \quad 3\sqrt{375} = 3 \times 5\sqrt{15} = 15\sqrt{15} \quad \checkmark$$

Solution: $abcd = \boxed{3375}$

Characteristics: Sequential enumeration + exact algebraic verification via radical simplification.

Prefix Id 6

Let $y = 100b + 10c + d$. From $\sqrt{1000a + y} = a\sqrt{y}$, squaring both sides yields:

$$y = \frac{1000a}{a^2 - 1}$$

Enumerate $a = 9, 8, \dots, 1$ to find integer $y \in [100, 999]$:

- $a = 9$: $y = 9000/80 = 11.25$ (not an integer)
-
- $a = 3$: $y = 3000/8 = 375$ ✓

Verification by exact simplification:

$$\sqrt{3375} \approx 58.09, \quad 3\sqrt{375} \approx 58.09\checkmark$$

Solution: $abcd = \boxed{3375}$

Characteristics: Linear reverse derivation + numerical (approximate) verification

Prefix Id 9

From $y = \frac{1000a}{a^2-1}$, for y to be an integer, $(a^2 - 1)$ must divide $1000a$.

Key insight: List all divisors of 1000:

$$1, 2, 4, 5, 8, 10, 20, 25, 40, 50, 100, 125, 200, 250, 500, 1000$$

Reverse matching: Find which divisor d satisfies $d = a^2 - 1$ for some digit a :

- $d = 8$: $a^2 = 9 = 3^2 \Rightarrow a = 3 \checkmark$
- Other divisors: $a^2 = d + 1$ is not a perfect square

Thus $a = 3$, $y = \frac{3000}{8} = 375$. Solution: $abcd = \boxed{3375}$

Characteristics: Divisor analysis with reverse matching; a fundamentally different starting point from enumeration.

Prefix Id 10

Same derivation: $y = \frac{1000a}{a^2-1}$. Enumerate $a = 2, 3, \dots, 9$, find $a = 3$, $y = 375$.

Difference from Traditional: Verification via *prime factorization*:

$$3375 = 5^3 \times 3^3 = 125 \times 27$$

$$\sqrt{3375} = \sqrt{5^3 \times 3^3} = 5 \cdot 3 \cdot \sqrt{5 \times 3} = 15\sqrt{15}$$

$$3\sqrt{375} = 3\sqrt{5^3 \times 3} = 3 \times 5\sqrt{15} = 15\sqrt{15} \checkmark$$

Characteristics: Sequential enumeration + prime factorization verification.

Prefix Id 20 From $y = \frac{1000a}{a^2-1}$, for y to be an integer, $(a^2 - 1)$ must divide $1000a$.

Exhaustively check *all 15 divisors* of 1000, testing whether $d + 1$ is a perfect square:

$d = a^2 - 1$	$a^2 = d + 1$	Result
2	3	not a perfect square
4	5	not a perfect square
5	6	not a perfect square
8	$9 = 3^2$	$a = 3 \checkmark$
10	11	not a perfect square
20	21	not a perfect square
25	26	not a perfect square
40	41	not a perfect square
50	51	not a perfect square
100	101	not a perfect square
125	126	not a perfect square
200	201	not a perfect square
250	251	not a perfect square
500	501	not a perfect square
1000	1001	not a perfect square

Only $d = 8$ yields $a = 3$. Thus $y = \frac{3000}{8} = 375$. Solution: $abcd = \boxed{3375}$

Characteristics: Exhaustive divisor checking; guarantees uniqueness of solution.

Prefix Id 27

New variable setup: Let $N = \overline{abcd}$ (four-digit number), $M = \overline{bcd}$ (three-digit number).

Relations: $N = 1000a + M$ and $N = a^2M$ (from squaring $\sqrt{N} = a\sqrt{M}$).

Key transformation: Eliminate M by substituting $M = N - 1000a$:

$$N = a^2(N - 1000a) = a^2N - 1000a^3$$
$$N(1 - a^2) = -1000a^3 \quad \Rightarrow \quad N = \frac{1000a^3}{a^2 - 1}$$

Difference from Traditional: Directly solve for N (target), not M .

Non-standard enumeration: Start from middle value $a = 5$, then expand outward:

$$a = 5, 6, 7, 8, 9, 4, 3, \dots$$

- $a = 5$: $N = 125000/24 \approx 5208.33$ (not an integer)
- $a = 6$: $N = 216000/35 \approx 6171.43$ (not an integer)
- \vdots
- $a = 3$: $N = 27000/8 = 3375$ ✓

Solution: $abcd = \boxed{3375}$

Characteristics:

- Alternative algebraic formulation: solve for N directly instead of M
 - Target-oriented modeling: derive expression for the final answer
 - Middle-out enumeration strategy
-

Additional Problem-Solving Strategies in Prefix Inference Compared to Traditional Inference

Traditional Inference follows a single path: derive $y = \frac{1000a}{a^2-1}$, enumerate $a = 2$ to 9 , stop upon finding $a = 3$, $y = 375$.

Prefix Inference introduces the following additional strategies:

1. **Exact Algebraic Verification** (Prefix-2): After finding the solution, verify by simplifying radicals: $\sqrt{3375} = 15\sqrt{15}$ and $3\sqrt{375} = 15\sqrt{15}$.
2. **Numerical Approximation Verification** (Prefix-6): Verify by computing decimal values: $\sqrt{3375} \approx 58.0948$ and $3\sqrt{375} \approx 58.0947$.
3. **Prime Factorization Verification** (Prefix-10): Verify by decomposing $3375 = 5^3 \times 3^3$, then simplify $\sqrt{3375} = 15\sqrt{15}$.
4. **Reverse Enumeration Order** (Prefix-6): Test $a = 10, 9, 8, \dots, 3$ (large to small) instead of $a = 2, 3, \dots, 9$.
5. **Middle-Out Enumeration Order** (Prefix-27): Test $a = 5, 6, 7, 8, 9, 4, 3$ (starting from middle, expanding outward).

6. **Divisor Analysis with Reverse Matching** (Prefix-9): Instead of enumerating a , list divisors of 1000 and find which divisor equals $a^2 - 1$. Only $8 = 3^2 - 1$ works, giving $a = 3$.
7. **Exhaustive Divisor Checking** (Prefix-20): Check all 15 divisors d of 1000, testing whether $d + 1$ is a perfect square. Guarantees uniqueness.
8. **Alternative Algebraic Formulation** (Prefix-27): Directly solve for target $N = \frac{1000a^3}{a^2-1}$ instead of $M = \frac{1000a}{a^2-1}$, using substitution $M = N - 1000a$.

I Detail Studies of AIME24

Below, we provide an example from the AIME24 dataset that includes both Traditional Inference and various Prefix Inferences, demonstrating how Prefix Inference explores diverse reasoning paths.

AIME24

Question: Every morning Aya goes for a 9-kilometer-long walk and stops at a coffee shop afterwards. When she walks at a constant speed of s kilometers per hour, the walk takes her 4 hours, including t minutes spent in the coffee shop. When she walks $s + 2$ kilometers per hour, the walk takes her 2 hours and 24 minutes, including t minutes spent in the coffee shop. Suppose Aya walks at $s + \frac{1}{2}$ kilometers per hour. Find the number of minutes the walk takes her, including the t minutes spent in the coffee shop.

Traditional inference

Define s as speed (km/h) and t as coffee shop time (hours):

$$\begin{cases} \frac{9}{s} + t = 4 & \text{Eq. (1)} \\ \frac{9}{s+2} + t = 2.4 & \text{Eq. (2)} \end{cases}$$

Subtract Eq. (2) from Eq. (1) to eliminate t :

$$\frac{9}{s} - \frac{9}{s+2} = 4 - 2.4 \implies \frac{9(s+2) - 9s}{s(s+2)} = 1.6$$

$$\frac{18}{s^2 + 2s} = 1.6 \implies 1.6(s^2 + 2s) = 18$$

$$s^2 + 2s - 11.25 = 0$$

Solving for s using the quadratic formula:

$$s = \frac{-2 \pm \sqrt{2^2 - 4(1)(-11.25)}}{2(1)} = \frac{-2 \pm \sqrt{49}}{2}$$

$$s = \frac{-2 + 7}{2} = 2.5 \text{ km/h}$$

Substitute $s = 2.5$ back to find t :

$$t = 4 - \frac{9}{2.5} = 4 - 3.6 = 0.4$$

Final calculation for the new speed $s + 0.5 = 3$:

$$T_{\text{new}} = \frac{9}{2.5 + 0.5} + 0.4 = 3 + 0.4 = 3.4 \text{ hours}$$

$$3.4 \times 60 = \boxed{204} \text{ minutes}$$

Prefix ID 12

Step 1: Isolate the Reciprocal Speed

Let $x = \frac{t}{60}$ be the time spent in the coffee shop in hours. We rewrite the equations to isolate the terms containing s :

$$\begin{cases} \frac{9}{s} = 4 - x & \implies s = \frac{9}{4-x} \quad \text{--- (1)} \\ \frac{9}{s+2} = 2.4 - x & \implies s + 2 = \frac{9}{2.4-x} \quad \text{--- (2)} \end{cases}$$

Step 2: Eliminate s by Substitution

Substitute s from Eq. (1) into Eq. (2):

$$\frac{9}{4-x} + 2 = \frac{9}{2.4-x}$$

Step 3: Solve the Rational Equation for x

Multiply the entire equation by $(4-x)(2.4-x)$ to clear denominators:

$$9(2.4-x) + 2(4-x)(2.4-x) = 9(4-x)$$

Expand and simplify:

$$21.6 - 9x + 2(x^2 - 6.4x + 9.6) = 36 - 9x$$

$$21.6 + 2x^2 - 12.8x + 19.2 = 36$$

$$2x^2 - 12.8x + 4.8 = 0 \implies x^2 - 6.4x + 2.4 = 0$$

Step 4: Solve the Quadratic for t

Using the quadratic formula for x :

$$x = \frac{6.4 \pm \sqrt{(-6.4)^2 - 4(1)(2.4)}}{2} = \frac{6.4 \pm \sqrt{40.96 - 9.6}}{2}$$

$$x = \frac{6.4 \pm \sqrt{31.36}}{2} = \frac{6.4 \pm 5.6}{2}$$

This gives $x = 6$ (invalid, exceeds total time) or $x = 0.4$. Thus, $t = 0.4 \times 60 = 24$ minutes.

Step 5: Final Calculation

Substitute $x = 0.4$ back into Eq. (1) to find s :

$$\frac{9}{s} = 4 - 0.4 = 3.6 \implies s = 2.5 \text{ km/h}$$

For the speed $s + 0.5 = 3$ km/h, the total time is:

$$T = \left(\frac{9}{3}\right) \times 60 + 24 = 180 + 24 = \boxed{204} \text{ minutes}$$

Prefix ID 19

Step 1: Isolate the Walking Time Difference

Since the coffee shop time t is constant, the difference in total time ($4 - 2.4 = 1.6$ hours) is entirely due to the difference in walking time. Let W_1 be the walking time at speed s , and W_2 be the walking time at speed $s + 2$.

$$W_1 - W_2 = 1.6 \text{ hours} \quad \text{--- (1)}$$

Step 2: Use the Inverse Proportion of Speed and Time

For a constant distance (9 km), speed and time are inversely proportional:

$$\frac{W_1}{W_2} = \frac{s+2}{s} = 1 + \frac{2}{s}$$

Substituting $W_1 = W_2 + 1.6$ from Eq. (1):

$$\frac{W_2 + 1.6}{W_2} = 1 + \frac{1.6}{W_2} \implies 1 + \frac{1.6}{W_2} = 1 + \frac{2}{s}$$

This yields a direct linear relationship between speed and walking time:

$$\frac{1.6}{W_2} = \frac{2}{s} \implies s = \frac{2W_2}{1.6} = 1.25W_2$$

Step 3: Solve for Walking Time (W_2)

Substitute $s = 1.25W_2$ into the distance formula $W_2 = \frac{9}{s+2}$:

$$W_2 = \frac{9}{1.25W_2 + 2} \implies 1.25W_2^2 + 2W_2 - 9 = 0$$

Multiplying by 4 to clear decimals:

$$5W_2^2 + 8W_2 - 36 = 0$$

Factoring the quadratic: $(5W_2 + 18)(W_2 - 2) = 0$. Since $W_2 > 0$, we find $W_2 = 2$ hours.

Step 4: Resolve the Constants

If walking time $W_2 = 2$ hours, then:

- Coffee time: $t = 2.4$ (total) $- 2$ (walk) $= 0.4$ hours $= 24$ min.
- Speed $s + 2$: $s + 2 = \frac{9 \text{ km}}{2 \text{ h}} = 4.5 \implies s = 2.5 \text{ km/h}$.

Step 5: Final Result

At speed $s + 0.5 = 3 \text{ km/h}$, walking time is $\frac{9}{3} = 3$ hours (180 min). Total time: $180 + 24 = \boxed{204}$ minutes.

Prefix ID 27

Step 1: Time Conversion and Equation Setup

Convert the second total time to hours: $2 \text{ h } 24 \text{ min} = 2 + \frac{24}{60} = 2.4 \text{ h}$. Let s be the speed and $T_c = \frac{t}{60}$ be the coffee shop time in hours:

$$\begin{cases} \frac{9}{s} + T_c = 4 & \text{(Scenario 1)} \\ \frac{9}{s+2} + T_c = 2.4 & \text{(Scenario 2)} \end{cases}$$

Step 2: Subtraction and Rational Simplification

Subtracting the equations to eliminate T_c :

$$\frac{9}{s} - \frac{9}{s+2} = 4 - 2.4 \implies \frac{9(s+2) - 9s}{s(s+2)} = 1.6$$

$$\frac{18}{s^2 + 2s} = 1.6 \implies 18 = 1.6s^2 + 3.2s$$

Step 3: Coefficient Normalization

Rearrange and scale the quadratic equation to simplify calculation:

$$1.6s^2 + 3.2s - 18 = 0$$

Multiply by 10 and divide by 4 to normalize:

$$16s^2 + 32s - 180 = 0 \implies 4s^2 + 8s - 45 = 0$$

Step 4: Solving the Quadratic

Using the quadratic formula where $a = 4, b = 8, c = -45$:

$$s = \frac{-8 \pm \sqrt{8^2 - 4(4)(-45)}}{2(4)} = \frac{-8 \pm \sqrt{64 + 720}}{8}$$
$$s = \frac{-8 \pm \sqrt{784}}{8} = \frac{-8 \pm 28}{8}$$

Since $s > 0$, we have $s = \frac{20}{8} = 2.5$ km/h.

Step 5: Back-substitution and Final Answer

Solve for t (minutes):

$$\frac{9}{2.5} + \frac{t}{60} = 4 \implies 3.6 + \frac{t}{60} = 4 \implies \frac{t}{60} = 0.4 \implies t = 24 \text{ min}$$

New scenario with speed $s + 0.5 = 3$ km/h:

$$\text{Total Time} = \left(\frac{9}{3}\right) \times 60 + 24 = 180 + 24 = \boxed{204} \text{ minutes}$$

Prefix ID 31

Step 1: Shift to Minute-Based Equations

Instead of working with hours and decimals, we multiply the time equations by 60 immediately to align all units to minutes. Let s be the speed (km/h) and t be the coffee shop time (min):

$$\begin{cases} \frac{540}{s} + t = 240 & \text{(Scenario 1: 4 hours = 240 min)} \\ \frac{540}{s+2} + t = 144 & \text{(Scenario 2: 2.4 hours = 144 min)} \end{cases}$$

Step 2:

Subtracting the equations to isolate the walking time difference:

$$\frac{540}{s} - \frac{540}{s+2} = 240 - 144 = 96$$

Factor out 540 and simplify the fractions:

$$540 \left(\frac{2}{s(s+2)} \right) = 96 \implies \frac{1080}{s^2 + 2s} = 96$$

Step 3:

Rearrange into a quadratic form: $96s^2 + 192s - 1080 = 0$. Divide the entire equation by its GCD (24):

$$4s^2 + 8s - 45 = 0$$

Step 4:

Using the quadratic formula for $4s^2 + 8s - 45 = 0$:

$$s = \frac{-8 \pm \sqrt{64 - 4(4)(-45)}}{8} = \frac{-8 \pm \sqrt{784}}{8} = \frac{-8 \pm 28}{8}$$

Since $s > 0$, $s = \frac{20}{8} = 2.5$ km/h.

Step 5:

Substitute $s = 2.5$ into the minute-based equation:

$$\frac{540}{2.5} + t = 240 \implies 216 + t = 240 \implies t = 24 \text{ min}$$

For the final speed $s + 0.5 = 3$ km/h:

$$T_{\text{total}} = \frac{9}{3} \times 60 + 24 = 180 + 24 = \boxed{204} \text{ minutes}$$

Additional Problem-Solving Strategies in Prefix Inference Compared to Traditional Inference

Traditional inference defines speed s and coffee time t to build a distance-time system. Eliminate t to produce the quadratic $s^2 + 2s - 11.25 = 0$. Solve for s (retaining the positive root), back-substitute to find t , and finally compute the total time at speed $s + 0.5$.

Prefix Inference introduces the following additional strategies:

1. Structural Re-modeling

(Prefix 12): Reciprocal Isolation. Instead of direct elimination, express s as an explicit function of the coffee time x . By utilizing the identity $(s + 2) - s = 2$, this method isolates a single variable early, making the logical flow from coffee time to speed more transparent.

2. Physical Intuition

Prefix 19: Time Difference Analysis. This approach leverages the physical insight that the total time difference equals the walking time difference (since coffee time is constant). By using the inverse proportion between speed and time ($\frac{W_1}{W_2} = \frac{s+2}{s}$), it reduces the system to a simple linear relationship, significantly simplifying the final factorization.

3. Computational Optimization

(Prefix 27 & 31): Numerical Standardizing. These methods focus on reducing "mental tax" during calculation. Prefix 27 (Normalization) transforms decimal quadratics into integer-coefficient equations (e.g., $4s^2 + 8s - 45 = 0$) to facilitate mental factoring. Prefix 31 (Unit Pre-conversion) unifies all units into minutes at the start, preventing unit-conversion errors and naturally producing cleaner coefficients.

J Detai Studies of AIME25

Below, we provide an example from the AIME25 dataset that includes both Traditional Inference and various Prefix Inferences, demonstrating how Prefix Inference explores diverse reasoning paths.

AIME25

Question: Find the sum of all integer bases $b > 9$ for which 17_b is a divisor of 97_b .

Traditional inference

Step 1: Convert to Base 10

Represent the given base b numbers in decimal form:

$$17_b = 1 \cdot b + 7 = b + 7, \quad 97_b = 9 \cdot b + 7 = 9b + 7$$

Step 2: Establish the Divisibility Equation

For $b + 7$ to be a divisor of $9b + 7$, there must exist an integer k such that:

$$9b + 7 = k(b + 7)$$

Step 3: Isolate the Variable b

Rearrange the equation to express b in terms of k :

$$9b + 7 = kb + 7k \implies b(9 - k) = 7k - 7 \implies b = \frac{7k - 7}{9 - k}$$

Step 4: Range Analysis and Testing

Since $b > 9$, we must have $k < 9$ and k must be an integer. We test $k \in \{1, 2, \dots, 8\}$:

- If $k = 8$: $b = \frac{7(8) - 7}{9 - 8} = 49$ (Valid, $49 > 9$)
- If $k = 7$: $b = \frac{7(7) - 7}{9 - 7} = 21$ (Valid, $21 > 9$)
- If $k = 5$: $b = \frac{7(5) - 7}{9 - 5} = 7$ (Invalid, $b \leq 9$)

Step 5: Final Summation

The valid integer bases are $b = 49$ and $b = 21$. Their sum is:

$$49 + 21 = 70$$

Prefix ID 7

Step 1: Set up the Divisibility Condition

We are given that $b + 7$ must divide $9b + 7$. In notation:

$$(b + 7) \mid (9b + 7)$$

Step 2: Apply the Linear Combination Property

Since $b + 7$ divides itself, it also divides any multiple of itself. We subtract $9(b + 7)$ from $9b + 7$ to eliminate the variable b :

$$(b + 7) \mid (9b + 7) - 9(b + 7) \implies (b + 7) \mid (9b + 7 - 9b - 63)$$

This simplifies to:

$$(b + 7) \mid -56, \quad \text{or simply} \quad (b + 7) \mid 56$$

Step 3: Find Divisors under Constraints

Since $b > 9$, the divisor $b + 7$ must be greater than $9 + 7 = 16$. The divisors of 56 that are greater than 16 are 28 and 56.

Step 4: Solve for b and Summation

Setting $b + 7$ equal to each divisor:

$$b + 7 = 28 \implies b = 21$$

$$b + 7 = 56 \implies b = 49$$

The sum of all valid bases is $21 + 49 = 70$.

Prefix ID 22

Step 1: Express b as a Rational Function

From the condition $9b + 7 = k(b + 7)$, we have the expression:

$$b = \frac{7k - 7}{9 - k}$$

Step 2: Perform Algebraic Long Division

To simplify the fraction, rewrite the numerator to include a multiple of the denominator $(9 - k)$:

$$b = \frac{-7(9 - k) + 63 - 7}{9 - k} = \frac{-7(9 - k) + 56}{9 - k}$$

$$b = -7 + \frac{56}{9 - k}$$

Step 3: Analyze the Integer Condition

For b to be an integer, $9 - k$ must be a divisor of 56. Furthermore, for $b > 9$:

$$-7 + \frac{56}{9 - k} > 9 \implies \frac{56}{9 - k} > 16$$

Step 4: Identify Possible Divisors

The positive divisors of 56 are $\{1, 2, 4, 7, 8, 14, 28, 56\}$. We test $9 - k$ as a divisor:

- If $9 - k = 1$, then $b = -7 + 56 = 49$
- If $9 - k = 2$, then $b = -7 + 28 = 21$
- If $9 - k = 4$, then $b = -7 + 14 = 7$ (Invalid, $b \leq 9$)

Step 5: Summing the Solutions

The sum of valid bases b is $49 + 21 = 70$.

Additional Problem-Solving Strategies in Prefix Inference Compared to Traditional Inference

The traditional inference introduces a multiplier parameter k , transforms the divisibility relation into an equation, and uses variable separation to express b as a function of k , then performs an exhaustive search within the specified range of k .

Prefix Inference introduces the following additional strategies:

1. **Number Theory Method:** Prefix-7 utilizes the linear nature of the integer division (if $a \mid b$, then $a \mid mb - b$), transforming the number theory problem into a simpler equation by eliminating the variable b .
2. **Factorization Method:** Prefix-22 preserves the functional form and uses methods resembling "long division" to transfer the variable b in the numerator while keeping the structure of the equation in the form of integer + variable terms.

K Formal Derivation of Gradient Conflict under 1-to-M SFT

We provide a formal derivation showing that gradient conflict under 1-to-M SFT is a structural consequence of the softmax simplex constraint, not merely an empirical conjecture.

Consider an instruction I with two distinct responses R_1 and R_2 . Under standard autoregressive SFT, both responses are conditioned on the identical hidden state $h = f_\theta(I)$. At the first decoding position, the model produces a logit vector $z = Wh$, and the probability of token v is:

$$p_v = \frac{\exp(z_v)}{\sum_j \exp(z_j)}. \quad (7)$$

Let v_1 and v_2 denote the first tokens of R_1 and R_2 respectively, where $v_1 \neq v_2$. The SFT loss for each response at the first token position is $\mathcal{L}_m = -\log p_\theta(v_m | I)$. The standard softmax gradient with respect to the logit vector gives:

$$\nabla_z \mathcal{L}_1 = p - e_{v_1}, \quad \nabla_z \mathcal{L}_2 = p - e_{v_2}, \quad (8)$$

where p is the full probability vector and e_v is the one-hot indicator at position v .

Computing the gradient inner product:

$$\langle \nabla_z \mathcal{L}_1, \nabla_z \mathcal{L}_2 \rangle = (p - e_{v_1})^\top (p - e_{v_2}) = \|p\|^2 - p_{v_1} - p_{v_2} + \underbrace{\mathbb{1}[v_1 = v_2]}_{=0}. \quad (9)$$

Since $v_1 \neq v_2$, we have:

$$\langle \nabla_z \mathcal{L}_1, \nabla_z \mathcal{L}_2 \rangle = \|p\|^2 - p_{v_1} - p_{v_2}. \quad (10)$$

Gradient conflict occurs when $\langle \nabla_z \mathcal{L}_1, \nabla_z \mathcal{L}_2 \rangle < 0$. Suppose the model assigns combined probability $p_{v_1} + p_{v_2} = s$ with $p_{v_1} = p_{v_2} = s/2$ (symmetric case), and the remaining mass $1 - s$ is spread over $V - 2$ tokens. Then:

$$\|p\|^2 = 2 \cdot \left(\frac{s}{2}\right)^2 + (V - 2) \cdot \left(\frac{1 - s}{V - 2}\right)^2 = \frac{s^2}{2} + \frac{(1 - s)^2}{V - 2}. \quad (11)$$

When the remaining mass is approximately uniform over a large vocabulary ($V \gg 2$):

$$\|p\|^2 \approx \frac{s^2}{2}. \quad (12)$$

The conflict condition $\|p\|^2 < s$ becomes $\frac{s^2}{2} < s$, i.e., $s < 2$, which is always satisfied since $s \leq 1$.

To empirically validate the above analysis, we randomly sample 100 training instances from our dataset and measure the gradient cosine similarity between different response pairs for the same instruction. Under standard SFT (identical conditioning), the average gradient cosine similarity is approximately -0.37 to -0.81 , confirming substantial gradient conflict. Under P-SFT (prefix-conditioned), the average gradient cosine similarity shifts to approximately -0.003 to 0.09 , indicating that the conflict is effectively mitigated to near-orthogonality. These measurements are consistent with our theoretical prediction.

L Semantic Fidelity under Prefix Conditioning

To verify that prefix conditioning does not corrupt task semantics beyond the 100-sample human check reported in the main text, we evaluate prefix-conditioned inference on two additional domains: factual QA (GPQA-Diamond) and coding (LiveCodeBench). For each benchmark, we measure (i) the diversity gain induced by prefixing and (ii) the accuracy variance across three independent runs.

On GPQA-Diamond, prefix-conditioned inference increases response diversity by $+83.9\%$ while accuracy varies within $\pm 3.0\%$ across three runs. On LiveCodeBench, diversity increases by $+31.2\%$ while accuracy varies within $\pm 1.7\%$. Combined with the IF ACC = 1.00 and Format ACC = 1.00 results in Table 3, these measurements confirm that our semantically-neutral prefix design meaningfully diversifies the prior context without degrading task-level correctness across mathematical, factual, and programming domains.