

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

- 90 Samples Can Beat 1K in the Wild

Sondos Mahmoud Bsharat and Zhiqiang Shen*

VILA Lab, MBZUAI

* E-Mail: zhiqiang.shen@mbzuai.ac.ae (Correspondence)

Abstract

Large language models (LLMs) have demonstrated impressive reasoning capabilities when provided with chain-of-thought exemplars, but curating large reasoning datasets remains laborious and resource-intensive. In this work, we introduce Prompting Test-Time Scaling (P-TTS), a simple yet effective inference-time data augmentation strategy for enhancing LLM reasoning through finetuning. Rather than collecting thousands or even millions of examples, P-TTS leverages a small pool of only 90 manually selected reasoning instances and systematically varies exemplar augmentation through principled instruction prompting intensities at test time to synthesize diverse reasoning trajectory contexts. Then we finetune the various sizes of Qwen-2.5 models on P-TTS data. Across a suite of mathematical reasoning AIME2024 & 25, MATH500, and GPQA-Diamond, our P-TTS-7B and 32B models outperform the prior competitive baselines like S1 and S1.1 (1K-shot), achieving absolute accuracy gains of +26.66% and +30.00% on AIME'24 (7B), and +13.34% and +6.67% on AIME'25 (7B); P-TTS-32B yields gains of +23.33% and +16.63% on AIME'24, and +26.63% and +3.33% on AIME'25 (vs. S1 and S1.1, respectively), with comparable or better performance on MATH500 and GPQA-Diamond. We further show that P-TTS enhances zero-shot generalization accuracy on out-of-domain reasoning benchmarks of Gaokao, Kaoyan, OlympiadBench, AMC23, GradeSchoolMath, and Minerva. Ablation studies confirm that both exemplar diversity and scaled sampling schedules are critical drivers of improvement. Our analysis suggests that test-time scaling effectively explores the latent space of reasoning patterns, amplifying LLM problem-solving with minimal annotation overhead, and further unlocking the reasoning potential and capabilities of LLMs. Prompting Test-Time Scaling offers a practical, low-cost way to elicit LLM reasoning in resource-constrained or rapidly evolving domains. Our code and data are available at https://github.com/VILA-Lab/PTTS.

Keywords: Prompting Test-Time Scaling; LLM Reasoning; Large Language Models; Principled Instructions;

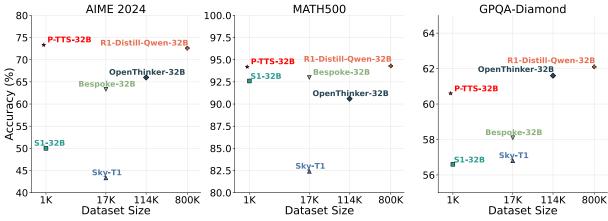


Figure 1. Comparison of 32B-scale models on AIME 2024 (left), MATH500 (middle), and GPQA Diamond (right). Model performance on AIME 2024, MATH500, and GPQA-Diamond benchmarks as a function of dataset size. Each point represents a different model, with our P-TTS-32B (red star) showing competitive performance from a significantly smaller dataset. The x-axis scale highlights the differences in training data sizes across models.

1. Introduction

Large language models (LLMs) [RNS+18, AAA+23, TAB+23, Ant25] attain strong deductive and quantitative reasoning once equipped with curated chains of thought (CoT) [WWS+22b] or tool-augmented exemplars [MGH+24]. 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—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 at test time can simulate the effect of large reasoning datasets, without actually collecting them.

In this work, we propose Prompting Test-Time Scaling (P-TTS) as an LLM data augmentation method: given a compact seed pool of just 90 high-quality math reasoning exemplars, we algorithmically expand the prompt space at inference by 1) exemplar subsampling under diversity constraints using various principled instructions [BMS23], 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 set of independently constructed prompt contexts, whose answers are collected. This converts test-time prompting into a scalable 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 through two orthogonal dimensions: 1) Semantic/knowledge diversity in CoT responses – measuring how P-TTS expands the coverage of knowledge and concepts. We compare the trade-off between accuracy and knowledge diversity gain across four principled prompting strategies: Reward, Correctness, Penalty, and Think. 2) Language trigram diversity – quantifying lexical and phrasing variety using distinct n-gram ratios and entropy, applied both to full responses and isolated reasoning traces. Results show that reward framing achieves the greatest lexical diversity, reflecting stronger surface-level variation from the base examples. More broadly, we observe synergistic benefits: semantic/knowledge diversity reduces overfitting to narrow reasoning templates and enhances transfer; trigram diversity mitigates lexical/template lock-in and strengthens robustness; ordering perturbations counter position bias slightly; and our model-driven self-augmentation introduces novel but still on-manifold rationales. Finally, scaling curves reveal diminishing returns beyond roughly 6 prompt-augmentations per base question, suggesting this as a practical deployment point.

Contributions of this work. (1) We introduce Prompting Test-Time Scaling, a simple yet effective framework for LLM inference-time reasoning data augmentation. (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. We release code and the augmented exemplar pool to facilitate reproducibility and facilitate rapid transfer to new domains. Collectively, P-TTS reframes test-time prompting from a one-shot design choice into a scalable, stochastic process, unlocking robust reasoning without additional data collection or massive labeled datasets.

2. Related Work

LLM Reasoning. The growing availability of frontier autoregressive pretrained large language models that expose long-form rationales (e.g., GPT-o1 [JKL+24], Gemini [TAB+23], Claude 3.7 Sonnet [Ant25], DeepSeek-R1 [GYZ+25]) has integrated RL or SFT pipelines that *explicitly* incorporate intermediate reasoning. Diffusion LLM [LCGS25, ZGZG25] also shows potential in reasoning capability recently. In many open-sourced reasoning LLM settings [MYS+25, Bes25, GMK+25], a teacher model is usually prompted to produce chain-of-thought (CoT) explanations for curated inputs, and the resulting (prompt, rationale, answer) triples supervise a student model. For instance, Bespoke-Stratos [Lab25] collects teacher-generated explanations from DeepSeek-R1, while the OpenThinker models [GMK+25] train on the OpenThoughts corpora built with rationales elicited from teachers such as DeepSeek-R1 or QwQ-32B. These methods demonstrate that model-generated reasoning is a scalable supervision signal, but they typically rely on tens to hundreds of thousands of exemplars and substantial collection costs. 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 [BMR⁺20, KMH⁺20], inference-time strategies improve performance without updating model weights. Classic approaches tune decoding hyperparameters (temperature, top-k, nucleus sampling) to modulate diversity [HBD⁺19, FLD18, MPWC23], or generate multiple completions from a fixed prompt and aggregate answers, as in few-shot CoT [WWS⁺22b] and Self-Consistency [WWS⁺22a]. Recent "think-more" style methods (e.g., s1/s1.1) adjust the *reasoning budget*—allocating more deliberate tokens at test time—without altering exemplars or performing additional training [MYS⁺25]. 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. MetaMath [YJS+23] bootstraps diverse math problems via generate-and-verify loops, while ReasoningMix [ZCH+25] 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 [BMS23]. Our approach instantiates a small subset of such patterns that are directly compatible with math-style reasoning (reward, penalty, correctness, step-by-step) and formalizes them as deterministic wrap operators. This

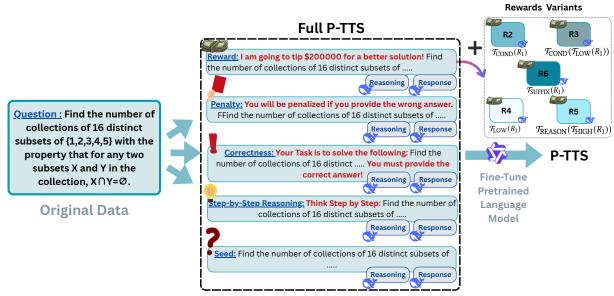


Figure 2. Overview of the P-TTS data augmentation process. Starting from a small set of high-quality math problems (AIME-style), we generate diverse prompt variants through instruction reframing, such as reward-based encouragement, penalty warnings, and step-by-step guidance. These augmented prompts are used to elicit high-quality LLM completions, which are then collected as synthetic reasoning data to fine-tune.

yields 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 [MYS⁺25] and LIMO [YHX⁺25] claim training on ~1K 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") [BMS23] to each seed example, producing prompt variants that preserve the original problem while modulating the *instructional framing* (e.g., reward/penalty cues, correctness emphasis, or step-wise guidance). Concretely, a principle p is realized by a template τ_p that deterministically wraps the raw question q, yielding $q^{(p)} = \tau_p \parallel q$ (string concatenation \parallel). 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. 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 8).

3.2. Dataset Seed Selection Strategy

We adopt a *seed-based* construction paradigm [ZAD $^+$ 25]: a small, vetted set of problems serves as the substrate for systematic instructional variation. Our seeds comprise N = 90 problems from AIME 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 [HYS $^+$ 25, AAA $^+$ 23]. This compact, high-quality set enables controlled scaling via instructional wrappers without compromising semantic fidelity.

3.3. Selection of Instructional Principles

We instantiate four core principles

$$\mathcal{P}_{core} = \{Reward, Penalty, Correctness, StepByStep\},$$

chosen for their direct applicability to math reasoning and prior evidence of consistent gains across model families [BMS23]. Each principle p is bound to a fixed template τ_p and applied as a wrap operator without modifying q's tokens:

$$q_i^{(p)} = \tau_p \parallel q_i. \tag{1}$$

This guarantees semantic invariance: removing the principled template τ_p deterministically recovers question q_i . Table 1 summarizes the templates and induced operators.

Instructional Principle <i>p</i>	Template τ_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 MUST"	$q^{(\text{corr})} = \tau_{\text{corr}} \parallel q$
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 1. Core P-TTS instructional wrappers. Each operator preserves problem semantics by concatenating a fixed template to the unmodified q.

3.4. P-TTS Dataset Construction

Our primary dataset is drawn from the AIME benchmarks (2022–2024) [Art] and consists of N = 90 unique problems with gold answers,

$$O_{\text{seed}} = \{(q_i, a_i)_{i=1}^N,$$
 (2)

where q_i is the original seed question or problem without any additional prompting, and $a_i \in \{0, 1, ..., 999\}$ is the integer-style ground-truth for the associated question. O_{seed} is the seed question-answer pairs.

Original question (null prompts, i.e., \varnothing). We first query a teacher model T (DeepSeek-R1 [GYZ⁺25]) on the *unmodified* question text, which we view as a null principle $p = \varnothing$ with

$$q_i^{(\varnothing)} = q_i. (3)$$

where $q_i^{(\varnothing)}$ is its null-prompt form (unchanged). For each $q_i^{(\varnothing)}$ the teacher returns a reasoning trace $r_i^{(\varnothing)}$ and a full response $y_i^{(\varnothing)}$ yielding

$$\mathcal{D}_{\text{seed}} = \left\{ \left(q_i^{(\varnothing)}, y_i^{(\varnothing)}, r_i^{(\varnothing)}, a_i \right) \right\}_{i=1}^N. \tag{4}$$

where $r_i^{(\varnothing)}$ is the teacher-generated reasoning trace, $y_i^{(\varnothing)}$ is the full teacher response, a_i is the ground-truth answer, and N=90 is the total number of seed problems.

Selected/Core principle transformations. As shown in Sec. 3.3 and Table 1, we select four principles as our *core set*. Each principle $p \in \mathcal{P}_{core}$ is implemented as a deterministic operator $f_p(\cdot)$ that *wraps* the original text with a fixed instructional template τ_p while leaving the tokens of q_i unmodified:

$$q_i^{(p)} = f_p(q_i) = \tau_p \parallel q_i \quad \text{(string concatenation } \parallel). \tag{5}$$

where $q_i^{(p)}$ is the wrapped question under principle p, and τ_p is the fixed instructional template. This construction preserves the mathematical content, since removing τ_p recovers the original question q_i . Querying T with each $q_i^{(p)}$ produces $(r_i^{(p)}, y_i^{(p)})$, and we collect

$$\mathcal{D}_{\text{core}} = \left\{ \left(q_i^{(p)}, \, r_i^{(p)}, \, y_i^{(p)}, \, a_i \right) \, \middle| \, i = 1, \dots, N; \, p \in \mathcal{P}_{\text{core}} \right\}. \tag{6}$$

where $r_i^{(p)}$ is the teacher reasoning trace, $y_i^{(p)}$ is the teacher response, a_i is the ground-truth answer, N is the number of seed problems, and \mathcal{P}_{core} is the set of four selected principles.

Reward framing variants for *reward principle***.** Our single-principle ablation (Sec. 4.3) shows that *Reward Framing* yields the largest accuracy gain among the core strategies. To test whether this effect depends on exact wording, we create six paraphrased Reward prompts that vary in incentive magnitude, placement, and phrasing strength. $V_{\text{Reward}} = \{R_1, R_2, R_3, R_4, R_5, R_6\}$. Applying each Reward variant to all N problems and querying T gives :

$$\mathcal{D}_{\text{Reward}} = \left\{ \left(\hat{q}_i^{(R_j)}, r_i^{(R_j)}, y_i^{(R_j)}, a_i^* \right) \middle| i = 1, \dots, N; \ R_j \in \mathcal{V}_{\text{Reward}} \right\}. \tag{7}$$

where $\hat{q}_i^{(R_j)}$ is the problem q_i wrapped with Reward variant R_j and \mathcal{V}_{Reward} the set of six Reward Framing variants. Table 1 summarizes the four core principles, and Table 2 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, as shown in Table 2. 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 Table 13. Intriguingly, we further notice that when the question is relatively short, the difference in accuracy is actually small.

Dataset configurations and scale. We construct *four* dataset families from the same N seeds, distinguishing single–principle sets from their unions:

Appendix D 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.

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

Table 2. 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 I denotes identity. Operator definitions: \mathcal{T}_{COND} : C (add conditionality); \mathcal{T}_{LOW} : $M \downarrow (reduce magnitude in USD)$; \mathcal{T}_{HIGH} : $M \uparrow (increase magnitude in USD)$; \mathcal{T}_{SUFFIX} : π =suffix (change placement); \mathcal{T}_{REASON} : $\rho \neq \emptyset$ (add reasoning cue, e.g., step-by-step).

(i) **Single-P-TTS** (per principle). For each $p \in \mathcal{P}_{core}$ we build a *separate* dataset

$$\mathcal{D}_{(p)}^{\text{single}} = \left\{ \left(q_i^{(p)}, y_i^{(p)}, r_i^{(p)}, a_i^* \right) \right\}_{i=1}^N, \quad \left| \mathcal{D}_{(p)}^{\text{single}} \right| = N = 90.$$

Thus, there are four Single-P-TTS datasets (one per principle). When we report "Single", we train one model per p (and report results per p or their mean, as specified in Sec. 4.3).

(ii) **Core-P-TTS (union of singles).** The core set is the (disjoint) union over all four principles:

$$\mathcal{D}_{\text{core}} = \bigsqcup_{p \in \mathcal{P}_{\text{core}}} \mathcal{D}_{(p)}^{\text{single}}, \qquad |\mathcal{D}_{\text{core}}| = 4N = 360.$$

(iii) Seed combined with the core P-TTS. We add the null-prompt (seed) $\mathcal{D}_{\text{seed}}$ and $\mathcal{D}_{\text{Reward}}$ to obtain

$$\mathcal{D}_{\text{seed+core}} = \mathcal{D}_{\text{seed}} \cup \mathcal{D}_{\text{core}}, \qquad |\mathcal{D}_{\text{seed+core}}| = 5N = 450.$$

(iv) **Full P-TTS.** Let \mathcal{V}_{Reward} denote the set of reward paraphrases K total variants applied to all seeds, producing \mathcal{D}_{Reward} . Because one variant is already used in the Core set, the additional Reward set has size (K-1)N. The full corpus is

$$\mathcal{D}_{\text{full P-TTS}} = \mathcal{D}_{\text{seed}} \cup \mathcal{D}_{\text{core}} \cup \mathcal{D}_{\text{Reward}}, \qquad |\mathcal{D}_{\text{full P-TTS}}| = (1 + 4 + (K - 1))N.$$

In our experiments we use 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. 3).

3.5. 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) [YLY+25] separately on each configuration from Sec. 3.4, following an SFT recipe adapted from s1 [MYS+25]. The student is trained to predict full assistant outputs (reasoning + answer) with token-level cross-entropy computed on assistant tokens only (user tokens masked). 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.

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

Models are trained to predict the full assistant output—reasoning trace and final answer. 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.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our P-TTS models on four public reasoning benchmarks: AIME24 [AIM24] (30 problems) and AIME25 [AIM25] (15 problems) from the American Invitational Mathematics Examination; AIME includes problems from algebra, arithmetic, geometry, number theory, combinatorics, and probability. MATH500 [HBK+21] is a 500-problem competition-math subset; we adopt the publicly released OpenAI selection used in prior work. GPQA-Diamond [RHS+24] 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 [GBB+21, BSS+24]. 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). In addition to these four 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 9, include Gaokao, Kaoyan, OlympiadBench [HLB+24], AMC23, GradeSchoolMath, and Minerva.

Baselines. We benchmark P-TTS against three categories of reasoning models. (i) Closed-source (API-only) models: OpenAI's o1 series [Ope24, Ope25] and Google's experimental Gemini 2.0 Flash Thinking variant [Clo24]. (ii) Open-weight models: DeepSeek-R1 series

[GYZ⁺25] and Qwen's QwQ-32B-preview [Qwe24, YLY⁺25] . (iii) Open-weight SFT models on Qwen2.5-Instruct with public data on openly available reasoning corpora: including Bespoke-Stratos-32B [Bes25], OpenThinker-32B [Tea25b, GMK⁺25], Sky-T1-32B-Preview [Tea25a], and the S1/S1.1 w/o BF [MYS⁺25]checkpoints.

Diversity Metrics. We compute two complementary metrics—semantic and surface—level—to quantify how each single-principle variant in \mathcal{D}_{core} adds information beyond the seed set (\mathcal{D}_{seed}). Semantic diversity (Diversity Gain). Following [YJS⁺23, Bil22], we compute diversity gain to quantify knowledge-level novelty. Given a seed dataset \mathcal{D}_{seed} and a new dataset \mathcal{D}_{core} , we define $DG = \frac{1}{M} \sum_{x_i \in \mathcal{D}_{core}} \min_{x_j \in \mathcal{D}_{seed}} ||f(x_i) - f(x_j)||_2^2$, where $f(\cdot)$ is an embedding function and $M = |\mathcal{D}_{core}|$. We use OpenAI's text-embedding-ada-002 as f for feature extraction. Higher values indicate greater semantic divergence from the base data. Surface—level diversity (trigram diversity). We compute trigram diversity [LYH⁺22], defined as the ratio of non-overlapping trigrams between two texts². We average this score over all sample pairs between each P-TTS principle variant in \mathcal{D}_{core} and its corresponding baseline instance in \mathcal{D}_{seed} .

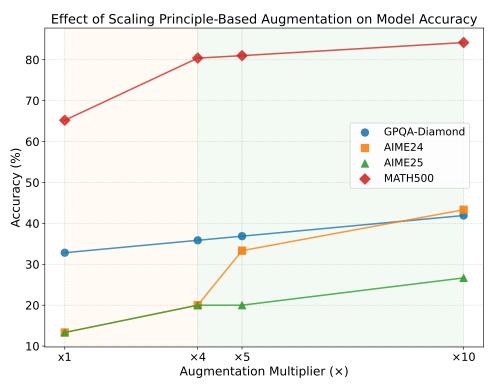


Figure 3. Accuracy improvement with increased principled augmentation on 7B model. We evaluate how model accuracy scales with the number of augmented training examples. Here, **x1** refers to P-TTS_{Reward} (90 examples), **x4** to P-TTS_{Core} (360 examples), **x5** to P-TTS_{Core+Orig} (450 examples), and **x10** to P-TTS_{Full1} (900 examples). Accuracy improves consistently across all evaluation sets with larger, principle-guided augmentations.

4.2. Teacher Model for Data Construction

Our objective is to construct a **compact yet high-quality** mathematical reasoning corpus for supervised fine-tuning. Specifically, we aim to identify the possible **smallest** training corpus that still yields the highest downstream accuracy. To obtain both full responses and explicit reasoning traces, we consider large language models that expose chain-of-thought generation through their public APIs. We benchmark three reasoning models: Claude-3-Opus, DeepSeek-R1, and OpenAI

² For texts x, y, $TD(x, y) = 1 - \frac{|Tri(x) \cap Tri(y)|}{|Tri(x) \cap Tri(y)|}$, where Tri(x) denotes the set of *distinct* word-level trigrams in x.

Omni-4 on four tasks. Claude-3-Opus and DeepSeek-R1 natively return aligned answer–reasoning pairs, while Omni-4 requires an augmented prompt to elicit full reasoning. From each model, we collect 90 answer–reasoning pairs and fine-tuned a Qwen2.5-7B-Instruct on the resulting dataset. Table 12 reports accuracy averaged over all four benchmarks. The Qwen2.5-7B-Instruct model fine-tuned on DeepSeek-R1 outputs consistently outperforms counterparts trained on Claude and Omni-4 outputs under the same small corpus. Based on this, we adopt **DeepSeek-R1** as the teacher for all subsequent data construction.

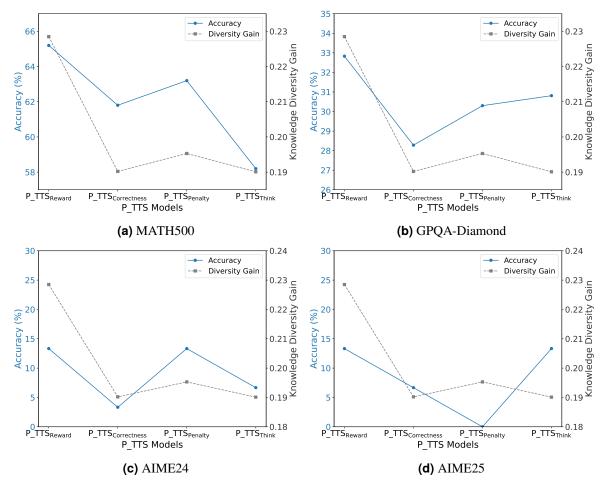


Figure 4. Knowledge Diversity Gain vs. Accuracy for different P-TTS variants across four benchmarks. We compare the trade-off between Accuracy (blue solid line, left y-axis) and Knowledge Diversity Gain (gray dashed line, right y-axis) on 7B model for four principled prompting strategies: Reward, Correctness, Penalty, and Think. Diversity Gain is computed relative to the original P-TTS baseline.

4.3. Ablation Studies

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

Single-P-TTS. To assess the impact of each principle independently, we fine-tune separate Qwen2.5-7B-Instruct models on the **Single P-TTS** subsets—90 examples per model—each applying only one core principle $p \in \mathcal{P}_{core}$. We compare these models to a baseline trained on the same seed problem without any instructional framing (P-TTS_{Seed}). As shown in Table 3, all **Single P-TTS** variants outperform the baseline on average. The Reward-based model (P-TTS_{Reward}) yields the highest overall gain (+6.67%), improving accuracy across all benchmarks. Penalty-based model (P-TTS_{Penalty}) also delivers strong results, especially on MATH500, though with a drop

on AIME25. Correctness-based model (P-TTS_{Correctness}) offers modest improvements, while Think-based model (P-TTS_{Think}) increases the average but underperforms on MATH500. These findings highlight that even minimal augmentations, i.e., just 90 principle-guided examples, can yield measurable improvements. Among the four principles, Reward and Penalty framings are the most effective when applied independently.

Model	#Ex.	AIME-24	AIME-25	MATH500	GPQA-D	Avg.
Sin	gle-Prin	ciple P-TTS A	blation (90 I	Examples Eacl	n)	
P-TTS _{Seed} (baseline)	90	3.33	6.67	60.20	27.78	24.50
P-TTS _{Reward}	90	13.33	13.33	65.20	32.83	31.17
P-TTS _{Correctness}	90	3.33	6.67	61.80	28.28	25.02
P-TTS _{Penalty}	90	13.33	0.00	63.20	30.30	26.71
P-TTS _{Think}	90	6.67	13.33	58.20	30.81	27.25
Pairwise Principl	e Ablati	on Centered	on Reward F	raming (180 I	Examples Ea	ch)
P-TTS _{Reward} ∪Penalty	180	10.00	20.00	75.20	32.32	34.38
P-TTS _{Reward} ∪Correctness	180	23.33	20.00	75.40	37.37	39.02
P-TTS _{Reward} ∪Think	180	13.33	13.33	72.80	31.82	32.82

Table 3. Accuracy (%) of Single and Pairwise Principle Ablation using 7B model. Top: Single-principle P-TTS variants each trained on 90 instructional prompts. Bottom: Pairwise ablations centered on Reward framing, trained on 180 prompts combining Reward with one other principle. All models use Qwen2.5-7B-Instruct fine-tuning.

Pairwise-P-TTS. To further evaluate the usefulness of each principle and whether they can be effectively combined, we explore the case where a principle that is not highly effective on its own might still contribute positively when paired with another. Specifically, since P-TTS_{Reward} shows the strongest performance in the **Single P-TTS** ablation, we fix the Reward framing and incrementally add one additional principle at a time, resulting in datasets of 180 examples. Table 3 shows the performance of each pairwise combination. We observe that combining Reward framing with either Correctness or Penalty significantly boosts performance across most benchmarks, particularly on MATH500 and AIME2024. Also, the P-TTS_{Reward∪Correctness} combination achieves the highest overall accuracy (39.02%), suggesting a synergistic effect between reward framing and correctness emphasis. In contrast, combining Reward framing with Thinking yields only modest improvements, indicating diminishing returns when both principles primarily influence the reasoning process.

Core-P-TTS. To assess the relative importance of each principle when used in combination, we conduct a leave-one-out ablation over the full Core P-TTS dataset \mathcal{D}_{core} (360 examples). We fine-tune a separate Qwen2.5-7B-Instruct model after removing one principle at a time, reducing the training set to 270 examples. As shown in Table 4, the exclusion of the Reward framing leads to the largest performance drop (from 39.06% to 35.40%), confirming its central role in driving improvements. Removing Correctness or Penalty framing causes moderate degradation, while the absence of Step-by-Step thinking has minimal or slightly positive effects. These results indicate that Reward-based cues are the most impactful when principles are used in combination, whereas Step-by-Step prompting contributes the least in multi-principle settings. Fig. 5 further supports this finding, showing that the incremental addition of principles leads to consistent accuracy improvements.

4.3.2. Measuring the Effects of Augmentation Size

Table 5 presents the results for three configurations: (1) P-TTS_{Core} with 360 examples, (2) P-TTS_{Core+Seed} with 450 examples, and (3) P-TTS_{Core+Seed+RewardVar} with 900 examples, which

Model	#Ex.	AIME2024	AIME2025	MATH500	GPQA-D	Avg.
P-TTS _{Core} (all 4)	360	20.00	20.00	80.40	35.86	39.06
P-TTS _{Core\Reward}	270	16.67	13.33	78.80	32.83	35.40
P-TTS _{Core\Correctness}	270	13.33	20.00	79.80	35.86	37.24
P-TTS _{Core\Penalty}	270	20.00	20.00	79.80	34.34	38.53
P-TTS _{Core\Think}	270	20.00	26.67	78.20	32.32	39.29

Table 4. 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).

represents the full dataset (core, seed, and paraphrastic reward variants). Performance improves consistently with dataset size: the model trained on the full 900-example dataset achieves an average accuracy of 49.03%, outperforming all other configurations and surpassing the 1k-example S1.1 baseline (38.99%). The largest average gain occurs between 450 and 900 examples (+6.23%), with improvements on all benchmarks. The increase from 360 to 450 examples is particularly notable on AIME24 (+13.33%), indicating that mixing seed and wrapped questions with the selected core principles is beneficial. Fig. 3 shows the same trend: scaling augmentation from ×1 to ×10 yields gains across all four test sets, especially on MATH500 and AIME24. These results demonstrate that principled prompt augmentation scales effectively and enables competitive performance with far fewer training examples than traditional supervised fine-tuning.

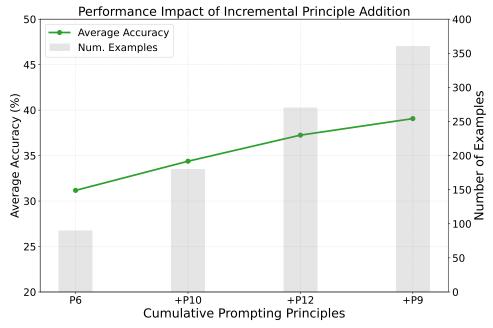


Figure 5. Impact of incremental principle addition on average accuracy. As additional prompting principles are cumulatively incorporated into training $(P6 \rightarrow +P10 \rightarrow +P12 \rightarrow +P9)$, both the number of training examples and model accuracy increase. Bars (right axis) denote the total number of examples after each addition; the green line (left axis) shows the resulting average accuracy across evaluation benchmarks. This highlights the compounding benefit of principled augmentation.

4.4. P-TTS Dataset Analysis

Diversity. The accuracy results in Table 3 show that each Single P-TTS variant outperforms the seed baseline on most benchmarks and on average. P-TTS_{Reward} yields the largest absolute gain, improving performance by approximately 6.7% on AIME25 and 5.0% on GPQA-DIAMOND,

Model	#Ex	AIME24	AIME25	MATH500	GPQA-D	Avg.
Qwen2.5-7B-Instruct (base)	_	13.33	6.67	76.40	36.36	33.19
S1-7B	1K	16.67	13.33	77.20	41.41	37.15
S1.1-7B	1K	13.33	20.00	81.20	41.41	38.99
P_TTS _{Core} -7B	360	20.00	20.00	80.40	35.86	39.07
P_TTS _{Core+ Seed} -7B	450	33.33↑	20.00	81.00↑	36.87↑	42.80^{\uparrow}
P-TTS _{Core+} Seed+ RewardVar-7B	900	43.33 [↑]	26.67 [↑]	84.20 [↑]	41.92 [↑]	49.03 [↑]

Table 5. 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**; ↑ marks a gain over the immediately preceding configuration.

while P-TTS_{Penalty} achieves the strongest improvement on MATH500. Even the weakest variant, P-TTS_{Correctness}, matches or surpasses the baseline on three out of four datasets. These accuracy trends correlate with the diversity analysis in Fig. 4. Single P-TTS *variants* with higher *semantic diversity* (Diversity Gain), such as P-TTS_{Reward} and P-TTS_{Penalty}, also exhibit larger accuracy improvements. Table 6 complements this by quantifying *surface-level diversity* (trigram diversity): P-TTS_{Reward} attains the highest scores across both final responses and reasoning traces, consistent with the observed performance gains. This pattern is consistent with prior findings that semantically diverse supervision and diversity-promoting augmentations improve generalization and robustness [BMZ⁺24, QSS⁺20].

Prompt Variant	Responses ↑	Reasoning ↑
Reward Framing	0.8363	0.9280
Correctness Framing	0.8227	0.9264
Penalty Framing	0.8223	0.9266
Step-by-Step Thinking	0.8295	0.9254

Table 6. Trigram diversity of teacher outputs under different prompting strategies. We report average lexical diversity (↑) for full responses and isolated reasoning traces. Reward framing yields the highest diversity, indicating stronger surface-level variation from original examples.

	Resp	onse To	kens	Reasoning Tokens			
Prompt Variant	Mean	Min	Max	Mean	Min	Max	
Seed	579.6	314	950	11,872	1129	31604	
Reward Framing	597.5	376	1,029	12,436	1745	34509	
Correctness Framing	575.7	322	962	10,463	1420	34514	
Penalty Framing	583.1	268	962	11,321	1133	29723	
Step-by-Step Thinking	577.6	312	936	11,113	1343	28270	

Table 7. Teacher output verbosity across prompting strategies. We report token count statistics (mean, min, max) for both full responses and extracted reasoning traces produced by the teacher model across single-principle prompt variants. These numbers reflect raw outputs used to construct the training data and illustrate how different instructional cues influence explanation length.

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 in Table 7. 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.

P-TTS-32B

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 3.83×10^{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: $m1 - m2 = -2.5 \log 10 (L1/L2)$ indeed $m1 - m2 = -2.5 \log 10 (L1/L2)$. So if we know the difference in magnitudes, we can find the ratio of luminosities. In our case, m1 (Sun) is +5, m2 (star) is 0. So m1 - m2 = 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 L1 is luminosity of Sun, L2 is luminosity of star. So m1 - m2 = 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 3.83×10^{35} ergs/s. Final Answer 3.83×10^{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 3.83×10^{35} ergs/s. Final Answer 3.83×10^{35}

Figure 6. Example of a P-TTS-32B response. See Appendix F for comparison with baseline model outputs.

4.5. Main Results

Overall Performance Comparison. We compare our P-TTS-32B model, trained on just 900 principle-augmented examples, against a diverse set of competitive baselines, including closed-source APIs, open-source models, and open-weight instruction-tuned variants of Qwen2.5-32B. While closed models like o1-preview and o1-mini perform strongly on certain tasks, P-TTS-32B outperforms them on AIME2024 and MATH500, and delivers competitive results on GPQA-Diamond despite using less data. Compared to open-weight models that finetuned on Qwen2.5-instaruct-32B like OpenThinker-32B (114K examples) and Bespoke-32B (17K), our model achieves superior or comparable performance across all benchmarks. P-TTS-32B achieves an average accuracy of 70.35%, exceeding all open-weight instruction-tuned baselines, and narrowing the gap with large-scale models like DeepSeek-R1, which require over 800K training examples. These findings highlight the efficiency of principled instructional data and demonstrate

Model	# Train Size	AIME2024	AIME2025	MATH500	GPQA-Diamond	Avg
	close	d-source mode	ls			
o1-preview[Ope25]	_	56.7	_	85.5	78.3	_
o1-mini [Ope25]	_	63.6	_	90.0	60.0	_
Gemini 2.0 Flash Think	_	60.0	_	_	_	_
	орег	ı-source model.	s			
Qwen2.5-32B-Instruct [YYZ ⁺ 24]	_	26.7	_	84.0	49.0	_
QwQ-32B [Tea25c]	_	50.0	_	90.6	54.5	_
DeepSeek-R1 [GYZ ⁺ 25]	≫800K	79.8	_	97.3	71.5	
DeepSeek-R1-Distill-Qwen-32B [GYZ ⁺ 25]	800K	72.6	_	94.3	62.1	_
Оре	n-weight & oper	n-data SFT on Q	Qwen2.5-Instru	ct		
OpenThinker-32B [Tea25b]	114K	66.0	53.3	90.6	61.6	67.9
Bespoke-32B [Bes25]	17K	63.3	_	93.0	58.1	_
Sky-T1-32B-Preview [Tea25a]	17K	43.3	_	82.4	56.8	_
S1-32B [MYS ⁺ 25]	1K	50.0	26.7	92.6	56.6	56.5
S1.1-32B [MYS+25]	1K	56.7	50.0	94.4	60.6	65.4
P-TTS-32B (ours)	90 →900	73.3	53.3	94.2	60.6	70.4

Table 8. Accuracy comparison of 32B-scale models on four reasoning benchmarks: AIME2024, AIME2025, 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 via principled prompting strategies to generate up to 900 training examples. Despite the small training size, P-TTS-32B achieves competitive or superior performance, outperforming several models trained on datasets with hundreds of thousands of examples. *Notes:* Results for *Gemini* and *Qwen* are taken from [MYS⁺25] (we follow their evaluation settings).

Model	OlympiadBench	Gaokao	Kaoyan	Minerva	GradeSchool	AMC23	Avg.
OpenAI-o1-preview	52.1	62.1	51.5	47.1	62.8	81.8	59.6
Qwen2.5-32B-Instruct	45.3	72.1	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	59.3	46.7	61.4	77.5	55.8
P-TTS (Ours)	63.9	51.9	52.3	51.5	53.8	87.5	60.2

Table 9. Zero-shot generalization accuracy (%) on out-of-domain reasoning benchmarks. P-TTS is trained only on AIME22–24.

that high accuracy can be attained through lightweight, targeted supervision.

Cross-Domain and Multilingual Generalization. Although P-TTS is trained only on 90 English AIME-style problems (AIME22–24), it exhibits robust zero-shot transfer to benchmarks that differ in *language*, *curriculum*, and *problem format*. In Table 9, we evaluate on Chinese exam datasets (Gaokao, Kaoyan), U.S. competition and school math (OlympiadBench, AMC23, GradeSchoolMath), and scientific quantitative reasoning (Minerva). These tasks introduce shifts in linguistic style (Chinese vs. English), assessment design (competition vs. entrance exams vs. textbook problems), and reasoning presentation (concise Olympiad proofs vs. step-by-step classroom narratives). Despite no multilingual supervision and no direct exposure to these benchmarks during finetuning, P-TTS attains competitive accuracy across the board, narrowing the gap with models trained on one to two orders of magnitude more data. This suggests that principled prompt augmentation through varying instructional framing and exemplar structure encourages *prompt-space coverage* that translates into language and curriculum robustness, elicits knowledge from pre-trained models, rather than overfitting to a single benchmark family.

Scaling across model sizes. We evaluate how P-TTS performance scales with model size across the 7B, 14B, and 32B model variants. At every scale, P-TTS outperforms both S1 and S1.1 on most benchmarks, as shown in table 11. For instance, P-TTS-7B surpasses S1.1-7B by +30.0% on

Benchmark	7B Models		14B M	Iodels	32B Models			
	P-TTS	S1	S1.1	P-TTS	S1.1	P-TTS	S1	S1.1
AIME2024	43.33	16.67	13.33	53.33	33.33	73.33	50.00	56.70
AIME2025	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 10. Accuracy (%) on Four Benchmarks with Grouped Model Sizes. Each group shows results for core M1, S1, and S1.1.

AIME2024 and achieves comparable results on MATH500 with fewer examples. At 14B level, P-TTS continues to lead across all tasks, reaching 53.33% on AIME2024 and 90.4% on MATH500. Notably, at the 32B scale, P-TTS achieves 73.33% on AIME2024 and 94.20% on MATH500, outperforming both S1 and S1.1 despite their larger training sizes. 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 performance across diverse benchmarks.

5. 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 configurations with augmentation multipliers $m \in \{1, 4, 5, 10\}$, P-TTS consistently improves supervised reasoning relative to the null prompt and *even starting from only 90 seeds*, matches or surpasses models trained on substantially larger, static datasets. P-TTS demonstrates that *instructional prompt reformulation* is a powerful and overlooked scaling lever. With only 90 carefully chosen seeds, principled wrapping and its paraphrased variants produce supervision that competes with (and at times exceeds) 1K-shot baselines in the wild, substantially lowering the data curation burden for robust LLM reasoning.

Future Work. We see several promising directions: (1) adaptive, per-instance selection of instructional wrappers via learned policies; (2) integration with retrieval and verifier/reranker pipelines to couple wrapper diversity with factual grounding; (3) principled scheduling of wrapper mixtures over training epochs to mimic curriculum learning; and (4) systematic study of wrapper transfer across tasks, languages, and modalities.

6. Limitations

There are several potential limitations. First, the evaluation is concentrated on math-style problems with single numeric answers (AIME22–24), so external validity to open-ended, multimodal, or multilingual reasoning remains to explore further. Second, P-TTS depends on a single family 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. Third, 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. Fourth, despite contamination mitigation, residual leakage from publicly available AIME material cannot be conclusively ruled out. Finally, P-TTS

requires compute at data collection through inference generation (via wrapper ensembles).

References

- AAA⁺23. Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. (cit. on pp. 2 and 5.)
 - AIM24. AIME. 2024 aime i. https://artofproblemsolving.com/wiki/index.php/2024_AIME_I, 2024. Art of Problem Solving Wiki, accessed July 2025. (cit. on p. 8.)
 - AIM25. AIME. 2025 aime i. Art of Problem Solving Wiki, 2025. Held February 6, 2025. URL: https://artofproblemsolving.com/wiki/index.php/2025_AIME_I. (cit. on p. 8.)
 - Ant25. Anthropic AI. Claude 3.7 sonnet and claude code. Anthropic blog, February 2025. First hybrid reasoning large language model generally available. URL: https://www.anthropic.com/news/claude-3-7-sonnet. (cit. on pp. 2 and 3.)
 - Art. Art of Problem Solving (AoPS). Aime problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions. (cit. on p. 5.)
 - Bes25. Bespoke Labs. Bespoke-stratos-32b. https://huggingface.co/bespokelabs/Bespoke-Stratos-32B, 2025. Hugging Face model card, Apache-2.0 license. Fine-tuned Qwen2.5-32B-Instruct on Bespoke-Stratos-17k dataset derived via DeepSeek-R1 distillation. (cit. on pp. 3, 9, and 15.)
 - Bil22. Jeff Bilmes. Submodularity in machine learning and artificial intelligence. *arXiv preprint* arXiv:2202.00132, 2022. (cit. on p. 9.)
- BMR⁺20. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. (cit. on p. 3.)
 - BMS23. Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4. *arXiv preprint arXiv:2312.16171*, 2023. (cit. on pp. 2, 3, 4, and 5.)
- BMZ⁺24. Alexander Bukharin, Jiachang Mu, Zhengbao Zhang, Seyeon Lee, Kai-Wei Chang, Noah A. Smith, and Daniel Khashabi. Data diversity matters for robust instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2871–2885, 2024. URL: https://aclanthology.org/2024.findings-emnlp.195. (cit. on p. 13.)
- BSS⁺24. Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024. (cit. on p. 8.)
 - Clo24. Google Cloud. Flash thinking with generative ai. https://cloud.google.com/vertex-ai/generative-ai/docs/thinking, 2024. (cit. on p. 8.)
- FLD18. Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint* arXiv:1805.04833, 2018. (cit. on p. 3.)
- GBB+21. 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. EleutherAI/Imevaluation-harness: Evaluation Harness for Language Models, 2021. doi:10.5281/zenodo.5371628. (cit. on p. 8.)
- GMK⁺25. Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025. (cit. on pp. 3 and 9.)
- GYZ⁺25. Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. (cit. on pp. 3, 5, 9, and 15.)
- HBD⁺19. Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019. (cit. on p. 3.)
- HBK+21. Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint* arXiv:2103.03874, 2021. (cit. on p. 8.)
- HLB⁺24. 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. OlympiadBench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024. URL: https://arxiv.org/abs/2402.14008, doi: 10.48550/arXiv.2402.14008. (cit. on p. 8.)
- HYS⁺25. Shulin Huang, Linyi Yang, Yan Song, Shuang Chen, Leyang Cui, Ziyu Wan, Qingcheng Zeng, Ying

- Wen, Kun Shao, Weinan Zhang, et al. Thinkbench: Dynamic out-of-distribution evaluation for robust llm reasoning. *arXiv preprint arXiv:2502.16268*, 2025. (cit. on p. 5.)
- JKL⁺24. Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. (cit. on p. 3.)
- KMH⁺20. Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv* preprint arXiv:2001.08361, 2020. (cit. on p. 3.)
 - Lab25. Bespoke Labs. Bespoke-stratos-17k: A synthetic reasoning dataset of questions, reasoning traces, and answers. Hugging Face Dataset, 2025. Derived from DeepSeek-R1 via the Sky-T1 pipeline using Bespoke Curator. URL: https://huggingface.co/datasets/bespokelabs/Bespoke-Stratos-17k. (cit. on p. 3.)
- p. 3.)
 LCGS25. Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025. (cit. on p. 3.)
- LYH⁺22. Wenhao Li, Xiaoyuan Yi, Jinyi Hu, Maosong Sun, and Xing Xie. Evade the trap of mediocrity: Promoting diversity and novelty in text generation via concentrating attention. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10834–10858, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL: https://aclanthology.org/2022.emnlp-main.745, doi:10.18653/v1/2022.emnlp-main.745. (cit. on p. 9.)
- MGH⁺24. Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, and Aixin Sun. Sciagent: Tool-augmented language models for scientific reasoning. In *EMNLP*, 2024. (cit. on p. 2.)
- MPWC23. Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121, 2023. (cit. on p. 3.)
- MYS⁺25. Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025. (cit. on pp. 3, 4, 7, 9, and 15.)
 - Ope24. OpenAI. Learning to reason with llms, 2024. URL: https://openai.com/index/learning-to-reason-with-llms/. (cit. on p. 8.)
 - Ope25. OpenAI. Openai o3-mini. https://openai.com/index/openai-o3-mini/, 2025. (cit. on pp. 8 and 15.)
- QSS⁺20. Yanru Qu, Dinghan Shen, Yelong Shen, Sandra Sajeev, Jiawei Han, and Weizhu Chen. Coda: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding. *arXiv preprint arXiv:2010.08670*, 2020. (cit. on p. 13.)
- Qwe24. Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown. https://qwenlm.github.io/blog/qwq-32b-preview/, November 2024. QwQ-32B-Preview is an experimental reasoning model with open weights. (cit. on p. 9.)
- RHS⁺24. David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. (cit. on p. 8.)
- RNS⁺18. Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. (cit. on p. 2.)
- TAB+23. Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. (cit. on pp. 2 and 3.)
- Tea25a. NovaSky Team. Sky-t1: Fully open-source reasoning model with o1-preview performance in \$450 training cost, 2025. URL: https://novasky-ai.github.io/posts/sky-t1. (cit. on pp. 9 and 15.)
- Tea25b. OpenThoughts Team. Openthinker-32b. https://huggingface.co/open-thoughts/OpenThinker-32B, 2025. (cit. on pp. 9 and 15.)
- Tea25c. Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, 2025. (cit. on p. 15.)
- WWS⁺22a. Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv* preprint arXiv:2203.11171, 2022. (cit. on p. 3.)
- WWS⁺22b. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. (cit. on pp. 2 and 3.)
 - YHX⁺25. Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025. (cit. on p. 4.)
 - YJS⁺23. Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023. (cit. on pp. 3 and 9.)
 - YLY+25. An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,

- Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025. (cit. on pp. 7 and 9.)
- YYZ⁺24. An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. arXiv preprint arXiv:2412.15115, 2024. (cit. on p. 15.)
- ZAD+25. Alan Zhu, Parth Asawa, Jared Quincy Davis, Lingjiao Chen, Boris Hanin, Ion Stoica, Joseph E Gonzalez, and Matei Zaharia. Bare: Leveraging base language models for few-shot synthetic data generation. *arXiv preprint arXiv:2502.01697*, 2025. (cit. on p. 5.) ZCH+25. Tong Zheng, Lichang Chen, Simeng Han, R Thomas McCoy, and Heng Huang. Learning to reason
- via mixture-of-thought for logical reasoning. arXiv preprint arXiv:2505.15817, 2025. (cit. on p. 3.)
- ZGZG25. Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. arXiv preprint arXiv:2504.12216, 2025. (cit. on p. 3.)

Appendix

A. Extended Experiments on \mathcal{D}_{core}

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

Benchmark	7B Models			14B N	Models		32F	32B Models		
	P-TTS _{Core}	S1	S1.1	P-TTS _{Core}	S1.1	_	P-TTS _{Core}	S1	S1.1	
AIME2024	20.00	16.67	13.33	36.67	33.33	_	56.67	56.70	56.70	
AIME2025	20.00	13.33	20.00	33.33	33.33	_	46.67	26.70	60.00	
MATH500	80.40	77.20	81.20	89.80	91.60	_	94.00	93.00	95.40	
GPQA-Diamond	35.86	41.41	41.41	45.96	51.01	_	53.03	59.60	63.60	

Table 11. Accuracy (%) on Four Benchmarks with Grouped Model Sizes. Each group shows results for core P-TTS, S1, and S1.1.

B. Training Details

We fine-tuned the Qwen2.5-Instruct family at three scales—7B, 14B, and 32B—using our P-TTS datasets. 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 = 1×10^{-4}) and a base learning rate of 1×10^{-5} , warmed up linearly for the first 5% of steps and decayed to zero following a cosine schedule. Training was conducted in bfloat16 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.

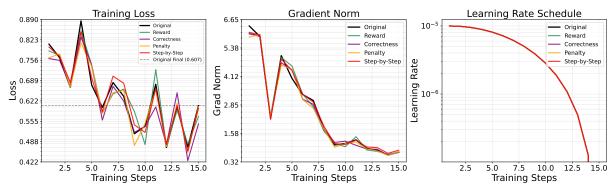


Figure 7. Training dynamics of P-TTS-32B.

C. Additional Results on Other Models

We also report benchmark accuracy for additional baselines beyond the Qwen2.5-Instruct family. Table 12 presents results for Claude, O4-mini-high, and hybrid settings with DeepSeek.

Model	Data Points	AIME24	AIME25	GPQA-Diamond	MATH
Claude	810	16.67	0.00	34.34	72.80
O4-mini-high	810	36.67	0.00	41.41	60.00
DeepSeek + Claude	1620	40.00	20.00	37.88	83.20
DeepSeek + O4mini	1620	36.67	13.33	35.86	74.00

Table 12. Performance comparison across benchmarks. Values represent accuracy (%) per dataset.

D. Evaluation of Reward Framing Variants

We evaluate variants of the reward-framing across different models. Table 13 summarizes the accuracy of O1-mini, Gemini, and DeepSeek on a fixed subset of math problems (AIME 2022–2024) under each variant.

Rewards	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 13. 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.

E. Example from $\mathcal{D}_{\text{full-P-TTS}}$

Original

Question

Let O = (0,0), $A = \left(\frac{1}{2},0\right)$, and $B = \left(0,\frac{\sqrt{3}}{2}\right)$ 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), $A=\left(\frac{1}{2},0\right)$, and $B=\left(0,\frac{\sqrt{3}}{2}\right)$, 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 θ 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 $\left(\frac{1}{8}, \frac{3\sqrt{3}}{8}\right)$.

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 8. 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), $A = \left(\frac{1}{2},0\right)$, and $B = \left(0,\frac{\sqrt{3}}{2}\right)$ 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), $A = \left(\frac{1}{2},0\right)$, and $B = \left(0,\frac{\sqrt{3}}{2}\right)$, 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 $\left(\frac{1}{8}, \frac{3\sqrt{3}}{8}\right)$. Squaring the distance from O to C:

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

- 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, sqrt(3)/2). Then there's this family of segments 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

Step-by-Step Reasoning

Question

Think step by step: Let O = (0,0), $A = \left(\frac{1}{2},0\right)$, and $B = \left(0,\frac{\sqrt{3}}{2}\right)$ 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), $A = \left(\frac{1}{2},0\right)$, and $B = \left(0,\frac{\sqrt{3}}{2}\right)$, 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 . . .

Penalty Framing

Question

You will be penalized if you provide the wrong answer. Let O = (0,0), $A = \left(\frac{1}{2},0\right)$, and $B = \left(0,\frac{\sqrt{3}}{2}\right)$ 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), $A=\left(\frac{1}{2},0\right)$, and $B=\left(0,\frac{\sqrt{3}}{2}\right)$, 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 $\left(\frac{1-t}{2}, \frac{\sqrt{3}}{2}t\right)$.
- 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 $C\left(\frac{1-t}{2}, \frac{\sqrt{3}}{2}t\right)$ 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
- 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 $\left(\frac{1}{8}, \frac{3\sqrt{3}}{8}\right)$.
- 6. **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, 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

Correctness Emphasis

Question

Your Task is to solve the following: Let O = (0,0), $A = \left(\frac{1}{2},0\right)$, and $B = \left(0,\frac{\sqrt{3}}{2}\right)$ 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), $A = \left(\frac{1}{2},0\right)$, and $B = \left(0,\frac{\sqrt{3}}{2}\right)$, 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.

- 1. **Parametrization of \overline{AB} **: The parametric equations for \overline{AB} are $x = \frac{1-t}{2}$ and $y = \frac{\sqrt{3}}{2}t$ for $t \in [0, 1]$.
- 2. **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 $x^{2/3} + y^{2/3} = 1$.
- 3. **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 $t = \frac{3}{4}$.
- 4. **Coordinates of C^{**} : Substituting $t = \frac{3}{4}$ into the parametric equations:

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

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

F. P-TTS-32B vs. S1-32B

```
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 3.83×10<sup>33</sup> ergs/s
                                                                                                   S1-32B
                       P-TTS-32B
Okay, so I need to find the luminosity of a star with
                                                                      think
magnitude 0, given that the Sun's absolute magnitude is +5
                                                                       The problem asks for the luminosity of a star with a given
and its luminosity is 3.83×10<sup>33</sup> ergs/s. Hmm, let's start by
                                                                       apparent magnitude, given the absolute magnitude of the
recalling the relationship between magnitude and
                                                                      Sun and the Sun's luminosity.
luminosity
                                                                       We are given:
First, absolute magnitude is a measure of a star's intrinsic
                                                                       Sun's absolute magnitude, M_sun = +5
brightness, as it would be seen from a standard distance.
                                                                       Magnitude of the star, m = 0
The Sun's absolute magnitude is given as +5. The star we're
                                                                       Luminosity of the Sun, L_sun = 3.83×10<sup>33</sup> ergs/s
interested in has an absolute magnitude of 0. The
                                                                      We need to find the luminosity of the star. L star.
difference in magnitude between the two stars is 5 - 0 = 5
                                                                       The relationship between absolute magnitude and
 magnitudes.
                                                                       luminosity is given by:
In astronomy, the difference in magnitudes relates to the
                                                                       M = -2.5 \cdot log_{10}(L/L_0)
ratio of luminosities. The formula is:
                                                                       where M is the absolute magnitude, L is the luminosity, and
m1 - m2 = -2.5·log<sub>10</sub>(L1/L2)
                                                                       L₀ is a reference luminosity.
Where m1 and m2 are the magnitudes, and L1 and L2 are the
                                                                      For the Sun, we have:
corresponding luminosities. Wait, but here we have
                                                                       M_sun = -2.5 \cdot log_{10}(L_sun/L_0)
absolute magnitudes, so this formula should apply. Let me
                                                                       5 = -2.5 \cdot \log_{10}(3.83 \times 10^{33} / L_0)
                                                                      We can solve this equation for L₀:
confirm
Yes, the formula for the difference in magnitudes is indeed
                                                                       5/(-2.5) = \log_{10}(3.83 \times 10^{33} / L_0)
m1 - m2 = -2.5 \cdot log_{10}(L1/L2). So if we know the difference in
                                                                       -2 = \log_{10}(3.83 \times 10^{33} / L_0)
magnitudes, we can find the ratio of luminosities.
                                                                       10-2 = 3.83×1033 / Lo
                                                                       0.01 = 3.83 \times 10^{33} / L_0
In our case, m1 (Sun) is +5, m2 (star) is 0. So m1 - m2 = 5 - 0 =
                                                                       L_0 = 3.83 \times 10^{33} / 0.01
Plugging into the formula:
                                                                       L<sub>o</sub> = 3.83×10<sup>33</sup> × 100
                                                                       L<sub>0</sub> = 3.83×10<sup>35</sup> ergs/s
5 = -2.5 \cdot \log_{10}(L_sun/L_star)
Wait, hold on. Let me make sure I have the formula right.
                                                                       Now, we need to find the luminosity of the star. The
The formula is m1 - m2 = -2.5·log_{10}(L1/L2). So if m1 is the
                                                                       apparent magnitude m is related to the absolute magnitude
magnitude of the first star (Sun), m2 is the magnitude of the
                                                                      M by the distance modulus:
second star (star of interest). Then L1 is luminosity of Sun,
                                                                       m - M = 5·log10(d) - 5
L2 is luminosity of star.
                                                                       where d is the distance to the star in parsecs.
So m1 - m2 = 5 - 0 = +5.
                                                                       The absolute magnitude of the star can be found using the
Therefore:
                                                                       same formula as for the Sun:
5 = -2.5 \cdot \log_{10}(L_sun/L_star)
                                                                       M_star = -2.5 \cdot log_{10}(L_star/L_0)
Let me solve for log10(L_sun/L_star):
                                                                      We are given the apparent magnitude of the star, m = 0. We
                                                                      need to find the absolute magnitude of the star. We can use
Divide both sides by -2.5:
5 / (-2.5) = log10(L_sun/L_star)
                                                                      the fact that the apparent magnitude of the Sun is about
Which is -2 = \log_{10}(L_sun/L_star)
Therefore, L_sun/L_star = 10-2 = 0.01
                                                                       Let's use the relationship between apparent magnitude,
So L star = L sun / 0.01 = L sun × 100
                                                                      absolute magnitude, and distance. For the Sun, which is at a
Therefore, the luminosity of the star is 100 times that of the
                                                                      distance of approximately 1 astronomical unit (AU), we
Given that the Sun's luminosity is 3.83×10<sup>33</sup> ergs/s, then the
                                                                       m_sun - M_sun = 5 \cdot log_1(d_sun) - 5
star's luminosity is 3.83×10<sup>33</sup> × 100 = 3.83×10<sup>35</sup> ergs/s.
                                                                       The average distance from Earth to the Sun is about
Wait, let me check that calculation again. 3.83×10³³
                                                                      1.496×10<sup>13</sup> cm.
multiplied by 100 is indeed 3.83×10<sup>35</sup>. Yes.
                                                                       1 parsec = 3.086×1018 cm.
                                                                       d_sun in parsecs = (1.496 \times 10^{13})/(3.086 \times 10^{18}) \approx 4.85 \times 10^{-6}
But let me verify the steps again to make sure I didn't make
a mistake.
Starting with the magnitude difference: Sun's absolute
                                                                       Using the apparent magnitude of the Sun, m_sun ≈ -26.74:
magnitude is +5, star's is 0, so difference is +5.
                                                                       -26.74 - 5 = 5·log<sub>10</sub>(4.85×10<sup>-6</sup>) - 5
Using the formula m1 - m2 = -2.5·log_{10}(L1/L2)
                                                                        -31.74 = 5 \cdot (\log_{10}(4.85) + \log_{10}(10^{-6})) - 5
Plugging in: 5 = -2.5 \cdot \log_{10}(L_sun/L_star)
                                                                       -31.74 = 5·(log<sub>10</sub>(4.85) - 6) - 5
                                                                       -31.74 = 5·log<sub>10</sub>(4.85) - 30 - 5
Divide both sides by -2.5: 5 / (-2.5) = -2 = log_{10}(L_sun/L_star)
Therefore, L_{sun}/L_{star} = 10^{-2} = 0.01
                                                                       -31.74 = 5×0.6856 - 35
Thus, L_star = L_sun / 0.01 = L_sun × 100
                                                                       -31.74 = 3.428 - 35
Therefore, 3.83×10<sup>33</sup> × 100 = 3.83×10<sup>35</sup> ergs/s.
                                                                       -31.74 = -31.572
                                                                       This is close enough considering the approximations used.
That seems correct. So the luminosity of the star is
3.83×10³⁵ ergs/s.
                                                                      Now, let's consider the star with magnitude 0. We need to
But let me check if there's another way to approach this to
                                                                       find its absolute magnitude. We can use the Sun as a
confirm.
                                                                      reference.
```

Figure 9. Qualitative comparison between P-TTS-32B and the baseline model.

G. Comparative Example: Original vs. Reward-Framed Response

Problem

Let ABCD be a parallelogram with $\angle BAD < 90^{\circ}$. A circle is tangent to \overline{DA} , \overline{AB} , and \overline{BC} and intersects diagonal \overline{AC} at points P and Q with AP = 3, PQ = 9, and QC = 16. Suppose that $AP^2 + 3PQ^2 + 9QC^2 = 16n$. Show that the area of ABCD can be written as $\frac{m}{n}\sqrt{n}$ where $m, n \in \mathbb{Z}_{>0}$ and n is squarefree. Find m + n.

Original

Response: Response: Given a triangle ABC with sides AB and BC, we have two configurations of circles: eight circles of radius 34 and 2024 circles of radius 1, each arranged sequentially tangent with two circles tangent to AB and BC respectively. We need to find the inradius of triangle ABC.

Key steps: 1. **Understanding the Configuration**: The circles are arranged in a chain, sequentially tangent, with two circles tangent to AB and BC. This implies a relationship between the number of circles, their radii, and the angle at vertex B. 2. **Angle Bisector and Inradius**: The inradius R of the triangle is related to the distance from the incenter to the sides. The chain of circles forms a path along the angle bisector of $\angle ABC$. 3. **Relating Configurations**: The product of the number of circles and their radii for both configurations must be proportional to the inradius. The key insight is that the ratio of the number of circles (253) and the inverse ratio of the radii (34) gives the inradius. 4. **Calculating the Inradius**: The inradius *R* is found by recognizing the proportional relationship between the two configurations. The ratio of the number of circles (2024/8 = 253) and the ratio of the radii (34/1 = 34) leads to the inradius being $\frac{253}{34}$.

Thus, the inradius R is $\frac{253}{34}$, and the sum of the numerator and denominator is 253 +

Reward Framing

Response: Given triangle ABC with sides AB and BC, we have two configurations of circles: eight circles of radius 34 and 2024 circles of radius 1, each sequentially tangent and tangent to AB and BC respectively. We need to find the inradius of triangle ABC.

1. Key Relationship: - The inradius R of triangle ABC can be related to the radius r and the number of circles N through the angle θ at vertex B. - The formula derived is: $R = r + 2r(N-1)\sin(\theta/2)$.

2. Setting Up Equations: - For the configuration with 8 circles of radius 34:

$$R = 34 + 2 \cdot 34 \cdot 7 \cdot \sin(\theta/2)$$

3. Equating the Expressions: - Equate the two expressions for *R*:

$$34 + 476\sin(\theta/2) = 1 + 4046\sin(\theta/2)$$

•••

4. Calculating the Inradius: - Substitute $\sin(\theta/2) = \frac{11}{1190}$ back into the equation for *R* using the smaller radius configuration:

$$R = 1 + 4046 \cdot \frac{11}{1190}$$

- Simplify the calculation:

5. Final Result: - The inradius R is $\frac{192}{5}$, where m = 192 and n = 5. Thus, m + n = 197.

197

34 = 287.

287

Reasoning Trace: Okay, let me try to work through this problem step by step. The problem is about a triangle ABC with sides AB and BC. There are eight circles of radius 34 that are sequentially tangent, and two of these circles are tangent to AB and BC respectively. Similarly, 2024 circles of radius 1 can be arranged in the same manner. We need to find the inradius of triangle ABC, expressed as a reduced fraction m/n, and then compute m + n. ... However, this is a guess based on the ratio. The problem likely involves this ratio, leading to the answer 287. **Final Answer** 287

Reasoning Trace: Alright, let me try to wrap my head around this problem. So, we have triangle ABC with sides AB and BC. There are eight circles of radius 34 that are sequentially tangent, meaning each one is tangent to the next, and the first and last ones are tangent to AB and BC respectively. Similarly, we can arrange 2024 circles of radius 1 in the same manner. The question is asking for the inradius of triangle ABC, expressed as a reduced fraction m/n, and then we need to find m + n. ... So, 22,253 / 595 = 37 + 2/5 = 37.4

So, 22,253 / 595 = 37 + 2/5 = 37.4Therefore, R = 1 + 37.4 = 38.4 = 192/5. Thus, the inradius is 192/5, so m=192, n=5, m + n=197.

This seems to check out, and the steps make sense. Therefore, the answer is 197. **Final Answer** 197