

# Prompting Test-Time Scaling Is A Strong LLM Reasoning Data Augmentation

Sondos Mahmoud Bsharat, Zhiqiang Shen<sup>†</sup>

VILA Lab, MBZUAI

{Sondos.Bsharat, Zhiqiang.Shen}@mbzuai.ac.ae

## Abstract

Large language models (LLMs) exhibit strong reasoning when guided by chain-of-thought exemplars, yet collecting large, high-quality reasoning datasets remains laborious and resource-intensive. We introduce Prompting Test-Time Scaling (P-TTS), a prompt-space data augmentation framework for enhancing LLM reasoning via fine-tuning. In P-TTS, scaling refers to systematic expansion of the prompt space during offline teacher-data generation, not to increased inference-time compute for the deployed student. Rather than collecting thousands of examples, P-TTS starts from a small pool of 90 manually selected reasoning instances and applies principled instruction templates and paraphrased prompt variants to elicit diverse reasoning trajectories from a teacher model, producing a compact synthetic training set. We fine-tune Qwen-2.5 models of multiple sizes on the resulting data. On reasoning benchmarks including AIME25, MATH500, and GPQA-Diamond, P-TTS consistently improves accuracy over competitive small-data baselines such as S1 and S1.1 (1K-shot), with the largest gains on AIME25 while remaining strong on MATH500 and GPQA-Diamond. P-TTS also improves generalization on out-of-domain reasoning evaluations. Ablations show that exemplar diversity and prompt-space scaling are critical drivers of improvement, suggesting that prompt scaling explores the latent space of reasoning patterns, amplifying LLM problem-solving with minimal annotation overhead. P-TTS offers a practical, low-cost way to elicit strong LLM reasoning in resource-constrained or rapidly evolving domains. Our code and data are available at <https://github.com/VILA-Lab/PTTS>.

## 1 Introduction

Large language models (LLMs) (Radford et al., 2018; Achiam et al., 2023; Team et al., 2023;

Anthropic AI, 2025) attain strong deductive and quantitative reasoning once equipped with curated chains of thought (CoT) (Wei et al., 2022) or tool-augmented exemplars (Ma et al., 2024). However, constructing thousand-scale reasoning corpora is costly: it requires prompt engineering, human verification of multi-step solutions, and continuous refresh to track dataset shifts *in the wild*. Moreover, static large-shot prompts are brittle, i.e., fixed exemplars can inadvertently bias the model toward spurious solution templates or fail under domain shift, limiting generalization despite high in-domain scores. Prior work has largely scaled pre-/post-training time data (pre-training, instruction tuning, supervised CoT) or model size, while inference-time strategies typically vary only in decoding parameters (temperature, sampling) or rerank multiple outputs from a single prompt. The combinatorial space of which exemplars to show, how to order them, and how to perturb them remains mostly unexploited. We argue that the prompt itself is a stochastic control knob whose systematic scaling in prompt space during offline teacher-data generation can simulate the effect of larger reasoning datasets, without requiring direct collection of additional human-annotated samples.

In this work, we propose Prompting Test-Time Scaling (P-TTS) as an LLM post-training data augmentation method. In P-TTS, scaling refers to systematic expansion of the prompt space during offline teacher-data generation. Given a compact seed pool of 90 high-quality math reasoning exemplars, we algorithmically expand the prompt space during offline teacher-data generation by 1) exemplar subsampling under diversity constraints using various principled instructions (Bsharat et al., 2023), 2) ordering perturbations that modulate inductive biases (recency, primacy), and 3) pseudo-sampling of paraphrased rationales and solution skeletons via the model itself. Each seed exemplar question is paired with a prompt ensemble, a

<sup>†</sup>Corresponding author.

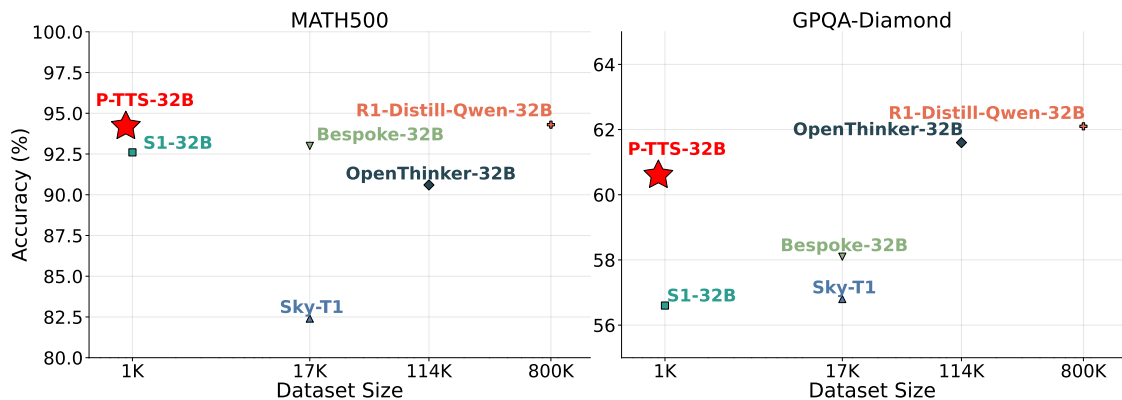


Figure 1: **Comparison of 32B-scale models on MATH500 (left) and GPQA-Diamond (right).** Each point represents a different model, with our P-TTS-32B (red star) showing competitive performance from a significantly smaller dataset. The x-axis scale highlights the differences in training data sizes across models.

set of independently constructed prompt contexts, whose answers are collected. This converts systematic prompt-space variation into a scalable data augmentation pipeline.

**Why can 90 beat 1K?** The key insight of our framework is that, conventionally, a fixed 1K-shot sample or prompt provides only one (or a few) points in the prompt-combinatorial space, whereas our P-TTS explores a far larger manifold of reasoning cues. From a bias-variance perspective, prompt ensembles for the same question reduce variance in reasoning trajectories and increase coverage of latent solution schemas. Information-theoretically, diverse promptings expose the model to a richer set of conditional priors over intermediate steps, effectively approximating a mixture-of-experts CoT without extra training. Empirically, we find that P-TTS surpasses the gap to 1K-shot baselines across reasoning and zero-shot generalization on out-of-domain tasks, and substantially improves robustness on naturally occurring, shifted test distributions.

We verify the effectiveness of P-TTS along two orthogonal dimensions: (1) *semantic/knowledge diversity* in CoT responses, which measures how P-TTS broadens the coverage of concepts and knowledge; and (2) *language trigram diversity*, which quantifies lexical and phrasing variation via distinct  $n$ -gram ratios and entropy, computed on both full responses and isolated reasoning traces.

The results show that reward framing yields the largest gains in lexical diversity, indicating stronger surface-level variation relative to the base examples. More broadly, we observe complementary effects: semantic/knowledge diversity helps avoid overfitting to narrow reasoning templates and improves

transfer; trigram diversity reduces lexical/template lock-in and strengthens robustness; ordering perturbations provide a modest reduction in position bias; and model-driven self-augmentation introduces novel yet still on-manifold rationales. Finally, scaling curves exhibit diminishing returns beyond roughly six prompt augmentations per base question, suggesting a practical deployment point.

**Contributions of this work.** (1) We introduce Prompting Test-Time Scaling, a simple yet effective prompt-space data augmentation framework for LLM reasoning. (2) We demonstrate that only 90 seed samples, when leveraged through our P-TTS, can outperform 1K-shot static prompts, reducing curation cost by an order of magnitude. (3) We provide empirical evidence that prompt-space exploration is an underutilized scaling dimension for LLM reasoning. Collectively, P-TTS reframes prompt design from a one-shot choice into a scalable augmentation process, unlocking robust reasoning without additional data collection or massive labeled datasets.

## 2 Related Work

**LLM Reasoning.** Frontier LLMs that expose long-form rationales (e.g., GPT-o1 (Jaech et al., 2024), Gemini (Team et al., 2023), Claude 3.7 Sonnet (Anthropic AI, 2025), DeepSeek-R1 (Guo et al., 2025)) have motivated RL or SFT pipelines that explicitly incorporate intermediate reasoning. In open-source settings (Muennighoff et al., 2025; Bespoke Labs, 2025; Guha et al., 2025), a teacher is prompted to produce CoT explanations, yielding (prompt, rationale, answer) triples to supervise a student (e.g., BESPOKE-STRATOS (Labs, 2025), OpenThinker (Guha et al., 2025)). While effective,

these approaches typically rely on tens to hundreds of thousands of exemplars. Our work targets a complementary regime: we exploit *instructional wrapping* to elicit diverse, high-utility rationales from only 90 seeds, yielding data efficiency competitive with 1K-scale baselines.

**Inference-Time Scaling.** Orthogonal to parameter or dataset scaling (Brown et al., 2020; Kaplan et al., 2020), inference-time strategies improve performance without updating model weights. Prior work scales decoding diversity (temperature, top- $k$ , nucleus sampling) (Holtzman et al., 2019; Fan et al., 2018; Meister et al., 2023), aggregates multiple generations from a fixed prompt (few-shot CoT (Wei et al., 2022), Self-Consistency (Wang et al., 2022)), or increases the *reasoning budget* (“think-more,” s1/s1.1) (Muennighoff et al., 2025). We build on this line by treating the *prompt itself* as a first-class scaling axis: P-TTS systematically varies instructional framing to create an ensemble of prompt contexts that can be aggregated at test time or distilled into compact training sets.

**Data Augmentation for Reasoning Tasks.** A complementary line synthesizes new samples or rationales to expand training corpora. META-MATH (Yu et al., 2023) bootstraps diverse math problems via generate-and-verify loops, while REASONINGMIX (Zheng et al., 2025) composes traces by interleaving steps across tasks. Such methods operate at the *data level* by creating novel items or reasoning sequences. In contrast, P-TTS operates at the *prompt level*: it preserves the original problems but injects controlled diversity through principle-guided wrappers (reward/penalty framing, correctness emphasis, step-by-step cues), which we show can be leveraged both at test time (ensemble prompting) and at training time (SFT over wrapper-elicited rationales).

**Principled instructions and prompt engineering.** General-purpose prompting frameworks catalog instruction patterns that improve reliability and adherence (Bsharat et al., 2023). Our approach instantiates a small subset compatible with math-style reasoning (reward, penalty, correctness, step-by-step) and formalizes them as deterministic wrap operators, yielding a semantically invariant augmentation space with clear ablation handles (template choice, placement, and paraphrase strength).

**Low-Resource Supervision.** Recent efforts show that carefully curated, small datasets can

deliver strong reasoning performance. S1 (Muennighoff et al., 2025) and LIMO (Ye et al., 2025) claim training on  $\sim 1\text{K}$  high-quality, challenging prompts as a competitive alternative to massive corpora. Our results complement these findings: with only 90 seeds, P-TTS converts principled prompt-space variation into supervision that matches or surpasses 1K-shot baselines, highlighting prompt-level scaling as a practical lever for low-resource regimes.

## 3 Methodology

### 3.1 Overview

We propose **Prompting Test-Time Scaling (P-TTS)**, a reasoning-centric data augmentation framework that expands a compact seed set via *instructional wrapping*. Rather than modifying task semantics, P-TTS applies a family of fixed textual wrappers (“principles”) (Bsharat et al., 2023) to each seed example, producing prompt variants that preserve the original problem while modulating the *instructional framing*. Concretely, a principle  $p$  is realized by a template  $\tau_p$  that deterministically wraps the raw question  $q$ . The union of the original prompts and their principle-conditioned variants forms a P-TTS *augmented corpus*. Unlike large-scale supervision or domain-specific curation, P-TTS relies on principle-guided prompt reformulation to elicit high-quality reasoning traces from a teacher model, which are then used for supervised fine-tuning (SFT) of a student (Appx. Fig. 4). In Sec. 4.3 we show that individual principles already yield measurable gains over the null prompt, and that training on the full P-TTS corpus (from only 90 seeds) can match or exceed models fine-tuned on substantially more data (Table 1).

We adopt a *seed-based* construction paradigm (Zhu et al., 2025): a small, vetted set of problems serves as the substrate for systematic instructional variation. Our seeds comprise  $N = 90$  AIME problems (2022–2024), selected for (i) **reasoning density** across algebra, combinatorics, number theory, geometry, and probability; (ii) **format and label reliability** (professionally authored items with definitive three-digit answers); and (iii) **contamination mitigation**: restricting to recent editions reduces overlap with widely scraped, earlier AIME corpora (Huang et al., 2025; Achiam et al., 2023). This compact, high-quality set enables controlled scaling via instructional wrappers without compromising semantic fidelity.

### 3.2 Selection of Instructional Principles

We instantiate four core principles:

$$\mathcal{P}_{\text{core}} = \{\text{Reward, Penalty, Correctness, StepByStep}\}, \quad (1)$$

chosen for their direct applicability to math reasoning and prior evidence of consistent gains across model families (Bsharat et al., 2023). Each principle  $p$  is bound to a fixed template  $\tau_p$  and applied as a wrap operator without modifying  $q$ ’s tokens:  $q_i^{(p)} = \tau_p \parallel q_i$ . This guarantees semantic invariance: removing the principled template  $\tau_p$  deterministically recovers question  $q_i$ . Table 9 in Appendix A summarizes the templates and induced operators.

### 3.3 P-TTS Dataset Construction

Our primary dataset is drawn from the AIME benchmarks (2022–2024) (Art of Problem Solving (AoPS)) and consists of  $N = 90$  unique problems with gold answers,  $\mathcal{O}_{\text{seed}} = \{(q_i, a_i)\}_{i=1}^N$ , where  $q_i$  is the original seed question without any additional prompting, and  $a_i \in \{0, 1, \dots, 999\}$  is the integer-style ground-truth for the associated question.  $\mathcal{O}_{\text{seed}}$  is the seed question-answer pairs.

**Original question (null prompt,  $p = \emptyset$ ).** We first query a teacher model  $T$  (DeepSeek-R1 (Guo et al., 2025)) on the unmodified question  $q_i^{(\emptyset)} = q_i$ . The teacher returns a reasoning trace  $r_i^{(\emptyset)}$  and a full response  $y_i^{(\emptyset)}$ , yielding

$$\mathcal{D}_{\text{seed}} = \{(q_i^{(\emptyset)}, y_i^{(\emptyset)}, r_i^{(\emptyset)}, a_i)\}_{i=1}^N. \quad (2)$$

**Selected/Core principle transformations.** For each principle  $p \in \mathcal{P}_{\text{core}}$ , implemented as a deterministic operator  $f_p(\cdot)$ , we *wrap* the original question with a fixed instructional template  $\tau_p$  while leaving the tokens of  $q_i$  unmodified:  $q_i^{(p)} = f_p(q_i) = \tau_p \parallel q_i$ . This preserves mathematical content, since removing  $\tau_p$  recovers the original question  $q_i$ . Querying  $T$  with each  $q_i^{(p)}$  produces  $(r_i^{(p)}, y_i^{(p)})$ , and we collect

$$\mathcal{D}_{\text{core}} = \{(q_i^{(p)}, r_i^{(p)}, y_i^{(p)}, a_i)\}_{i=1, \dots, N; p \in \mathcal{P}_{\text{core}}}. \quad (3)$$

**Reward framing variants for reward principle.** Our single-principle ablation (Sec. 4.3) shows that *Reward Framing* yields the largest gain. To test whether this effect depends on exact wording, we create six paraphrased Reward prompts that vary

in incentive magnitude, placement, and phrasing strength.<sup>1</sup>  $\mathcal{V}_{\text{Reward}} = \{R_1, R_2, R_3, R_4, R_5, R_6\}$ . Applying each Reward variant to all  $N$  problems and querying  $T$  gives :

$$\mathcal{D}_{\text{Reward}} = \{(\hat{q}_i^{(R_j)}, r_i^{(R_j)}, y_i^{(R_j)}, a_i^*)\}_{i=1}^N. \quad (4)$$

where  $R_j \in \mathcal{V}_{\text{Reward}}$  indexes the Reward variants,  $\hat{q}_i^{(R_j)}$  is the problem  $q_i$  wrapped with Reward variant  $R_j$ . Appendix A (Table 10) lists the Reward Framing paraphrases in full.

**Prompt order impact.** We also study how the ordering of prompts affects LLM responses and, in turn, their reasoning ability. Specifically, we experiment with placing the additional prompt either before (R1) or after (R4) the original question. In general, positioning prompts at the beginning leads the model to focus more on the prompt’s instructions and therefore achieves better accuracy, as shown in Appendix I (Table 14). Intriguingly, we further notice that when the question is relatively short, the accuracy difference is actually small.

**Dataset configurations and scale.** We construct four dataset families from the same  $N = 90$  seeds; full pseudocode for P-TTS data construction is provided in Appendix B.

- (i) **Single-P-TTS (per principle).** For each  $p \in \mathcal{P}_{\text{core}}$  we build a separate dataset  $\mathcal{D}_{(p)}^{\text{single}} = \{(q_i^{(p)}, r_i^{(p)}, y_i^{(p)}, a_i^*)\}_{i=1}^N$ , yielding four disjoint Single-P-TTS sets of size  $N = 90$  each. When we report “Single”, we train one model per  $p$  and report per-principle results or their mean (Sec. 4.3).
- (ii) **Core-P-TTS (union of singles).** The core set is the disjoint union over all four principles:  $\mathcal{D}_{\text{core}} = \bigsqcup_{p \in \mathcal{P}_{\text{core}}} \mathcal{D}_{(p)}^{\text{single}}$ , with  $|\mathcal{D}_{\text{core}}| = 4N = 360$ .
- (iii) **Seed combined with the core P-TTS.** We add the null-prompt seed set  $\mathcal{D}_{\text{seed}}$  to obtain  $\mathcal{D}_{\text{seed+core}} = \mathcal{D}_{\text{seed}} \cup \mathcal{D}_{\text{core}}$ , with  $|\mathcal{D}_{\text{seed+core}}| = 5N = 450$ .
- (iv) **Full P-TTS.** Let  $\mathcal{V}_{\text{Reward}}$  denote  $K$  reward paraphrases applied to all seeds, yielding

<sup>1</sup>Appendix I reports model accuracy across five Reward Framing variants (R1–R5) that differ in reward amount (e.g., \$20 vs. \$200,000) and location (e.g., beginning vs. end of prompt). Results are presented for o1-mini, Gemini, and DeepSeek.

$\mathcal{D}_{\text{Reward}}$ ; one variant is already included in  $\mathcal{D}_{\text{core}}$ , so the additional Reward portion has size  $(K - 1)N$ . The full corpus  $\mathcal{D}_{\text{full P-TTS}} = \mathcal{D}_{\text{seed}} \cup \mathcal{D}_{\text{core}} \cup \mathcal{D}_{\text{Reward}}$  therefore has size  $(1 + 4 + (K - 1))N$ . In our experiments  $K=6$ , so  $|\mathcal{D}_{\text{full P-TTS}}| = 10N = 900$ .

We parameterize the corpus size by the augmentation multiplier  $m := |\mathcal{D}|/N$ , where  $\mathcal{D}$  is the training corpus. In our study  $|\mathcal{D}| \in \{90, 360, 450, 900\}$  with  $N=90$ , so  $m \in \{1, 4, 5, 10\}$ , corresponding to Single, Core, Seed+Core, and Full, respectively, enabling controlled comparisons as a function of prompt diversity (Fig. 2).

### 3.4 Fine-Tuning with P-TTS Dataset Augmentations

We evaluate whether principle-guided wrapping improves supervised reasoning under constrained data. We fine-tune Qwen2.5-Instruct (7B/14B/32B) (Yang et al., 2025) separately on each configuration from Sec. 3.3, following an SFT recipe adapted from s1 (Muennighoff et al., 2025). The student is trained to predict full assistant outputs (reasoning trace and final answer) with token-level cross-entropy computed only on assistant tokens. Each dataset configuration (Original, Single, Core, Mix, and Full P-TTS) is used to train a separate model, enabling us to isolate the contribution of each prompting strategy and the effect of data scaling, i.e., isolating the impact of instructional diversity and corpus scale ( $m \in \{1, 4, 5, 10\}$ ) while keeping optimization and decoding fixed across runs. Our dataset scale (90  $\rightarrow$  900 examples) is intentionally small, allowing us to directly measure how principle-guided prompt reformulations affect supervised reasoning performance relative to models trained on much larger datasets. See Appendix N, O, and P for qualitative examples.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** We evaluate P-TTS on public reasoning benchmarks covering math and science: **AIME25** (AIME, 2025), **MATH500** (Hendrycks et al., 2021), and **GPQA-Diamond** (GPQA-D) (Rein et al., 2024). We additionally assess cross-domain and multilingual generalization on a broader suite (Table 4), including **Gaokao**, **Kaoyan**, **OlympiadBench** (He et al., 2024), **AMC23**, **GradeSchoolMath**, and **Min-**

**erva**. To further assess transfer beyond mathematics and science benchmarks to a qualitatively different reasoning domain, we additionally evaluate on **LegalBench** (Guha et al., 2023) using three tasks spanning evidence reasoning (**hearsay**), civil procedure (**personal jurisdiction**), and contract interpretation (**jcrew blocker**). We evaluate using lm-evaluation-harness (Gao et al., 2021; Biderman et al., 2024); benchmark details are provided in Appendix G.

**Baselines.** We benchmark P-TTS against three categories of reasoning models. (i) *Closed-source (API-only) models*: OpenAI’s o1 series (OpenAI, 2024, 2025) and Google’s experimental Gemini 2.0 Flash Thinking variant (Cloud, 2024). (ii) *Open-weight models*: DeepSeek-R1 series (Guo et al., 2025) and Qwen’s QwQ-32B-preview (Qwen Team, 2024; Yang et al., 2025). (iii) *Open-weight SFT models on Qwen2.5-Instruct with public data on openly available reasoning corpora*: including Bespoke-Stratos-32B (Bespoke Labs, 2025), OpenThinker-32B (Team, 2025b; Guha et al., 2025), Sky-T1-32B-Preview (Team, 2025a), and the S1/ S1.1 w/o BF (Muennighoff et al., 2025) checkpoints.

**Diversity Metrics.** We compute two complementary metrics—*semantic* and *surface-level*—to quantify how each single-principle variant in  $\mathcal{D}_{\text{core}}$  adds information beyond the seed set  $\mathcal{D}_{\text{seed}}$ . **Semantic diversity (Diversity Gain).** Following (Yu et al., 2023; Bilmes, 2022), we compute *diversity gain* to measure knowledge-level novelty via embedding distances, using OpenAI’s text-embedding-ada-002 as the feature extractor; higher values indicate greater semantic divergence from the seed data. **Surface-level diversity (trigram diversity).** Following (Li et al., 2022), we compute *trigram diversity* as the ratio of non-overlapping trigrams between two texts, averaged between each principle-augmented instance in  $\mathcal{D}_{\text{core}}$  and its corresponding baseline instance in  $\mathcal{D}_{\text{seed}}$ . Full metric definitions are provided in Appendix F.

### 4.2 Teacher Model for Data Construction

To construct a **compact yet high-quality** mathematical reasoning corpus, we compared three candidate teachers that expose chain-of-thought rationales via their public APIs: Claude, OpenAI O4-mini-high, and DeepSeek-R1. For each teacher, we collected a small set of 90 answer–reasoning pairs—reflecting our goal of iden-

Model	# Train Size	AIME25	MATH500	GPQA-Diamond	Avg
<i>closed-source models</i>					
o1-preview (OpenAI, 2025)	–	–	85.50	78.30	–
o1-mini (OpenAI, 2025)	–	–	90.00	60.00	–
Gemini 2.0 Flash Think	–	–	–	–	–
<i>open-source models</i>					
Qwen2.5-32B-Instruct (Yang et al., 2024)	–	–	84.00	49.00	–
QwQ-32B (Team, 2025c)	–	–	90.60	54.50	–
DeepSeek-R1 (Guo et al., 2025)	≫800K	–	<b>97.30</b>	<b>71.50</b>	–
DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025)	800K	–	94.30	62.10	–
<i>Open-weight &amp; open-data SFT on Qwen2.5-Instruct</i>					
OpenThinker-32B (Team, 2025b)	114K	53.33	90.60	61.60	68.50
Bespoke-32B (Bespoke Labs, 2025)	17K	–	93.00	58.10	–
Sky-T1-32B-Preview (Team, 2025a)	17K	–	82.40	56.80	–
S1-32B (Muennighoff et al., 2025)	1K	26.70	92.60	56.60	58.63
S1.1-32B (Muennighoff et al., 2025)	1K	50.00	<b>94.40</b>	<b>60.60</b>	68.33
<b>P-TTS-32B (Ours)</b>	<b>90 → 900</b>	<b>53.33</b>	<u>94.20</u>	<b>60.61</b>	<b>69.38</b>

Table 1: **Accuracy comparison of 32B-scale models** on AIME25, MATH500, and GPQA-Diamond. Models are grouped into closed-source APIs, open-source baselines, and open-weight fine-tuned variants of Qwen2.5-Instruct. Our method, **P-TTS-32B**, leverages only 90 seed examples augmented to generate up to 900 training examples. *Notes:* Results for *Gemini* and *Qwen* are taken from (Muennighoff et al., 2025) (we follow their evaluation settings).

tifying the smallest effective training corpus—and fine-tuned Qwen2.5-7B-Instruct on the resulting data. As detailed in Appendix H, DeepSeek-R1-based supervision yields the highest downstream accuracy across AIME25, MATH, and GPQA-Diamond, so we adopt **DeepSeek-R1** as the teacher for all subsequent experiments.

### 4.3 Ablation Studies

#### 4.3.1 Single-P-TTS: Measuring the Impact of Each Principle

**Single-P-TTS.** To assess each principle independently, we fine-tune separate Qwen2.5-7B-Instruct models on **Single-P-TTS** subsets (90 examples each), where only one core principle  $p \in \mathcal{P}_{\text{core}}$  is applied. We compare against P-TTS<sub>Seed</sub>, trained on the same seeds without instructional framing. As shown in Table 5, P-TTS<sub>Reward</sub> yields the strongest overall improvement, increasing average accuracy from 31.55% to 37.12% (+5.57 percentage points) and improving across all benchmarks. P-TTS<sub>Correctness</sub> and P-TTS<sub>Think</sub> also improve the average. P-TTS<sub>Penalty</sub> boosts MATH500 and GPQA-D but reduces AIME25, resulting in a slightly lower average than the baseline. These findings highlight that even minimal augmentations, i.e., just 90 principle-guided examples, can yield measurable improvements; among the four principles, Reward

is the most effective when applied in isolation.

**Pairwise-P-TTS.** To further evaluate the usefulness of each principle and whether they can be effectively combined, we study pairwise mixtures anchored on the strongest single-principle variant. Specifically, since P-TTS<sub>Reward</sub> performs best in the single-principle ablation, we fix Reward and add one additional principle (180-example datasets). Table 5 shows that all pairwise combinations improve average accuracy over Reward alone, with P-TTS<sub>Reward∪Correctness</sub> achieving the best overall result (44.26% average accuracy) and improving all three benchmarks. In contrast, Reward∪Think yields the smallest gain (39.32% average accuracy), improving only MATH500 while leaving AIME25 unchanged and slightly reducing GPQA-D.

**Core-P-TTS.** To assess the relative importance of each principle in combination, we perform a leave-one-out ablation over the full Core P-TTS dataset  $\mathcal{D}_{\text{core}}$  (360 examples). We fine-tune Qwen2.5-7B-Instruct on the full core set and then re-train after removing one principle at a time (270 examples). As shown in Appendix K (Table 15), removing Reward yields the largest drop in average accuracy (45.42% → 41.65%), while excluding Correctness or Penalty leads to smaller degradations. Removing Think slightly increases

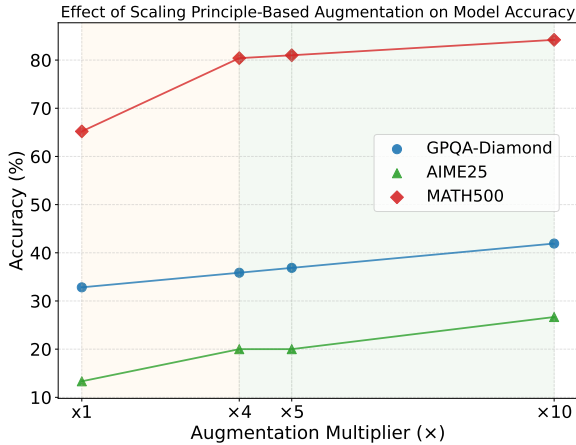


Figure 2: **Accuracy improvement with increased principled augmentation on 7B model.** We evaluate how model accuracy scales with the number of augmented training examples. Here,  $\times 1$  refers to  $P\text{-TTS}_{\text{Reward}}$  (90 examples),  $\times 4$  to  $P\text{-TTS}_{\text{Core}}$  (360 examples),  $\times 5$  to  $P\text{-TTS}_{\text{Core}+\text{Orig}}$  (450 examples), and  $\times 10$  to  $P\text{-TTS}_{\text{Full}}$  (900 examples). It is observed that accuracy improves consistently across all evaluation sets with larger, principle-guided augmentations.

the average despite improving AIME25, suggesting that its benefit is not consistent across benchmarks in the core mixture. Fig. 3 further supports this trend, showing that adding principles generally improves average accuracy.

### 4.3.2 Measuring the Effects of Augmentation Size

We study how scaling the P-TTS training set affects performance (Appendix L, Table 8). We compare: (i)  $P\text{-TTS}_{\text{Core}}$  (360 examples), (ii)  $P\text{-TTS}_{\text{Core}+\text{Seed}}$  (450), and (iii)  $P\text{-TTS}_{\text{Core}+\text{Seed}+\text{RewardVar}}$  (900; full set). Accuracy improves with dataset size: the full 900-example model reaches 50.93% average accuracy, outperforming both lower-data P-TTS configurations (360 and 450 examples) and the 1K-shot S1.1 baseline (47.54%). The largest gain comes from adding Reward variants (450  $\rightarrow$  900), yielding a +4.97% gain and improving AIME25, MATH500, and GPQA-D. Fig. 2 shows the same scaling trend as augmentation multiplier increases.

### 4.4 Comparison with Temperature Sampling

To isolate the effect of prompt diversity, we introduce a matched **Temperature Sampling** baseline that increases sampling from the teacher while keeping the prompt fixed. For each of the 90 seed problems, we sample 10 completions at temperature 0.7 using a single prompt, yielding 900 total

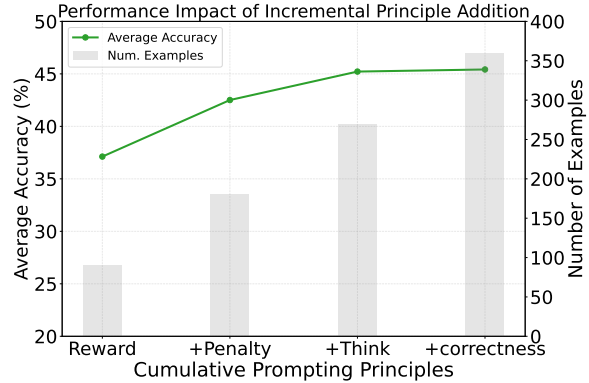


Figure 3: **Impact of incremental principle addition on average accuracy.** As prompting principles are cumulatively added (Reward  $\rightarrow$  +Penalty  $\rightarrow$  +Think  $\rightarrow$  +Correctness), both the number of training examples (bars, right axis) and the average accuracy across benchmarks (line, left axis) increase, illustrating the compounding benefit of principled augmentation.

teacher generations. This setup matches full P-TTS in the number of teacher calls and generation limits, but removes prompt variation. All other decoding settings are kept identical to P-TTS, with unspecified parameters left at their default values. As shown in Table 6, P-TTS consistently outperforms Temperature Sampling, achieving higher average accuracy (50.93% vs. 46.76%). These results suggest that the gains from P-TTS come not only from increased sampling, but from diversity induced by varying the prompt.

Model	AIME25	MATH500	GPQA-D	Avg.
S1-7B	13.33	77.20	41.41	43.98
S1.1-7B	20.00	81.20	41.41	47.54
TS-7B	20.00	82.40	37.88	46.76
<b>P-TTS-7B (Ours)</b>	<b>26.67</b>	<b>84.20</b>	<b>41.92</b>	<b>50.93</b>

Table 6: **Comparison with a Temperature Sampling baseline.** TS denotes Temperature Sampling with 900 total teacher generations. GPQA-D denotes GPQA-Diamond.

## 5 P-TTS Dataset Analysis

**Diversity.** Table 5 shows that P-TTS variants trained with principled instructional wrapping generally outperform the seed baseline, with  $P\text{-TTS}_{\text{Reward}}$  giving the strongest and most consistent gains (31.55%  $\rightarrow$  37.12%).  $P\text{-TTS}_{\text{Correctness}}$  provides smaller gains, while  $P\text{-TTS}_{\text{Think}}$  is most beneficial on AIME25 and GPQA-D.  $P\text{-TTS}_{\text{Penalty}}$  shifts performance toward M500 and GPQA-D, with weaker performance on AIME25,

Model	Hearsay	Personal_Jurisdiction	Jcrew_Blocker	Avg.
S1-7B	60.6	47.9	44.44	50.98
S1.1-7B	46.8	46.0	<b>62.96</b>	51.92
<b>P-TTS-7B (Ours)</b>	<b>61.70</b>	<b>52.00</b>	48.15	<b>53.95</b>

Table 2: **Zero-shot generalization accuracy (%) on LegalBench tasks.** P-TTS-7B achieves the best average performance across three legal reasoning tasks.

Prompt Variant	Responses $\uparrow$	Reasoning $\uparrow$
Reward Framing	<b>0.8363</b>	<b>0.9280</b>
Correctness Framing	0.8227	0.9264
Penalty Framing	0.8223	0.9266
Step-by-Step Thinking	0.8295	0.9254

Table 3: **Trigram diversity of teacher outputs under different prompting strategies.** We report average lexical diversity ( $\uparrow$ ) for full responses and isolated reasoning traces. It is observed that reward framing yields the highest diversity, indicating stronger surface-level variation from original examples.

leaving the overall average close to the baseline. These trends align with Appendix J (Fig. 6), where variants with higher semantic diversity (Diversity Gain)—most prominently Reward—also exhibit larger accuracy improvements. Table 3 further shows that Reward achieves the highest trigram diversity for both full responses and reasoning traces, consistent with prior evidence that diversity-promoting augmentations improve robustness and transfer (Bukharin et al., 2024; Qu et al., 2020).

**Prompting Impact on Output Length.** To further understand how each principle influences model behavior, we compare the token lengths of generated responses and reasoning traces (as shown in Appendix M, Table 16). We observe that Reward framing consistently produces longer and more detailed reasoning outputs, which aligns with its observed accuracy gains. While longer generations may imply increased inference costs, the improved reasoning quality may justify this tradeoff in high-stakes tasks.

**Teacher Generation Cost.** To assess the practical overhead of P-TTS, we analyze teacher token usage and the corresponding API cost across prompting variants (as shown in Appendix M, Tables 16 and 7). Across prompting variants, the token budget is dominated by long completion and reasoning traces, whereas prompt tokens contribute only a small fraction of the overall cost. At the scale of our experiments, the total teacher-generation cost is 1.53 USD for Core-P-TTS and 3.96 USD for Full P-TTS. These results suggest that, for compact

seed pools, the additional supervision provided by P-TTS can be obtained at a modest one-time cost.

## 5.1 Main Results

**Overall Performance Comparison.** We compare our P-TTS-32B model, trained on only 90  $\rightarrow$  900 principle-augmented examples, against a diverse set of baselines in Table 1 and Fig. 1, including closed-source APIs, open-source models, and open-weight SFT models built on Qwen2.5-32B-Instruct. Among open-weight and open-data SFT baselines on Qwen2.5-32B-Instruct, P-TTS-32B achieves comparable or better performance on AIME25 (53.33%) and slightly exceeds the best score among the compared open-weight baselines on GPQA-DIAMOND (60.61%), while remaining competitive on MATH500 (94.20%). Overall, P-TTS-32B reaches **69.38%** average accuracy, exceeding the compared open-weight baselines, and narrowing the gap to large-scale open-source models such as DeepSeek-R1 that rely on  $\gg$ 800K training examples. These results highlight the efficiency of principled instructional augmentation and show that strong reasoning performance can be attained with lightweight, targeted supervision.

**Cross-Domain and Multilingual Generalization.** Although P-TTS is trained only on AIME22–24 English seed problems (90  $\rightarrow$  900 via wrappers), it transfers zero-shot to out-of-domain benchmarks spanning Chinese exams (Gaokao, Kaoyan), U.S. math (OlympiadBench, AMC23, Grade-SchoolMath), and scientific quantitative reasoning (Minerva). As shown in Table 4, P-TTS achieves the best average accuracy (60.2%) and is competitive on most individual tasks, despite using orders of magnitude less supervision than prior SFT corpora. This suggests that principled prompt augmentation (varying instructional framing and exemplar structure) promotes robustness beyond the AIME distribution rather than overfitting to a single benchmark family.

Model	OlympiadBench	Gaokao	Kaoyan	Minerva	GradeSchool	AMC23	Avg.
OpenAI-o1-preview	52.1	62.1	51.5	47.1	<b>62.8</b>	81.8	59.6
Qwen2.5-32B-Instruct	45.3	<b>72.1</b>	48.2	41.2	56.7	64.0	54.6
OpenThoughts (114K)	56.3	63.2	54.7	41.1	39.0	80.5	55.8
NuminaMath (100K)	36.7	49.4	32.7	24.6	36.2	40.6	36.7
S1 (1K)	56.9	32.9	<b>59.3</b>	46.7	61.4	77.5	55.8
<b>P-TTS (Ours)</b>	<b>63.9</b>	51.9	52.3	<b>51.5</b>	53.8	<b>87.5</b>	<b>60.2</b>

Table 4: **Zero-shot generalization accuracy (%) on out-of-domain reasoning benchmarks.** P-TTS is trained only on AIME22–24–derived data (90 seeds  $\rightarrow$  900 augmented prompts).

Model	AIME25	M500	GPQA-D	Avg.
<b>Single-principle P-TTS (90 ex.)</b>				
P-TTS <sub>Seed</sub> (baseline)	6.67	60.20	27.78	31.55
P-TTS <sub>Reward</sub>	<b>13.33</b>	<b>65.20</b>	<b>32.83</b>	<b>37.12</b>
P-TTS <sub>Correctness</sub>	6.67	61.80	28.28	32.25
P-TTS <sub>Penalty</sub>	0.00	63.20	30.30	31.17
P-TTS <sub>Think</sub>	<b>13.33</b>	58.20	30.81	34.11
<b>Pairwise P-TTS Centered on Reward (180 ex.)</b>				
P-TTS <sub>Reward<math>\cup</math>Penalty</sub>	<b>20.00</b>	75.20	32.32	42.51
P-TTS <sub>Reward<math>\cup</math>Correctness</sub>	<b>20.00</b>	<b>75.40</b>	<b>37.37</b>	<b>44.26</b>
P-TTS <sub>Reward<math>\cup</math>Think</sub>	13.33	72.80	31.82	39.32

Table 5: **Accuracy (%) of single- and pairwise-principle P-TTS ablations using the 7B model.** M500 denotes MATH500, and GPQA-D denotes GPQA-Diamond.

**Generalization to Legal Reasoning.** To further assess generalization beyond mathematics and science benchmarks, we evaluate P-TTS-7B on three **legal reasoning** tasks from **LegalBench—hearsay, personal jurisdiction, and jcrew blocker** (Guha et al., 2023). As shown in Table 2, P-TTS-7B achieves the best average accuracy (53.95%), outperforming both S1-7B and S1.1-7B. It performs best on two of the three tasks (**hearsay** and **personal jurisdiction**) and remains competitive on **jcrew blocker**. These results provide further evidence that the benefits of P-TTS extend beyond mathematics to legal reasoning.

**Scaling across model sizes.** We evaluate P-TTS across 7B, 14B, and 32B Qwen2.5 variants (Table 12 in Appendix). At the 7B scale, P-TTS outperforms S1 and S1.1 (Muennighoff et al., 2025) on all three benchmarks (e.g., 26.67% vs. 20.00% on AIME25, 84.20% vs. 81.20% on MATH500). This trend largely persists with scale: at 32B scale, P-TTS substantially improves AIME25 (53.33%), is essentially tied on GPQA-DIAMOND (60.61%), and remains within 0.2 points of S1.1 on MATH500 (94.20% vs. 94.40%). These results highlight the robustness and efficiency of principled instruction tuning (P-TTS), demonstrating that even with minimal data (**90**  $\rightarrow$  **900**), it scales effectively and consistently enhances perfor-

mance across diverse benchmarks.

## 6 Conclusion

We presented **Prompting Test-Time Scaling (P-TTS)**, a lightweight yet effective framework that converts a compact seed set into a high-utility reasoning corpus by wrapping each problem with principled instructional prompts. Without changing task semantics, P-TTS systematically explores prompt-space via reward/penalty framing, correctness emphasis, and step-by-step guidance, eliciting high-quality rationales from a teacher model to supervise a student. Across augmentation multipliers  $m \in \{1, 4, 5, 10\}$ , P-TTS consistently improves over the null prompt under tight data budgets, starting from only 90 seeds, and it transfers to out-of-domain and multilingual reasoning benchmarks.

**Our Future Work.** The promising directions include adaptive per-instance wrapper selection, integration with retrieval and verifier/reranker pipelines, scheduling wrapper mixtures over training (e.g., curriculum-style), and studying wrapper transfer across tasks, languages, and modalities.

## 7 Limitations

There are several potential limitations. 1) Our *training seeds* are concentrated on math-style problems. While we evaluate cross-domain and multilingual transfer on additional benchmarks, broader external validity to open-ended generation, long-horizon reasoning, and multimodal settings remains an important direction for future work.

2) P-TTS depends on a single teacher model to generate rationales, any bias, error, or stylistic artifact in the teacher can be amplified by our wrappers and propagated to the student, especially since rationales were not human-audited.

3) While wrappers are designed to be semantically invariant, some templates (e.g., extreme reward/penalty framings) may shift reasoning behavior in undesired ways and could introduce ethical or

calibration issues. Sensitivity to wrapper mixture, placement, and decoding settings also suggests latent hyperparameter fragility. P-TTS may also provide limited gains when wrapper-based prompt variation fails to elicit meaningfully different reasoning trajectories, in which case it may offer little advantage over simpler repeated sampling from a fixed prompt.

4) As with most public benchmarks, residual pretraining exposure to AIME material cannot be conclusively ruled out, despite our contamination-mitigation steps. Finally, P-TTS requires compute at data collection time through inference generation (via wrapper ensembles).

## Ethics Statement

This work studies Prompting Test-Time Scaling (P-TTS) as a low-cost way to synthesize chain-of-thought style reasoning data from a small, manually curated seed set. While this can reduce annotation burden and broaden access to reasoning-capable models, it also raises ethical risks common to synthetic-data fine-tuning. For instance, teacher-generated rationales may contain subtle errors, hallucinated steps, or biased framing; scaling prompt variants can amplify such artifacts and potentially teach models to produce persuasive but incorrect explanations. To mitigate this, we emphasize accuracy-focused evaluation on standard reasoning benchmarks, report ablations that isolate when gains come from diversity vs. sampling schedules, and encourage practitioners to incorporate verification signals (e.g., final-answer checking) when applying P-TTS in high-stakes settings. Overall, we view P-TTS as an efficiency-oriented methodology for LLM reasoning whose benefits come from careful data augmentation, verification, and responsible preparation practices. Generative AI models were only used for language polishing.

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AIME. 2025. [2025 aime i](#). Art of Problem Solving Wiki. Held February 6, 2025.

Anthropic AI. 2025. [Claude 3.7 sonnet and claude code](#). Anthropic blog. First hybrid reasoning large language model generally available.

Art of Problem Solving (AoPS). Aime problems and solutions. [https://artofproblemsolving.com/wiki/index.php/AIME\\_Problems\\_and\\_Solutions](https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions).

Bespoke Labs. 2025. Bespoke-stratos-32b. <https://huggingface.co/bespokelabs/Bespoke-Stratos-32B>. Hugging Face model card, Apache-2.0 license. Fine-tuned Qwen2.5-32B-Instruct on Bespoke-Stratos-17k dataset derived via DeepSeek-R1 distillation.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, and 1 others. 2024. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*.

Jeff Bilmes. 2022. Submodularity in machine learning and artificial intelligence. *arXiv preprint arXiv:2202.00132*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2023. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4. *arXiv preprint arXiv:2312.16171*.

Alexander Bukharin, Jiachang Mu, Zhengbao Zhang, Seyeon Lee, Kai-Wei Chang, Noah A. Smith, and Daniel Khashabi. 2024. [Data diversity matters for robust instruction tuning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2871–2885.

Google Cloud. 2024. Flash thinking with generative ai. <https://cloud.google.com/vertex-ai/generative-ai/docs/thinking>.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Samuel Weinbach, and Connor Leahy. 2021. [EleutherAI/Im-evaluation-harness: Evaluation Harness for Language Models](#).

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, and 1 others. 2025. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters,

- Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279.
- 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*.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. **OlympiadBench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems**. *arXiv preprint arXiv:2402.14008*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.
- Shulin Huang, Linyi Yang, Yan Song, Shuang Chen, Leyang Cui, Ziyu Wan, Qingcheng Zeng, Ying Wen, Kun Shao, Weinan Zhang, and 1 others. 2025. Thinkbench: Dynamic out-of-distribution evaluation for robust llm reasoning. *arXiv preprint arXiv:2502.16268*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- Bespoke Labs. 2025. **Bespoke-stratos-17k: A synthetic reasoning dataset of questions, reasoning traces, and answers**. Hugging Face Dataset. Derived from DeepSeek-R1 via the Sky-T1 pipeline using Bespoke Curator.
- Wenhao Li, Xiaoyuan Yi, Jinyi Hu, Maosong Sun, and Xing Xie. 2022. **Evade the trap of mediocrity: Promoting diversity and novelty in text generation via concentrating attention**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10834–10858, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, and Aixin Sun. 2024. Sciagent: Tool-augmented language models for scientific reasoning. In *EMNLP*.
- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. 2025. **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.
- OpenAI. 2024. **Learning to reason with llms**.
- OpenAI. 2025. Openai o3-mini. <https://openai.com/index/openai-o3-mini/>.
- Yanru Qu, Dinghan Shen, Yelong Shen, Sandra Sajeew, Jiawei Han, and Weizhu Chen. 2020. Coda: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding. *arXiv preprint arXiv:2010.08670*.
- Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown. <https://qwenlm.github.io/blog/qwq-32b-preview/>. QwQ-32B-Preview is an experimental reasoning model with open weights.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- NovaSky Team. 2025a. **Sky-t1: Fully open-source reasoning model with o1-preview performance in \$450 training cost**.
- OpenThoughts Team. 2025b. **Openthinker-32b**. <https://huggingface.co/open-thoughts/Openthinker-32B>.
- Qwen Team. 2025c. Qwq-32b: Embracing the power of reinforcement learning.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Tong Zheng, Lichang Chen, Simeng Han, R Thomas McCoy, and Heng Huang. 2025. Learning to reason via mixture-of-thought for logical reasoning. *arXiv preprint arXiv:2505.15817*.

Alan Zhu, Parth Asawa, Jared Quincy Davis, Lingjiao Chen, Boris Hanin, Ion Stoica, Joseph E Gonzalez, and Matei Zaharia. 2025. Bare: Leveraging base language models for few-shot synthetic data generation. *arXiv preprint arXiv:2502.01697*.

# Appendix

## Contents

<b>A Instructional Wrappers and Reward-Framing Variants</b>	<b>13</b>
<b>B P-TTS Data Construction Algorithm</b>	<b>13</b>
<b>C Extended Experiments on <math>\mathcal{D}_{\text{core}}</math></b>	<b>13</b>
<b>D Scaling Across Model Sizes</b>	<b>13</b>
<b>E Training Details</b>	<b>13</b>
<b>F Diversity Metric Definitions</b>	<b>13</b>
<b>G Evaluation Details</b>	<b>14</b>
<b>H Additional Results on Other Models</b>	<b>14</b>
<b>I Evaluation of Reward Framing Variants</b>	<b>15</b>
<b>J Semantic and Lexical Diversity Analyses</b>	<b>15</b>
<b>K Core-P-TTS Leave-One-Out Ablation</b>	<b>15</b>
<b>L Data-Volume Ablation</b>	<b>15</b>
<b>M Teacher Output Length and Cost Across Prompt Variants</b>	<b>15</b>
<b>N Example from <math>\mathcal{D}_{\text{full-P-TTS}}</math></b>	<b>15</b>
<b>O P-TTS-32B vs. S1-32B</b>	<b>15</b>
<b>P Example of P-TTS-32B</b>	<b>15</b>

## A Instructional Wrappers and Reward-Framing Variants

Table 9 summarizes the core P-TTS instructional wrappers used to construct principle-guided prompts. Table 10 lists the Reward-framing phrases used in our P-TTS experiments.

## B P-TTS Data Construction Algorithm

Algorithm 1 summarizes the details of full P-TTS data construction procedure used throughout the paper, starting from the 90 AIME seeds and applying core principles and reward variants to query the teacher model.

## C Extended Experiments on $\mathcal{D}_{\text{core}}$

We further evaluate models trained on the  $\mathcal{D}_{\text{core}}$  dataset across multiple parameter scales. Table 11 reports grouped results for 7B, 14B, and 32B variants, highlighting consistent gains from principled data augmentation.

## D Scaling Across Model Sizes

Table 12 reports P-TTS performance across 7B, 14B, and 32B model variants compared to S1 and S1.1 on four reasoning benchmarks. This expands the main-text discussion in Sec. 5.1 and shows that P-TTS consistently matches or exceeds S1/S1.1 at all scales under a much smaller training budget.

## E Training Details

We fine-tuned the Qwen2.5-Instruct family at three scales—7B, 14B, and 32B—using our P-TTS datasets. Training dynamics are shown in Fig. 5. All models were trained for 5 epochs with an effective global batch size of 16 (micro-batch size of 1 with gradient accumulation). We used the AdamW optimizer ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.95$ , weight decay =  $1 \times 10^{-4}$ ) and a base learning rate of  $1 \times 10^{-5}$ , warmed up linearly for the first 5% of steps and decayed to zero following a cosine schedule. Training was conducted in bf16 precision with fully sharded data parallelism (FSDP) enabled. We set the maximum sequence length to 20k tokens to avoid truncation of reasoning traces. For supervision, loss was applied only to the reasoning and answer tokens, not the input question text. Across model scales, this consistent setup allowed us to directly compare how principled data augmentation transfers to different parameter sizes.

## F Diversity Metric Definitions

**Semantic diversity (Diversity Gain).** Following (Yu et al., 2023; Bilmes, 2022), given a seed set  $\mathcal{D}_{\text{seed}}$  and an augmented set  $\mathcal{D}_{\text{core}}$ , we compute

$$\text{DG} = \frac{1}{M} \sum_{x_i \in \mathcal{D}_{\text{core}}} \min_{x_j \in \mathcal{D}_{\text{seed}}} \|f(x_i) - f(x_j)\|_2^2,$$

where  $f(\cdot)$  is an embedding function and  $M = |\mathcal{D}_{\text{core}}|$ . We use OpenAI text-embedding-ada-002 for  $f(\cdot)$ .

**Surface-level diversity (trigram diversity).** Following (Li et al., 2022), for texts  $x, y$  we define

$$\text{TD}(x, y) = 1 - \frac{|\text{Tri}(x) \cap \text{Tri}(y)|}{|\text{Tri}(x) \cup \text{Tri}(y)|},$$

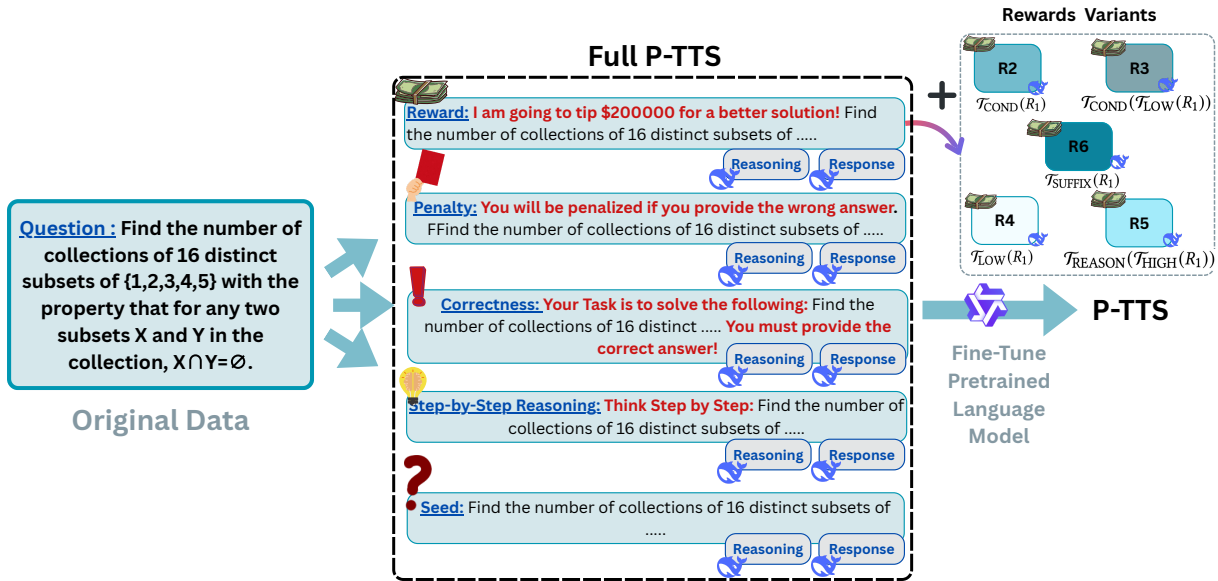


Figure 4: Overview of the P-TTS data augmentation process. Starting from a small set of high-quality math problems, we generate diverse prompt variants through instruction reframing. These augmented prompts are used to elicit high-quality LLM completions, which are then collected as synthetic reasoning data to fine-tune.

where  $\text{Tri}(x)$  denotes the set of distinct word-level trigrams in  $x$ . We average TD over pairs between each principle variant in  $\mathcal{D}_{\text{core}}$  and its corresponding instance in  $\mathcal{D}_{\text{seed}}$ .

## G Evaluation Details

**Benchmarks.** We evaluate our P-TTS models on three public reasoning benchmarks: **AIME25** (AIME, 2025) from the American Invitational Mathematics Examination; AIME includes problems from algebra, arithmetic, geometry, number theory, combinatorics, and probability. **MATH500** (Hendrycks et al., 2021) is a 500-problem competition-math subset; we adopt the publicly released OpenAI selection used in prior work. **GPQA-Diamond** (Rein et al., 2024) contains 198 PhD-level science questions from Biology, Chemistry, and Physics with reported expert performance of 69.7%. We evaluate using the lm-evaluation-harness framework (Gao et al., 2021; Biderman et al., 2024). In addition to these three core benchmarks, we further assess cross-domain and multilingual generalization using a broader set of reasoning tasks spanning Chinese exams, U.S. school math, olympiad-style problems, and scientific quantitative reasoning. These evaluations, shown in Table 4, include **Gaokao**, **Kaoyan**, **OlympiadBench** (He et al., 2024), **AMC23**, **GradeSchoolMath**, and **Minerva**. o further assess transfer to a qualitatively differ-

ent reasoning domain, we additionally evaluate on three tasks from **LegalBench** (Guha et al., 2023), a benchmark of legal reasoning tasks. Specifically, we use **hearsay**, which asks whether a piece of evidence counts as hearsay for a given issue; **personal jurisdiction**, which asks whether a court in a given forum may exercise personal jurisdiction over a defendant; and **jcrew blocker**, which asks whether a particular loan clause is a J.Crew blocker provision. These results are reported in Table 2.

**Protocol and metric.** To make results comparable across models and ablations, we disable sampling by setting the temperature to 0 (greedy decoding), so each input yields a deterministic output. Reported scores are accuracy (equivalent to pass@1).

## H Additional Results on Other Models

To justify our choice of DeepSeek-R1 as the teacher for P-TTS data construction, we compare several teacher configurations under a fixed small-data budget. For each row in Table 13, we generate the indicated number of answer–reasoning pairs from the corresponding teacher or hybrid setup and fine-tune a Qwen2.5-7B-Instruct student on that corpus. DeepSeek-based supervision (with or without hybrids) consistently outperforms Claude and O4-mini-high alone, and pure DeepSeek-R1 supervision delivers the best overall trade-off across AIME24/25, GPQA-Diamond, and MATH, moti-

Dataset Setting	#Calls	Total Cost (USD)
Core P-TTS (4 principles)	360	1.5289
Full P-TTS	900	3.9569

Table 7: Total teacher-generation cost for dataset construction.

---

**Algorithm 1:** Prompting Test-Time Scaling (P-TTS) Dataset Construction

---

**Input:** Seeds  $\mathcal{O}_{\text{seed}} = \{(q_i, a_i)\}_{i=1}^N$ ; core principles  $\mathcal{P}_{\text{core}}$ ; reward variants  $\mathcal{V}_{\text{Reward}}$ ; teacher  $T$

**Output:** Augmented dataset  $\mathcal{D}_{\text{fullP-TTS}}$

```

 $\mathcal{D}_{\text{fullP-TTS}} \leftarrow \emptyset$ 
foreach  $(q_i, a_i) \in \mathcal{O}_{\text{seed}}$  do
  Query  $T$  with  $q_i$  to obtain  $(r_i^{(\emptyset)}, y_i^{(\emptyset)})$ 
  Add  $(q_i, r_i^{(\emptyset)}, y_i^{(\emptyset)}, a_i)$  to  $\mathcal{D}_{\text{fullP-TTS}}$ 
  foreach  $p \in \mathcal{P}_{\text{core}}$  do
     $q_i^{(p)} \leftarrow \tau_p \parallel q_i$  // wrap;
    preserve  $q_i$ 
    Query  $T$  with  $q_i^{(p)}$  to obtain  $(r_i^{(p)}, y_i^{(p)})$ 
    Add  $(q_i^{(p)}, r_i^{(p)}, y_i^{(p)}, a_i)$ 
  foreach  $R_j \in \mathcal{V}_{\text{Reward}}$  do
     $q_i^{(R_j)} \leftarrow \tau_{R_j} \parallel q_i$ 
    Query  $T$  with  $q_i^{(R_j)}$  to obtain  $(r_i^{(R_j)}, y_i^{(R_j)})$ 
    Add  $(q_i^{(R_j)}, r_i^{(R_j)}, y_i^{(R_j)}, a_i)$ 
return  $\mathcal{D}_{\text{full P-TTS}}$ 

```

---

vating our choice in the main text.

## I Evaluation of Reward Framing Variants

We evaluate variants of the reward-framing across different models. Table 14 summarizes the accuracy of O1-mini, Gemini, and DeepSeek on a fixed subset of math problems (AIME 2022–2024) under each variant.

## J Semantic and Lexical Diversity Analyses

Fig. 6 plots accuracy versus semantic Diversity Gain for four principled prompting strategies (Reward, Correctness, Penalty, Think Step By Step) across MATH500, GPQA-Diamond, and AIME25 for the 7B model. Diversity Gain is computed relative to the original seed-only baseline.

Variants with higher semantic diversity, particularly Reward and Penalty, tend to yield larger accuracy improvements, complementing the trigram-diversity results reported in Table 3.

## K Core-P-TTS Leave-One-Out Ablation

Table 15 reports a leave-one-principle-out ablation over the Core-P-TTS dataset  $\mathcal{D}_{\text{core}}$ . We fine-tune Qwen2.5-7B-Instruct on the full core set (Reward+Correctness+Penalty+Think; 360 prompts) and then retrain after omitting each principle in turn (270 prompts), as discussed in Sec. 4.3.

## L Data-Volume Ablation

Table 8 shows results for scaling the P-TTS training set from 360 to 900 examples.

## M Teacher Output Length and Cost Across Prompt Variants

Table 16 reports the average completion length, extracted reasoning length, answer length, and estimated API cost of teacher generations across different single-principle prompt variants. Table 7 summarizes the total teacher-generation cost for constructing the P-TTS<sub>Core</sub> and full P-TTS datasets.

## N Example from $\mathcal{D}_{\text{full-P-TTS}}$

Fig. 7, 8, 9, 10, and 11 illustrate a seed problem (Original) and the corresponding wrappers, along with the teacher-generated response and reasoning trace for each.

## O P-TTS-32B vs. S1-32B

Fig. 13 shows a qualitative, side-by-side example comparison between P-TTS-32B and S1-32B.

## P Example of P-TTS-32B

Fig. 12 shows an example response produced by P-TTS-32B on a single input question.

Model	#Ex	AIME25	MATH500	GPQA-D	Avg.
Qwen2.5-7B-Instruct (base)	–	6.67	76.40	36.36	39.81
S1-7B	1K	13.33	77.20	41.41	43.98
S1.1-7B	1K	20.00	81.20	41.41	47.54
P-TTSCore-7B	360	20.00	80.40	35.86	45.42
P-TTSCore+ Seed-7B	450	20.00	81.00 <sup>†</sup>	36.87 <sup>†</sup>	45.96 <sup>†</sup>
P-TTSCore+ Seed+ RewardVar-7B	900	<b>26.67<sup>†</sup></b>	<b>84.20<sup>†</sup></b>	<b>41.92<sup>†</sup></b>	<b>50.93<sup>†</sup></b>

Table 8: **Data-volume ablation.** Accuracy (%) when fine-tuning (i) on Core P-TTS only (360 prompts), (ii) Core+ Seed (450), and (iii) Core+ Seed+ six Reward-variant prompts (900). Best values in each column are **bold**; <sup>†</sup> marks a gain over the immediately preceding configuration.

Instructional Principle $p$	Template $\tau_p$ (excerpt)	Operator
Reward	“I am going to tip \$200,000 for a better solution!”	$q^{(\text{rew})} = \tau_{\text{rew}} \parallel q$
Correctness	“Your task is... You <b>MUST</b> ...”	$q^{(\text{corr})} = \tau_{\text{corr}} \parallel q$
Penalty	“You will be penalized if you provide the wrong answer.”	$q^{(\text{pen})} = \tau_{\text{pen}} \parallel q$
StepByStep	“Think step by step.”	$q^{(\text{step})} = \tau_{\text{step}} \parallel q$

Table 9: Core P-TTS instructional wrappers. Each operator preserves problem semantics by concatenating a fixed template to the unmodified  $q$ .

Variant ID	Incentive Phrase	Transformation
<b>R1</b>	I am going to tip \$200,000 for a better solution!	$\mathcal{I}(R_1)$
<b>R2</b>	I am going to tip you \$200,000 <b>if you provide the correct answer!</b>	$\mathcal{T}_{\text{COND}}(R_1)$
<b>R3</b>	I am going to tip you <b>\$200 if you provide a better solution!</b>	$\mathcal{T}_{\text{COND}}(\mathcal{T}_{\text{LOW}}(R_1))$
<b>R4</b>	<b>Add at the end of the prompt:</b> I am going to tip \$200,000 for a better solution!	$\mathcal{T}_{\text{SUFFIX}}(R_1)$
<b>R5</b>	I am going to tip <b>\$200</b> for a better solution!	$\mathcal{T}_{\text{LOW}}(R_1)$
<b>R6</b>	I am going to tip you \$500,000 for a better solution and <b>step-by-step reasoning!</b>	$\mathcal{T}_{\text{REASON}}(\mathcal{T}_{\text{HIGH}}(R_1))$

Table 10: **Reward Framing** variants in P-TTS. Each variant is generated from the baseline  $R_1$  via an operator:  $R_j = \mathcal{T}_j(R_1)$ , where  $\mathcal{I}$  denotes identity. Operator definitions:  $\mathcal{T}_{\text{COND}} : C$  (add conditionality);  $\mathcal{T}_{\text{LOW}} : M \downarrow$  (reduce magnitude in USD);  $\mathcal{T}_{\text{HIGH}} : M \uparrow$  (increase magnitude in USD);  $\mathcal{T}_{\text{SUFFIX}} : \pi = \text{suffix}$  (change placement);  $\mathcal{T}_{\text{REASON}} : \rho \neq \emptyset$  (add reasoning cue, e.g., step-by-step).

Benchmark	7B Models			14B Models		32B Models		
	P-TTSCore	S1	S1.1	P-TTSCore	S1.1	P-TTSCore	S1	S1.1
AIME25	20.00	13.33	20.00	33.33	33.33	46.67	26.70	50.00
MATH500	80.40	77.20	81.20	89.80	91.60	94.00	92.60	94.40
GPQA-Diamond	35.86	41.41	41.41	45.96	51.01	53.03	56.60	60.60

Table 11: **Accuracy (%) on three benchmarks with grouped model sizes.** Each group shows results for core P-TTS, S1, and S1.1.

Benchmark	7B Models			14B Models		32B Models		
	P-TTS	S1	S1.1	P-TTS	S1.1	P-TTS	S1	S1.1
AIME25	26.67	13.33	20.00	26.67	33.33	53.33	26.70	50.00
MATH500	84.20	77.20	81.20	90.40	91.60	94.20	92.60	94.40
GPQA-Diamond	41.92	41.41	41.41	51.01	51.01	60.61	56.60	60.60

Table 12: **Accuracy (%) on three benchmarks with grouped model sizes.** Each group shows results for P-TTS, S1, and S1.1.

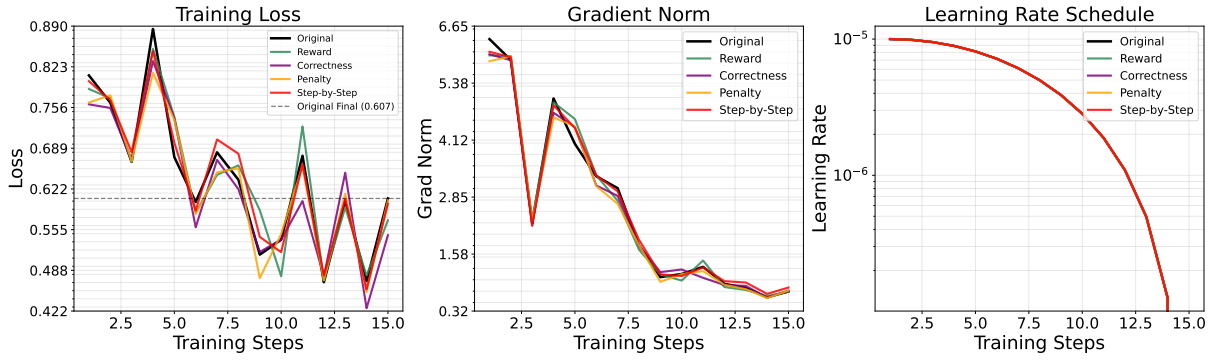


Figure 5: Training dynamics of P-TTS-32B.

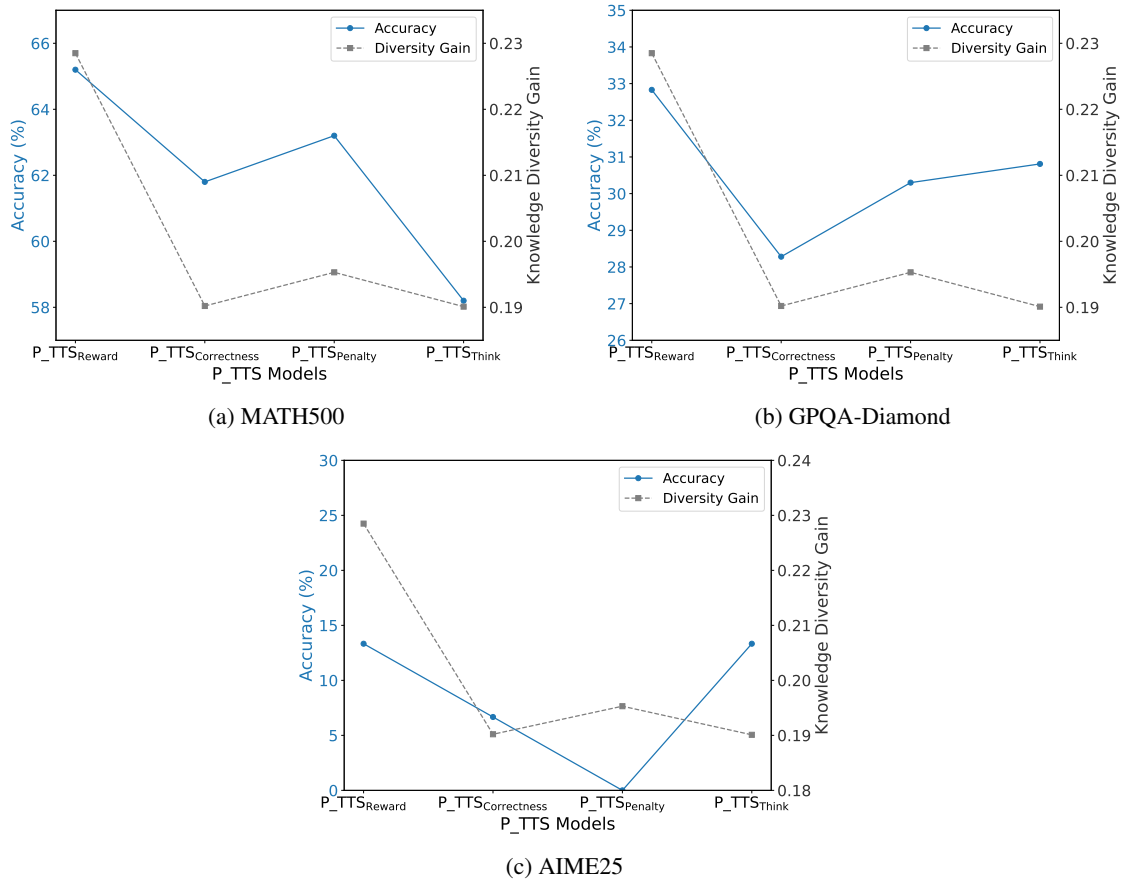


Figure 6: **Knowledge Diversity Gain vs. Accuracy for different P-TTS variants.** We compare the trade-off between Accuracy (blue solid line, left y-axis) and Knowledge Diversity Gain (gray dashed line, right y-axis) on 7B model for four principled prompting strategies: Reward, Correctness, Penalty, and Think. Diversity Gain is computed relative to the original P-TTS baseline.

Model	#Ex.	AIME25	GPQA-Diamond	MATH
Claude	810	0.00	34.34	72.80
O4-mini-high	810	0.00	41.41	60.00
DeepSeek + Claude	1620	20.00	37.88	83.20
DeepSeek + O4mini	1620	13.33	35.86	74.00

Table 13: Performance comparison across benchmarks. Values represent accuracy (%) per dataset.

Reward variant	O1-mini	Gemini	DeepSeek
R1 (Large Reward)	60.0%	33.3%	74.4%
R2 (Reward2)	52.2%	32.2%	73.3%
R3 (Reduced Reward2)	57.8%	32.2%	75.6%
R4 (Reward at End)	51.1%	28.9%	72.2%
R5 (Reduced Reward1)	56.7%	33.3%	67.8%

Table 14: Model Accuracy Across Five Variants of the Reward. Each row corresponds to a different formulation of the reward-based prompt, evaluated on a fixed subset of math questions. Results are reported as the percentage of correct answers for each model: O1-mini, Gemini, and DeepSeek. Table 10 lists the Reward-framing paraphrases.

Model	#Ex.	AIME25	MATH500	GPQA-D	Avg.
P-TTSCore (all 4)	360	20.00	80.40	35.86	45.42
P-TTSCore\Reward	270	13.33	78.80	32.83	41.65
P-TTSCore\Correctness	270	20.00	79.80	35.86	45.22
P-TTSCore\Penalty	270	20.00	79.80	34.34	44.71
P-TTSCore\Think	270	26.67	78.20	32.32	45.73

Table 15: **Leave-one-principle-out ablation.** We fine-tune Qwen2.5-7B-Instruct on the full Core-P-TTS set (Reward+Correctness+Penalty+Think; 360 prompts) and then re-train after omitting one principle (270 prompts).

Prompt Variant	# Calls	Completion	Reasoning	Answer	Total Cost (USD)
Seed	90	10638.3	10122.8	515.5	0.4031
Reward Framing	90	10685.8	10168.4	517.4	0.4065
Correctness Framing	90	9366.7	8856.4	510.3	0.3551
Penalty Framing	90	10204.4	9685.3	519.1	0.3867
Step-by-step	90	10046.4	9531.2	515.2	0.3806

Table 16: **Teacher token usage and estimated API cost across prompting variants.** Completion, reasoning, and answer columns report average token counts per teacher generation. Reasoning tokens are a subset of completion tokens, and answer tokens are computed as completion minus reasoning. Costs are reported in USD.

## Original

### Question

Let  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$  be points in the coordinate plane. Let  $\mathcal{F}$  be the family of segments  $\overline{PQ}$  of unit length lying in the first quadrant with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis. There is a unique point  $C$  on  $\overline{AB}$ , distinct from  $A$  and  $B$ , that does not belong to any segment from  $\mathcal{F}$  other than  $\overline{AB}$ . Then  $OC^2 = \frac{p}{q}$ , where  $p$  and  $q$  are relatively prime positive integers. Find  $p + q$ .

### Response

Given points  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$ , we need to find the unique point  $C$  on  $\overline{AB}$  that does not belong to any segment from the family  $\mathcal{F}$  other than  $\overline{AB}$ .

The family  $\mathcal{F}$  consists of segments  $\overline{PQ}$  of unit length with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis. Each segment in  $\mathcal{F}$  can be parametrized as  $x/\cos(\theta) + y/\sin(\theta) = 1$  where  $\theta$  is an angle parameter. The segment  $\overline{AB}$  itself corresponds to  $\theta = \pi/3$ .

To find the point  $C$ , we parametrize  $\overline{AB}$  as  $x = \frac{1-t}{2}$  and  $y = \frac{\sqrt{3}t}{2}$  for  $t \in [0, 1]$ . Substituting these coordinates into the equation of the line segment from  $\mathcal{F}$ , we derive the condition for  $C$  to lie only on  $\overline{AB}$ .

By solving the resulting equations and ensuring that the derivative condition (to check for a double root) is satisfied, we find that  $t = \frac{3}{4}$ . This gives the coordinates of  $C$  as  $(\frac{1}{8}, \frac{3\sqrt{3}}{8})$ .

Calculating  $OC^2$ :

$$OC^2 = \left(\frac{1}{8}\right)^2 + \left(\frac{3\sqrt{3}}{8}\right)^2 = \frac{1}{64} + \frac{27}{64} = \frac{28}{64} = \frac{7}{16}$$

Thus,  $OC^2 = \frac{7}{16}$  where  $p = 7$  and  $q = 16$ . Therefore,  $p + q = 7 + 16 = 23$ .

23

### Reasoning Trace

Okay, so I need to find the square of the distance from the origin  $O$  to this special point  $C$  on segment  $AB$ . The problem says that  $C$  is the unique point on  $AB$ , other than  $A$  and  $B$ , that isn't on any other segment from the family ....

Figure 7: Example of an original AIME-style math problem, along with its model-generated response and reasoning trace. In our methodology, these original seed examples are later paired with principle-based reformulations (e.g., Reward, Penalty, Correctness, Step-by-Step), enabling the construction of the full P-TTS augmented corpus.

## Reward

### Question

I am going to tip \$200000 for a better solution! Let  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$  be points in the coordinate plane. Let  $\mathcal{F}$  be the family of segments  $\overline{PQ}$  of unit length lying in the first quadrant with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis. There is a unique point  $C$  on  $\overline{AB}$ , distinct from  $A$  and  $B$ , that does not belong to any segment from  $\mathcal{F}$  other than  $\overline{AB}$ . Then  $OC^2 = \frac{p}{q}$ , where  $p$  and  $q$  are relatively prime positive integers. Find  $p + q$ .

### DeepSeek Response

Given points  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$ , we need to find the unique point  $C$  on  $\overline{AB}$  that does not belong to any segment from the family  $\mathcal{F}$  of unit length segments  $\overline{PQ}$  with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis.

1. **\*\*Equation of  $\overline{AB}$ \*\***: - The line segment  $\overline{AB}$  has endpoints  $A$  and  $B$ . - The equation of  $\overline{AB}$  is  $y = -\sqrt{3}x + \frac{\sqrt{3}}{2}$ .

2. **\*\*Parametrization of  $\overline{AB}$ \*\***: - Points on  $\overline{AB}$  can be parametrized as  $x = \frac{1-t}{2}$  and  $y = \frac{\sqrt{3}}{2}t$  for  $t \in [0, 1]$ .

3. **\*\*Envelope of the family  $\mathcal{F}$ \*\***: - The family  $\mathcal{F}$  consists of segments  $\overline{PQ}$  with  $P = (p, 0)$  and  $Q = (0, q)$  where  $p^2 + q^2 = 1$ . - The envelope of these segments forms an astroid given by  $x^{2/3} + y^{2/3} = 1$ .

4. **\*\*Finding the point of tangency\*\***: - The point  $C$  where  $\overline{AB}$  is tangent to the astroid is found by solving the tangency condition. - The slope of  $\overline{AB}$  is  $-\sqrt{3}$ , which must match the slope of the astroid at the point of tangency. - The point of tangency is found to be  $(\cos^3 \theta, \sin^3 \theta)$  where  $\theta = \frac{\pi}{3}$ , leading to coordinates  $(\frac{1}{8}, \frac{3\sqrt{3}}{8})$ .

5. **\*\*Calculating  $OC^2$ \*\***: - The coordinates of  $C$  are  $(\frac{1}{8}, \frac{3\sqrt{3}}{8})$ . - Squaring the distance from  $O$  to  $C$ :

$$OC^2 = (\frac{1}{8})^2 + (\frac{3\sqrt{3}}{8})^2 = \frac{1}{64} + \frac{27}{64} = \frac{28}{64} = \frac{7}{16}$$

- The fraction  $\frac{7}{16}$  has  $p = 7$  and  $q = 16$ , so  $p + q = 23$ .

Thus, the final answer is  $\boxed{23}$ .

### Reasoning Trace

Okay, so I have this geometry problem here. Let me try to parse it step by step.

First, they give me three points:  $O$  at  $(0,0)$ ,  $A$  at  $(1/2, 0)$ , and  $B$  at  $(0, \text{sqrt}(3)/2)$ . Then there's this family of segments  $\mathcal{F}$ , which consists of segments  $PQ$  of unit length lying in the first quadrant, with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis. The problem states that . . .

Figure 8: Reward-wrapper example: a seed problem wrapped with Reward framing, and the resulting teacher-generated response and reasoning trace.

## Step-by-Step Reasoning

### Question

Think step by step: Let  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$  be points in the coordinate plane. Let  $\mathcal{F}$  be the family of segments  $\overline{PQ}$  of unit length lying in the first quadrant with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis. There is a unique point  $C$  on  $\overline{AB}$ , distinct from  $A$  and  $B$ , that does not belong to any segment from  $\mathcal{F}$  other than  $\overline{AB}$ . Then  $OC^2 = \frac{p}{q}$ , where  $p$  and  $q$  are relatively prime positive integers. Find  $p + q$ .

### DeepSeek Response

Given points  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$ , we need to find a unique point  $C$  on  $\overline{AB}$  that does not belong to any other segment from the family  $\mathcal{F}$  of unit length segments  $\overline{PQ}$  with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis.

1. **\*\*Equation of  $\overline{AB}$ \*\***: - Parametrize  $\overline{AB}$  with  $t \in [0, 1]$ :

$$x(t) = \frac{1-t}{2}, \quad y(t) = \frac{\sqrt{3}}{2}t$$

2. **\*\*Family  $\mathcal{F}$ \*\***: - Segments  $\overline{PQ}$  with  $P = (p, 0)$  and  $Q = (0, q)$  such that  $p^2 + q^2 = 1$ .

3. **\*\*Envelope of  $\mathcal{F}$ \*\***: - The envelope of the family  $\mathcal{F}$  is the astroid  $x^{2/3} + y^{2/3} = 1$ .

4. **\*\*Intersection of  $\overline{AB}$  with the Astroid\*\***: - Substitute  $x = \frac{1-t}{2}$  and  $y = \frac{\sqrt{3}}{2}t$  into the astroid equation:

$$\left(\frac{1-t}{2}\right)^{2/3} + \left(\frac{\sqrt{3}}{2}t\right)^{2/3} = 1$$

- Solving this equation, we find  $t = \frac{3}{4}$ .

5. **\*\*Coordinates of  $C$ \*\***: - For  $t = \frac{3}{4}$ :

$$x = \frac{1 - \frac{3}{4}}{2} = \frac{1}{8}, \quad y = \frac{\sqrt{3}}{2} \cdot \frac{3}{4} = \frac{3\sqrt{3}}{8}$$

6. **\*\*Distance  $OC$ \*\***: - Calculate  $OC^2$ :

$$OC^2 = \left(\frac{1}{8}\right)^2 + \left(\frac{3\sqrt{3}}{8}\right)^2 = \frac{1}{64} + \frac{27}{64} = \frac{28}{64} = \frac{7}{16}$$

Thus,  $OC^2 = \frac{7}{16}$  where  $p = 7$  and  $q = 16$ . Therefore,  $p + q = 7 + 16 = 23$ .

23

### Reasoning Trace

Alright, let's try to tackle this problem step by step. First ...

Figure 9: StepByStep-wrapper example used in P-TTS.

## Penalty Framing

### Question

You will be penalized if you provide the wrong answer. Let  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$  be points in the coordinate plane. Let  $\mathcal{F}$  be the family of segments  $\overline{PQ}$  of unit length lying in the first quadrant with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis. There is a unique point  $C$  on  $\overline{AB}$ , distinct from  $A$  and  $B$ , that does not belong to any segment from  $\mathcal{F}$  other than  $\overline{AB}$ . Then  $OC^2 = \frac{p}{q}$ , where  $p$  and  $q$  are relatively prime positive integers. Find  $p + q$ .

### Response

Given points  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$ , we need to find the unique point  $C$  on  $\overline{AB}$  that does not belong to any other segment from the family  $\mathcal{F}$  of unit length segments  $\overline{PQ}$  with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis. We need to determine  $OC^2$  and express it as  $\frac{p}{q}$  where  $p$  and  $q$  are coprime positive integers, then find  $p + q$ .

1. **\*\*Parametrization of  $\overline{AB}$ \*\***: - The parametric equations for  $\overline{AB}$  are  $x = \frac{1}{2} - \frac{1}{2}t$  and  $y = \frac{\sqrt{3}}{2}t$  for  $t \in [0, 1]$ . Thus, any point  $C$  on  $\overline{AB}$  can be written as  $(\frac{1-t}{2}, \frac{\sqrt{3}}{2}t)$ .
2. **\*\*Equation of line  $\overline{PQ}$ \*\***: - For a segment  $\overline{PQ}$  in  $\mathcal{F}$  with endpoints  $P = (p, 0)$  and  $Q = (0, q)$ , the equation is  $\frac{x}{p} + \frac{y}{q} = 1$  where  $p^2 + q^2 = 1$ .
3. **\*\*Condition for point  $C$ \*\***: - Substituting  $C(\frac{1-t}{2}, \frac{\sqrt{3}}{2}t)$  into the line equation  $\frac{x}{p} + \frac{y}{q} = 1$  gives  $\frac{(1-t)/2}{p} + \frac{\sqrt{3}t/2}{q} = 1$ . - Using  $p = \cos \theta$  and  $q = \sin \theta$ , we derive the equation  $\frac{(1-t)/2}{\cos \theta} + \frac{\sqrt{3}t/2}{\sin \theta} = 1$ .
4. **\*\*Solving for  $t$ \*\***: - Simplifying and solving for  $t$  using trigonometric identities and ensuring the equation has a unique solution at  $\theta = \frac{\pi}{3}$  (corresponding to  $\overline{AB}$ ) leads to  $t = \frac{3}{4}$ .
5. **\*\*Coordinates of  $C$ \*\***: - Substituting  $t = \frac{3}{4}$  into the parametric equations,  $C$  is  $(\frac{1}{8}, \frac{3\sqrt{3}}{8})$ .
6. **\*\*Calculating  $OC^2$ \*\***: -  $OC^2 = (\frac{1}{8})^2 + (\frac{3\sqrt{3}}{8})^2 = \frac{1}{64} + \frac{27}{64} = \frac{28}{64} = \frac{7}{16}$ .  
Thus,  $p = 7$  and  $q = 16$ , leading to  $p + q = 23$ .

23

### Reasoning Trace

Okay, let's try to tackle this problem. First, let me make sure I understand what's being asked here. We have three points: O at (0,0), A at (1/2, 0), and B at (0, sqrt(3)/2). The line segment AB is part of the ....

Figure 10: Penalty-wrapper example used in P-TTS.

## Correctness Emphasis

### Question

Your Task is to solve the following: Let  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$  be points in the coordinate plane. Let  $\mathcal{F}$  be the family of segments  $\overline{PQ}$  of unit length lying in the first quadrant with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis. There is a unique point  $C$  on  $\overline{AB}$ , distinct from  $A$  and  $B$ , that does not belong to any segment from  $\mathcal{F}$  other than  $\overline{AB}$ . Then  $OC^2 = \frac{p}{q}$ , where  $p$  and  $q$  are relatively prime positive integers. Find  $p + q$ . You must provide the correct answer!

### DeepSeek Response

Given points  $O = (0, 0)$ ,  $A = (\frac{1}{2}, 0)$ , and  $B = (0, \frac{\sqrt{3}}{2})$ , we need to find the point  $C$  on  $\overline{AB}$  that does not belong to any other segment from the family  $\mathcal{F}$  of unit length segments in the first quadrant with endpoints on the axes.

- Parametrization of  $\overline{AB}$** : - The parametric equations for  $\overline{AB}$  are  $x = \frac{1-t}{2}$  and  $y = \frac{\sqrt{3}}{2}t$  for  $t \in [0, 1]$ .
- Envelope of the family  $\mathcal{F}$** : - The envelope of the family of segments  $PQ$  with  $P$  on the  $x$ -axis and  $Q$  on the  $y$ -axis, where  $PQ$  has unit length, forms an astroid given by  $x^{2/3} + y^{2/3} = 1$ .
- Finding the intersection of  $\overline{AB}$  with the astroid**: - Substitute the parametric equations of  $\overline{AB}$  into the astroid equation:

$$\left(\frac{1-t}{2}\right)^{2/3} + \left(\frac{\sqrt{3}}{2}t\right)^{2/3} = 1$$

- Solve for  $t$  by ensuring the gradient of  $\overline{AB}$  matches the gradient of the astroid at the point of tangency. This leads to the point  $C$  where  $t = \frac{3}{4}$ .

- Coordinates of  $C$** : - Substituting  $t = \frac{3}{4}$  into the parametric equations:

$$x = \frac{1 - \frac{3}{4}}{2} = \frac{1}{8}, \quad y = \frac{\sqrt{3}}{2} \cdot \frac{3}{4} = \frac{3\sqrt{3}}{8}$$

- Calculating  $OC^2$** : - The coordinates of  $C$  are  $(\frac{1}{8}, \frac{3\sqrt{3}}{8})$ . - Compute  $OC^2$ :

$$OC^2 = \left(\frac{1}{8}\right)^2 + \left(\frac{3\sqrt{3}}{8}\right)^2 = \frac{1}{64} + \frac{27}{64} = \frac{28}{64} = \frac{7}{16}$$

Thus,  $OC^2 = \frac{7}{16}$ , and the sum  $p + q = 7 + 16 = 23$ .

23

### Reasoning Trace

Okay, let's try to tackle this problem step by step. First, let's make sure ....

Figure 11: Correctness-wrapper example.

## Question

If the Sun's absolute magnitude is +5, find the luminosity of a star of magnitude 0 in ergs/s.  
 A useful constant: the luminosity of the sun is  $3.83 \times 10^{33}$  ergs/s

## Response

Okay, so I need to find the luminosity of a star with magnitude 0, given that the Sun's absolute magnitude is +5 and its luminosity is  $3.83 \times 10^{33}$  ergs/s. Hmm, let's start by recalling the relationship between magnitude and luminosity. First, absolute magnitude ..... given as +5. The star we're interested in has an absolute magnitude of 0. The difference in magnitude between the two stars is  $5 - 0 = 5$  magnitudes. In astronomy, the difference in magnitudes relates to the ratio of luminosities. The formula is:  $m_1 - m_2 = -2.5 \log_{10}(L_1 / L_2)$  ..... indeed  $m_1 - m_2 = -2.5 \log_{10}(L_1 / L_2)$ . So if we know the difference in magnitudes, we can find the ratio of luminosities. In our case,  $m_1$  (Sun) is +5,  $m_2$  (star) is 0. So  $m_1 - m_2 = 5 - 0 = +5$ . Plugging into the formula:  $5 = -2.5 \log_{10}(L_{\text{sun}}/L_{\text{star}})$  Wait, hold on. Let me make sure I have the formula right. .... Then  $L_1$  is luminosity of Sun,  $L_2$  is luminosity of star. So  $m_1 - m_2 = 5 - 0 = +5$ . Therefore:  $5 = -2.5 \log_{10}(L_{\text{sun}}/L_{\text{star}})$  Let me solve for  $\log_{10}(L_{\text{sun}}/L_{\text{star}})$ : Divide both sides by -2.5:  $\frac{5}{-2.5} = \log_{10}(L_{\text{sun}}/L_{\text{star}})$  Which is  $-2 = \log_{10}(L_{\text{sun}}/L_{\text{star}})$  Therefore,  $L_{\text{sun}}/L_{\text{star}} = 10^{-2} = 0.01$  So  $L_{\text{star}} = L_{\text{sun}}/0.01 = L_{\text{sun}} \times 100$  Therefore, ..... is  $3.83 \times 10^{33} \times 100 = 3.83 \times 10^{35}$  ergs/s. Wait, let me check that calculation again. .... Yep, same result. Therefore, the answer should be  $3.83 \times 10^{35}$  ergs/s. Final Answer  $3.83 \times 10^{35}$  .... Alternatively, since each 5 magnitudes corresponds to a factor of 100 in luminosity, a difference of 5 magnitudes (from +5 to 0) means the star is 100 times more luminous than the Sun. So again,  $3.83 \times 10^{33} \times 100 = 3.83 \times 10^{35}$  ergs/s. Therefore, the luminosity of the star is  $3.83 \times 10^{35}$  ergs/s. Final Answer  $3.83 \times 10^{35}$  .....

Figure 12: Example of a P-TTS-32B response.

**Question :** If the Sun's absolute magnitude is +5, find the luminosity of a star of magnitude 0 in ergs/s. A useful constant: the luminosity of the sun is  $3.83 \times 10^{33}$  ergs/s

P-TTS-32B

Okay, so I need to find the luminosity of a star with magnitude 0, given that the Sun's absolute magnitude is +5 and its luminosity is  $3.83 \times 10^{33}$  ergs/s. Hmm, let's start by recalling the relationship between magnitude and luminosity.

First, absolute magnitude is a measure of a star's intrinsic brightness, as it would be seen from a standard distance. The Sun's absolute magnitude is given as +5. The star we're interested in has an absolute magnitude of 0. The difference in magnitude between the two stars is  $5 - 0 = 5$  magnitudes.

In astronomy, the difference in magnitudes relates to the ratio of luminosities. The formula is:

$$m_1 - m_2 = -2.5 \cdot \log_{10}(L_1/L_2)$$

Where  $m_1$  and  $m_2$  are the magnitudes, and  $L_1$  and  $L_2$  are the corresponding luminosities. Wait, but here we have absolute magnitudes, so this formula should apply. Let me confirm.

Yes, the formula for the difference in magnitudes is indeed  $m_1 - m_2 = -2.5 \cdot \log_{10}(L_1/L_2)$ . So if we know the difference in magnitudes, we can find the ratio of luminosities. In our case,  $m_1$  (Sun) is +5,  $m_2$  (star) is 0. So  $m_1 - m_2 = 5 - 0 = +5$ .

Plugging into the formula:

$$5 = -2.5 \cdot \log_{10}(L_{\text{sun}}/L_{\text{star}})$$

Wait, hold on. Let me make sure I have the formula right.

The formula is  $m_1 - m_2 = -2.5 \cdot \log_{10}(L_1/L_2)$ . So if  $m_1$  is the magnitude of the first star (Sun),  $m_2$  is the magnitude of the second star (star of interest). Then  $L_1$  is luminosity of Sun,  $L_2$  is luminosity of star.

So  $m_1 - m_2 = 5 - 0 = +5$ .

Therefore:

$$5 = -2.5 \cdot \log_{10}(L_{\text{sun}}/L_{\text{star}})$$

Let me solve for  $\log_{10}(L_{\text{sun}}/L_{\text{star}})$ :

$$5 / (-2.5) = \log_{10}(L_{\text{sun}}/L_{\text{star}})$$

$$-2 = \log_{10}(L_{\text{sun}}/L_{\text{star}})$$

$$\text{Therefore, } L_{\text{sun}}/L_{\text{star}} = 10^{-2} = 0.01$$

$$\text{So } L_{\text{star}} = L_{\text{sun}} / 0.01 = L_{\text{sun}} \times 100$$

Therefore, the luminosity of the star is 100 times that of the Sun.

Given that the Sun's luminosity is  $3.83 \times 10^{33}$  ergs/s, then the star's luminosity is  $3.83 \times 10^{33} \times 100 = 3.83 \times 10^{35}$  ergs/s.

Wait, let me check that calculation again.  $3.83 \times 10^{33}$  multiplied by 100 is indeed  $3.83 \times 10^{35}$ . Yes.

But let me verify the steps again to make sure I didn't make a mistake.

Starting with the magnitude difference: Sun's absolute magnitude is +5, star's is 0, so difference is +5.

Using the formula  $m_1 - m_2 = -2.5 \cdot \log_{10}(L_1/L_2)$

$$\text{Plugging in: } 5 = -2.5 \cdot \log_{10}(L_{\text{sun}}/L_{\text{star}})$$

$$\text{Divide both sides by } -2.5: 5 / (-2.5) = -2 = \log_{10}(L_{\text{sun}}/L_{\text{star}})$$

$$\text{Therefore, } L_{\text{sun}}/L_{\text{star}} = 10^{-2} = 0.01$$

$$\text{Thus, } L_{\text{star}} = L_{\text{sun}} / 0.01 = L_{\text{sun}} \times 100$$

$$\text{Therefore, } 3.83 \times 10^{33} \times 100 = 3.83 \times 10^{35} \text{ ergs/s.}$$

That seems correct. So the luminosity of the star is  $3.83 \times 10^{35}$  ergs/s.

But let me check if there's another way to approach this to confirm.

...

S1-32B

think

The problem asks for the luminosity of a star with a given apparent magnitude, given the absolute magnitude of the Sun and the Sun's luminosity.

We are given:

$$\text{Sun's absolute magnitude, } M_{\text{sun}} = +5$$

$$\text{Magnitude of the star, } m = 0$$

$$\text{Luminosity of the Sun, } L_{\text{sun}} = 3.83 \times 10^{33} \text{ ergs/s}$$

We need to find the luminosity of the star,  $L_{\text{star}}$ .

The relationship between absolute magnitude and luminosity is given by:

$$M = -2.5 \cdot \log_{10}(L/L_0)$$

where  $M$  is the absolute magnitude,  $L$  is the luminosity, and  $L_0$  is a reference luminosity.

For the Sun, we have:

$$M_{\text{sun}} = -2.5 \cdot \log_{10}(L_{\text{sun}}/L_0)$$

$$5 = -2.5 \cdot \log_{10}(3.83 \times 10^{33} / L_0)$$

We can solve this equation for  $L_0$ :

$$5 / (-2.5) = \log_{10}(3.83 \times 10^{33} / L_0)$$

$$-2 = \log_{10}(3.83 \times 10^{33} / L_0)$$

$$10^{-2} = 3.83 \times 10^{33} / L_0$$

$$0.01 = 3.83 \times 10^{33} / L_0$$

$$L_0 = 3.83 \times 10^{33} / 0.01$$

$$L_0 = 3.83 \times 10^{33} \times 100$$

$$L_0 = 3.83 \times 10^{35} \text{ ergs/s}$$

Now, we need to find the luminosity of the star. The apparent magnitude  $m$  is related to the absolute magnitude  $M$  by the distance modulus:

$$m - M = 5 \cdot \log_{10}(d) - 5$$

where  $d$  is the distance to the star in parsecs.

The absolute magnitude of the star can be found using the same formula as for the Sun:

$$M_{\text{star}} = -2.5 \cdot \log_{10}(L_{\text{star}}/L_0)$$

We are given the apparent magnitude of the star,  $m = 0$ . We need to find the absolute magnitude of the star. We can use the fact that the apparent magnitude of the Sun is about -26.74.

Let's use the relationship between apparent magnitude, absolute magnitude, and distance. For the Sun, which is at a distance of approximately 1 astronomical unit (AU), we have:

$$m_{\text{sun}} - M_{\text{sun}} = 5 \cdot \log_{10}(d_{\text{sun}}) - 5$$

The average distance from Earth to the Sun is about  $1.496 \times 10^{13}$  cm.

$$1 \text{ parsec} = 3.086 \times 10^{18} \text{ cm.}$$

$$d_{\text{sun}} \text{ in parsecs} = (1.496 \times 10^{13}) / (3.086 \times 10^{18}) \approx 4.85 \times 10^{-6} \text{ parsecs.}$$

Using the apparent magnitude of the Sun,  $m_{\text{sun}} \approx -26.74$ :

$$-26.74 - 5 = 5 \cdot \log_{10}(4.85 \times 10^{-6}) - 5$$

$$-31.74 = 5 \cdot (\log_{10}(4.85) + \log_{10}(10^{-6})) - 5$$

$$-31.74 = 5 \cdot (\log_{10}(4.85) - 6) - 5$$

$$-31.74 = 5 \cdot \log_{10}(4.85) - 30 - 5$$

$$-31.74 = 5 \times 0.6856 - 35$$

$$-31.74 = 3.428 - 35$$

$$-31.74 = -31.572$$

This is close enough considering the approximations used.

Now, let's consider the star with magnitude 0. We need to find its absolute magnitude. We can use the Sun as a reference.

...

Figure 13: Qualitative comparison between P-TTS-32B and the baseline model.