Training-Free Test-Time Contrastive Learning for Large Language Models

Kaiwen Zheng[1*] **Kai Zhou**[1*] **Jinwu Hu**[1,2*] **Te Gu**[1] **Mingkai Peng**[1] **Fei Liu**[1†]
[1]South China University of Technology, [2]Pazhou Laboratory

Abstract

Large language models (LLMs) demonstrate strong reasoning capabilities, but their performance often degrades under distribution shift. Existing test-time adaptation (TTA) methods rely on gradient-based updates that require white-box access and need substantial overhead, while training-free alternatives are either static or depend on external guidance. In this paper, we propose Training-Free Test-Time Contrastive Learning (**TF-TTCL**), a training-free adaptation framework that enables a frozen LLM to improve online by distilling supervision from its own inference experiences. Specifically, TF-TTCL implements a dynamic "Explore-Reflect-Steer" loop through three core modules: 1) Semantic Query Augmentation first diversifies problem views via multi-agent role-playing to generate different reasoning trajectories; 2) Contrastive Experience Distillation then captures the semantic gap between superior and inferior trajectories, distilling them into explicit textual rules; and 3) Contextual Rule Retrieval finally activates these stored rules during inference to dynamically steer the frozen LLM toward robust reasoning patterns while avoiding observed errors. Extensive experiments on closed-ended reasoning tasks and open-ended evaluation tasks demonstrate that TF-TTCL consistently outperforms strong zero-shot baselines and representative TTA methods under online evaluation. Code is available at https://github.com/KevinSCUTer/TF-TTCL.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable reasoning and problem-solving capabilities (Achiam et al., 2023; Guo et al., 2025). However, the previous "train-once, deploy-anywhere" paradigm faces a fundamental limitation: the static parameters of a frozen model often struggle to generalize to out-of-distribution queries or complex reasoning tasks in dynamic data streams. To address this, recent research has shifted toward Test-Time Adaptation (TTA), which adapts the model on the fly using test instances to bridge the distribution gap (Wang et al., 2021; Niu et al., 2022; Hu et al., 2025a). This paradigm underscores the need for models that can learn continuously from their own inference experiences.

However, effective test-time learning remains challenging in practice. Most existing TTA methods rely on *gradient-based parameter updates* (Wang et al., 2021; Hardt and Sun, 2024; Hu et al., 2025a; Zuo et al., 2025), which assume white-box access to model internals and introduce non-negligible computational and memory overhead during inference. These assumptions limit their applicability to modern, user-facing LLM deployment scenarios, where models are typically frozen and accessed as black boxes (*e.g.*, via APIs).

Training-free alternatives avoid parameter updates but introduce a different limitation. Static prompting strategies, such as Chain-of-Thought (CoT) (Wei et al., 2022), lack the flexibility to adapt reasoning to specific test instances. Conversely, dynamic approaches like Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Yao et al., 2023b) or feedback-driven optimization (Huang et al., 2023; Cai et al., 2025) rely heavily on external knowledge guidance. These methods require curated knowledge databases or ground-truth verifiers (*e.g.*, unit tests), which are not always readily available in real-world deployment. These limitations reveal a fundamental gap: current test-time adaptation paradigms either depend on parameter updates or assume access to external guidance, limiting their applicability to black-box LLMs.

This gap motivates the need for a training-free adaptation paradigm. The primary challenge is *extracting reliable error signals from the frozen*

[*]Equal contribution. Email: kaiwenzhenggz@gmail.com, kayjoe0723@gmail.com, fhujinwu@gmail.com
[†]Corresponding author. Email: feiliu@scut.edu.cn

Findings of the Association for Computational Linguistics: ACL 2026, pages 29644–29676

Table 1: Comparison of different test-time paradigms. **TF-TTCL (Ours)** is a gradient-free adaptation framework capable of online evaluation, requiring neither source data nor an external knowledge.

Paradigms	External Knowledge	Source Data	Gradient-Free	Online
Retrieval Augmentation Generation (Lewis et al., 2020)	✓	✗	✓	✓
Test-Time Adaptation (Wang et al., 2021)	✗	✗	✗	✓
Test-Time Training (Hardt and Sun, 2024)	✓	✓	✗	✓
Test-Time Reinforcement Learning (Zuo et al., 2025)	✗	✗	✗	✗
Test-Time Learning (Hu et al., 2025a)	✗	✗	✗	✓
Reasoning-Bank (Ouyang et al., 2025)	✓	✗	✗	✓
Training-Free Test-Time Contrastive Learning (Ours)	✗	✗	✓	✓

model's own output without external guidance. We draw inspiration from human cognitive processes, specifically reflective learning from errors (Schön, 1983). Such reflection can arise from internal comparison even in the absence of immediate external feedback, aligning with the core principle of *contrastive learning* (Chen et al., 2020): while ground truth is unavailable, the relative semantic gap between a model's superior and inferior outputs contains rich supervisory information. Crucially, instead of updating parameters, we distill these gaps into explicit textual rules. Stored in memory, these rules act as "semantic gradients". They dynamically guide the frozen LLM to reinforce positive patterns and avoid past errors in online evaluation.

In this paper, we propose Training-Free Test-Time Contrastive Learning (TF-TTCL), a framework that enables frozen LLMs to self-improve online through a dynamic "Explore-Reflect-Steer" loop. TF-TTCL first employs a Semantic Query Augmentation module, where multi-agent role-playing emulates the data augmentation effect of contrastive learning: a TEACHER generates high-confidence anchor answers from the original query, while a TUTOR introduces semantic variations via query rewriting, encouraging the STUDENT to explore diverse reasoning paths. The resulting outputs are then distilled by a Contrastive Experience Distillation mechanism, which organizes responses according to consistency and uncertainty, extracts contrastive positive and negative signals, and summarizes them as explicit rules stored in an experience rule repository. During online evaluation, incoming queries are guided by a Contextual Rule Retrieval strategy that activates relevant rules to steer the frozen LLM toward effective reasoning patterns while avoiding previously observed errors. Our main contributions are summarized as follows:

- **Novel Training-Free Test-time Paradigm:** We introduce TF-TTCL, a training-free framework that enables frozen or black-box LLMs to self-improve online by distilling and reusing self-derived contrastive supervision, eliminating the need for gradient access or external knowledge guidance.
- **Contrastive Rule Distillation:** We introduce a mechanism that synthesizes "semantic gradients" from self-generated data. By employing multi-agent role-playing to augment query views and contrasting superior versus inferior trajectories, we distill explicit positive and negative rules that dynamically steer reasoning without modifying model weights.
- **Empirical Effectiveness:** Extensive experiments on closed-ended reasoning tasks and open-ended evaluation tasks demonstrate that TF-TTCL significantly outperforms both zero-shot baselines and existing test-time adaptation methods in online evaluation.

2 Related Work

2.1 Test-Time Adaptation

Test-Time Adaptation (TTA) originated in computer vision to address distribution shifts by updating model parameters online. Early works like Tent (Wang et al., 2021) minimize entropy to adapt batch normalization layers, while EATA (Niu et al., 2022) introduces weight regularization to mitigate catastrophic forgetting. More recently, COME (Zhang et al., 2025b) stabilizes this process by enforcing conservative confidence constraints.

Extending this paradigm to LLMs, gradient-based approaches optimize parameters on test streams: TTT-NN (Hardt and Sun, 2024) fine-tunes parameters on retrieved neighbors to approximate long-context memory, and TLM (Hu et al., 2025a) utilizes perplexity minimization to align models with an unseen domain. While Test-Time Reinforcement Learning (TTRL) (Zuo et al.,

2025) shows that LLMs can self-improve using consensus-based pseudo-rewards, it typically follows a multi-pass paradigm: the model first iterates over test instances to update its parameters and only then performs the final evaluation. This departs from realistic settings where requests arrive sequentially. In contrast, our method enforces a strictly online, single-pass protocol, requiring the model to answer each query immediately upon arrival, without any prior access to the test data.

2.2 Context Engineering

Context engineering (Mei et al., 2025) has progressed from simple prompting to sophisticated, memory-augmented systems. Initial efforts structure reasoning via Chain-of-Thought (CoT) (Wei et al., 2022) and Tree-of-Thought (ToT) (Yao et al., 2023a), while Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Gao et al., 2023) injects static external knowledge. The latest efforts shift toward self-evolving systems. Frameworks like ExpeL (Zhao et al., 2024) and AvaTaR (Wu et al., 2024) accumulate experiential trajectories to refine future reasoning, while gradient-free optimizers such as Training-Free GRPO (Cai et al., 2025) and LLM-based prompt optimizers (Tang et al., 2025) refine policies or instructions without backward propagation. Furthermore, ReasoningBank (Ouyang et al., 2025) introduces reasoning memories for scalable agent evolution.

Despite these advances, significant limitations persist. Standard CoT and ToT are stateless and cannot dynamically correct errors. Methods leveraging memory and iterative reflection, including ExpeL and AvaTaR, are primarily offline frameworks. ExpeL relies on external environmental rewards for reinforcement, and AvaTaR depends on ground-truth availability to extract insights. Neither can operate in our strict test-time setting. Similarly, Training-Free GRPO relies heavily on verifiable ground-truth rewards; without them, it degenerates into majority voting, limiting its applicability in domains lacking gold standards. While recent frameworks like ReasoningBank support online test-time scaling without ground-truth labels, they still necessitate deterministic external feedback (e.g., code execution results) combined with an LLM-as-Judge to partition trajectories. In scenarios lacking explicit external feedback, such systems default to a naive LLM-as-Judge, which suffers from severe self-confirmation bias. In contrast, TF-TTCL employs an **unsupervised, feedback-free protocol**.

Algorithm 1 The pipeline of TF-TTCL.

Input: Test stream $\mathcal{D}_{\text{test}}$, frozen LLM M_θ, instructions for agents $\mathcal{T}, \mathcal{A}, \mathcal{S}$. Repository $\mathcal{R} \leftarrow \varnothing$.
Output: Repository $\mathcal{R}$. Online response y_t.

for each query x_t in $\mathcal{D}_{\text{test}}$ **do**
  Retrieve rules $\mathbf{r}_{\text{ret}}$ from $\mathcal{R}$ via Eq. (8).
  Obtain anchor response $y_t^{\mathcal{T}} \leftarrow \mathcal{T}(x_t, \mathbf{r}_{\text{ret}})$
  Obtain response candidate set $\mathcal{Y}_t \leftarrow \{y_t^{\mathcal{T}}\}$.
  Obtain rewritten queries $\{x_t^{(n)}\}$ via Eq. (2).
  for $n = 1$ **to** N **do**
    Sample response $y_t^{(n)}$ via Eq. (3).
    $\mathcal{Y}_t \leftarrow \mathcal{Y}_t \cup \{y_t^{(n)}\}$
  end for
  Partition $\mathcal{Y}_t$ into positive and negative candidate sets $\mathcal{Y}^+$ and $\mathcal{Y}^-$, respectively.
  Obtain positive y_t^+ from $\mathcal{Y}^+$ via Eq. (4)
  Obtain negative y_t^- from $\mathcal{Y}^-$ via Eq. (6)
  $y_t \leftarrow y_t^+$
  Summarize new rules $\mathbf{r}_{\text{new}}$ via Eq. (7)
  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathbf{r}_{\text{new}}$
end for

By distilling explicit contrastive rules directly from self-generated outputs, we enable frozen LLMs to self-improve online at test time without relying on gradients, external environments, or ground-truths.

3 Problem Formulation

Without loss of generality, let $P(x)$ denote the training distribution and $Q(x)$ denote the test-time distribution. Let M_θ be a large language model (LLM) trained on data sampled from $P(x)$. Under standard training, the model parameters θ are optimized to perform well on in-distribution inputs $x \sim P(x)$. However, in practical deployments, the test-time inputs often exhibit distribution shifts, and many instances are drawn from $Q(x) \neq P(x)$. As a result, the model's predictions can become unreliable and the overall performance may degrade substantially. Test-time learning (TTL) aims to mitigate this degradation by improving the model's behavior using test-time signals. In this paper, we focus on *training-free test-time learning* for LLMs: the base model M_θ is **frozen** throughout the entire test-time process. The system interacts with an online stream $\mathcal{D}_{\text{test}} = \{(x_t, y_t^*)\}_{t=1}^T$, where $t \in \{1, \ldots, T\}$ indexes the time step and y_t^* denotes the (*inaccessible*) ground-truth target for x_t. At step t, the system observes the input $x_t \sim Q(x)$, generates an output y_t. To enable test-time im-

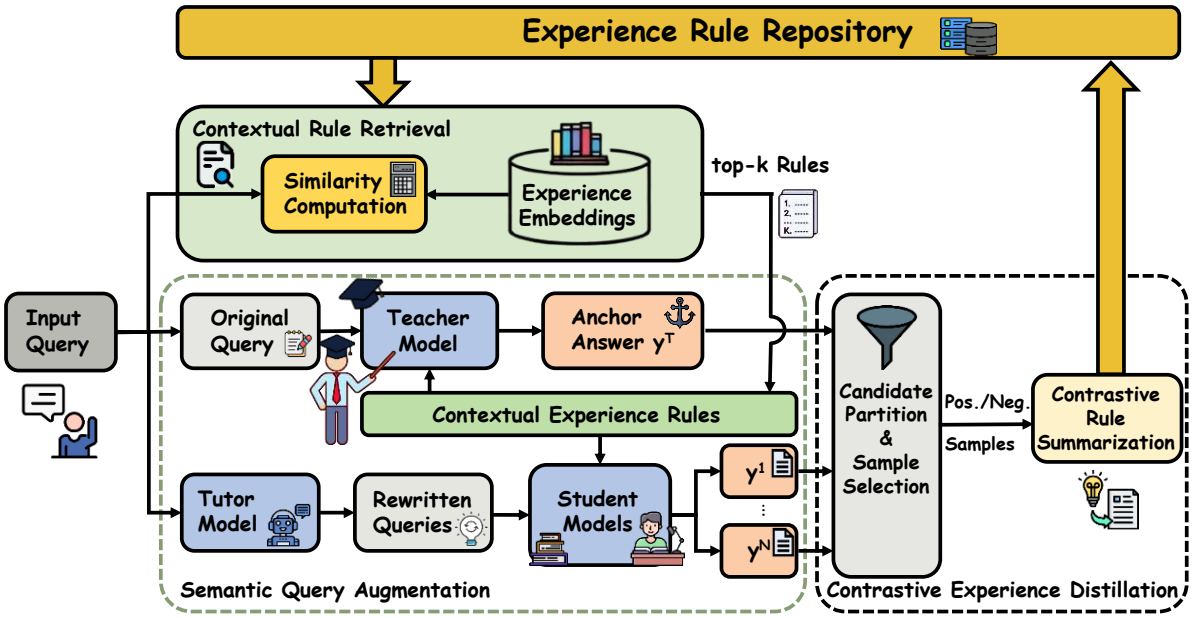

Figure 1: Overview of the TF-TTCL framework. 1) Semantic Query Augmentation: Employs multi-agent role-playing to probe diverse reasoning trajectories. 2) Contrastive Experience Distillation: Distills the semantic gap between selected positive and negative samples into textual rules for memory. 3) Contextual Rule Retrieval: Retrieves relevant historical insights from the rule repository to guide the inference.

provement under frozen parameters, we maintain an experience rule repository $\mathcal{R}_t$, initialized as $\mathcal{R}_0 \leftarrow \varnothing$, which accumulates transferable information distilled from past test-time interactions. Before generating at step t, the system retrieves a subset $\mathbf{r}_{\text{ret}} \subset \mathcal{R}_{t-1}$ and conditions the model on it, such that $y_t \sim M_\theta(y \mid x_t, \mathbf{r}_{\text{ret}})$. After producing y_t, the system extracts new transferable rules $\mathbf{r}_{\text{new}}$ from the current interaction and updates the repository via $\mathcal{R}_t \leftarrow \mathcal{R}_{t-1} \cup \mathbf{r}_{\text{new}}$. Our objective is to maximize the expected cumulative output quality over the test stream:

$$\max \sum_{t=1}^{T} \mathbb{E}_{y_t}\left[\mathcal{Q}(y_t, y_t^*)\right], \tag{1}$$

where $\mathcal{Q}(y_t, y_t^*)$ is a task-specific quality function measuring how well y_t aligns with y_t^*, and the expectation is taken with respect to the model's generation distribution.

4 Training-Free Contrastive Learning

In this paper, we propose Training-Free Test-Time Contrastive Learning (**TF-TTCL**), a training-free self-improvement framework for large language models. The overall pipeline is in Algorithm 1 and illustrated in Figure 1. Our design is inspired by contrastive learning (Chen et al., 2020; Schön, 1983): effective self-correction requires not only identifying a superior solution but also articulating why it outperforms inferior alternatives. Since the model parameters θ are frozen, we implement this contrastive learning loop through an evolving external repository and three coordinated modules.

First, the Semantic Query Augmentation module (§ 4.1) emulates test-time data augmentation: it employs a multi-agent role-playing strategy (TEACHER, TUTOR, STUDENT) to rewrite queries, compelling the model to generate diverse reasoning paths. Subsequently, the Contrastive Experience Distillation module (§ 4.2) captures the semantic gap between superior and inferior outputs. Instead of gradient updates, it distills these contrasts into explicit positive and negative rules which update the Experience Rule Repository. Finally, the Contextual Rule Retrieval module (§ 4.3) applies these rules to steer future inference, ensuring that experience rules learned from the past are dynamically transferred to new queries.

4.1 Semantic Query Augmentation

A key challenge in training-free test-time learning is to construct useful contrastive candidates without ground-truth labels: the model must explore diverse reasoning trajectories while avoiding degenerate variations caused by decoding randomness. To address this, we propose Semantic Query Augmentation (**SQA**), which generates multiple

semantically equivalent but stylistically different query variants and collects their corresponding responses. Concretely, SQA adopts a role-playing strategy with three agents: the **TEACHER** ($\mathcal{T}$), the **TUTOR** ($\mathcal{A}$), and the **STUDENT** ($\mathcal{S}$). All agents share the same LLM M_θ but use different system prompts and decoding configurations.

Anchor Output Generation. The TEACHER $\mathcal{T}$ prioritizes stable generation. Given original query x_t and retrieved rules $\mathbf{r}_{\text{ret}}$, it uses greedy decoding to produce a high-confidence response $y_t^{\mathcal{T}}$.

Query Augmentation. We design a query augmentation approach to explore the model's uncertainty under various linguistic expressions. Given the original query x_t, the TUTOR $\mathcal{A}$ rewrites it into N stylistically distinct variants to simulate input distribution shifts:

$$\{x_t^{(n)}\}_{n=1}^{N} = \mathcal{A}(x_t). \tag{2}$$

Response Sampling under Augmented Queries. For each semantically augmented query, the STUDENT $\mathcal{S}$ samples a response in parallel, conditioned on the same retrieved rules $\mathbf{r}_{\text{ret}}$, ensuring consistent knowledge across inputs:

$$y_t^{(n)} \sim \mathcal{S}\left(y \mid x_t^{(n)}, \mathbf{r}_{\text{ret}}\right), \quad \forall n \in \{1, \ldots, N\}. \tag{3}$$

Finally, we combine the TEACHER and STUDENT responses into a set of contrastive candidates $\mathcal{Y}_t$.

4.2 Contrastive Experience Distillation

While exploration exposes diverse reasoning paths, the raw candidate set is inherently noisy. Blindly utilizing these unlabeled candidates risks reinforcing the model's own hallucinations rather than correcting them. To this end, we propose Contrastive Experience Distillation (**CED**), a two-stage distillation mechanism that identifies reliable positives and informative (hard) negatives from the candidate set $\mathcal{Y}_t$ for subsequent rule distillation, as illustrated in Figure 2.

Consistency-Based Candidate Partitioning. To robustly partition the contrastive candidates $\mathcal{Y}_t$ into positive candidates ($\mathcal{Y}^+$) and negative candidates ($\mathcal{Y}^-$), we consider two evaluation regimes:

1) *Closed-ended Reasoning Task* (CRT): For tasks with a single ground-truth answer, we apply majority voting to partition $\mathcal{Y}_t$. If all agents produce different answers, we discard the sample and skip rule summarization to prevent propagating hallucinations. If all responses fall into a single cluster, we set $\mathcal{Y}^+ \leftarrow \mathcal{Y}_t$. Otherwise, we let the largest cluster define $\mathcal{Y}^+$ and assign the remaining clusters to $\mathcal{Y}^-$. In case of a tie, we set $\mathcal{Y}^+$ to the cluster containing the lowest-perplexity candidate.

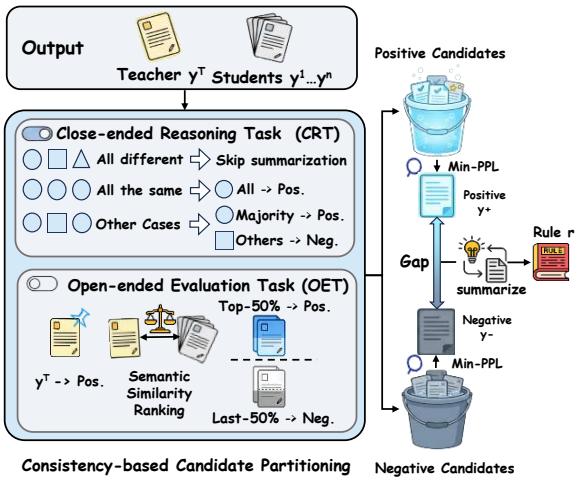

Figure 2: The pipeline of Contrastive Experience Distillation. Our approach partitions outputs into Positive and Negative Candidates using consistency-based candidate partitioning. We select the positive and negative candidates via min-PPL selection. The final adaptation rules are generated by summarizing the reasoning gap.

2) *Open-ended Evaluation Task* (OET): For tasks that admit multiple plausible answers, we use the TEACHER 's response $y_t^{\mathcal{T}}$ as a semantic reference. Then we compute embedding-based similarity between each candidate and $y_t^{\mathcal{T}}$. We define $\mathcal{Y}^+$ as the top 50% most similar candidates, and assign the remaining divergent responses to $\mathcal{Y}^-$.

Uncertainty-Aware Sample Selection. We adopt sequence-level generation perplexity (PPL) as a proxy for the model's confidence (Hu et al., 2025a). From $\mathcal{Y}^+$, we select the positive sample y_t^+ with the lowest perplexity, identifying the candidate that best aligns with the model's distribution:

$$y_t^+ = \arg\min_{y \in \mathcal{Y}^+} \mathcal{P}(y). \tag{4}$$

We compute the sequence-level perplexity $\mathcal{P}(y)$ as:

$$\mathcal{P}(y) = \exp\left(\frac{1}{L}\sum_{i=1}^{L} -\log M_\theta(y_{[i]} \mid x, y_{[1:i-1]})\right), \tag{5}$$

where $y_{[i]}$ denotes the i-th token of response y, L is the sequence length, x is the input query, and M_θ is the LLM probability distribution. Crucially, for $\mathcal{Y}^-$, we also select the candidate with the minimum perplexity to identify negative y_t^-. This choice is motivated by findings that LLMs often produce

Example

Query: Henry and 3 of his friends order 7 pizzas for lunch. Each pizza is cut into 8 slices. If Henry and his friends want to share the pizzas equally, how many slices can each of them have?

Useful Positive Rule (r^+):
Divide the total quantity by the number of recipients to solve problems involving equal distribution. Ensure the total recipient count includes all individuals mentioned (e.g., Henry plus his 3 friends equals 4 people).

Useful Negative Rule (r^-):
Avoid dividing the total by only the number of "friends" (3) while neglecting the subject. This miscounts the total number of equal shares and leads to an overestimation of the individual portion.

Figure 3: A representative example demonstrating the extraction of useful contrastive rules (r^+, r^-) from reasoning gaps, which serve as explicit guidance for subsequent problem-solving steps.

confident hallucinations (Zhang et al., 2023). By selecting the minimum-perplexity (min-PPL) candidate from $\mathcal{Y}^-$, we target errors that the model is most confident about, providing the strongest signal for rectifying the decision boundary (Robinson et al., 2021):

$$y_t^- = \arg\min_{y \in \mathcal{Y}^-} \mathcal{P}(y). \tag{6}$$

Contrastive Rule Summarization. We employ the summarizer (the same LLM with a different system prompt) to distill the reasoning gap between the selected positive response y_t^+ and the hard negative y_t^- into corrective guidelines. To provide comprehensive guidance, we explicitly generate two distinct types of rules: a *positive rule* r_t^+ (what to do) and a *negative rule* r_t^- (what to avoid):

$$\{r_t^+, r_t^-\} = \text{Summary}\left(x_t,\ y_t^+,\ y_t^-\right). \tag{7}$$

These new rules $\mathbf{r}_{\text{new}} = \{r_t^+, r_t^-\}$ are then appended to the repository $\mathcal{R}$. To provide a concrete intuition of these distilled rules, Figure 3 illustrates a representative rule pair derived from a math problem. See Appendix D for more cases.

4.3 Contextual Rule Retrieval

To close the self-improvement loop, we propose Contextual Rule Retrieval (**CRR**), which maintains a long-term memory $\mathcal{R}$ that continuously stores reusable rules distilled by the Contrastive Experience Distillation. Unlike static RAG, $\mathcal{R}$ is updated online and queried at inference time.

Organize Positive and Negative Rule Sets. A key challenge is to distinguish positive signals from negative ones. To avoid confusion, we maintain two disjoint memory sets: a positive-rule set $\mathcal{R}_{\text{pos}}$ containing r^+, and a negative-rule set $\mathcal{R}_{\text{neg}}$ containing r^-. Each memory entry is stored as a key–value pair $(\mathbf{e}, r)$, where the value is a rule $r \in \mathcal{R} = \mathcal{R}_{\text{pos}} \cup \mathcal{R}_{\text{neg}}$, and the key $\mathbf{e} = \text{Embed}(r)$ is the embedding of r.

Rules Retrieval. Given a new query x_t, we compute $\mathbf{e}_t = \text{Embed}(x_t)$ and retrieve Top-K positive and Top-K negative rules from $\mathcal{R}_{\text{pos}}$ and $\mathcal{R}_{\text{neg}}$ using cosine similarity:

$$\begin{aligned} \mathbf{r}_{\text{ret}}^+ &= \text{Top-K}_{\,r \in \mathcal{R}_{\text{pos}}}\left[\cos(\mathbf{e}_t, \mathbf{e}_r)\right], \\ \mathbf{r}_{\text{ret}}^- &= \text{Top-K}_{\,r \in \mathcal{R}_{\text{neg}}}\left[\cos(\mathbf{e}_t, \mathbf{e}_r)\right], \end{aligned} \tag{8}$$

where $\mathbf{e}_r$ is the stored embedding associated with rule r. The retrieved context as $\mathbf{r}_{\text{ret}} = \mathbf{r}_{\text{ret}}^+ \cup \mathbf{r}_{\text{ret}}^-$.

Integrate Retrieved Rules into Structured Context. We use a structured prompt template to clearly demarcate the retrieved knowledge. The final context $\mathbf{r}_{\text{ret}}$ is formed by concatenating the positive and negative sets with explicit instruction headers. Labeling $\mathbf{r}_{\text{ret}}^-$ imposes a negative constraint, pruning known error paths. Labeling $\mathbf{r}_{\text{ret}}^+$ guides the model toward proven solutions. This structured injection maximizes the utility of retrieved knowledge without any parameter updates.

5 Experiments

5.1 Experimental Settings

Datasets. We evaluate the model's reasoning ability on the test sets of a series of benchmarks representing Closed-ended Reasoning Task, including GSM8k, MATH-500, AIME24, and Minerva, covering difficulty levels from grade-school arithmetic to competition-level problems. We use DomainBench (Hu et al., 2025a), which spans four specialized domains, including Geography, Agriculture, Medicine, and Finance, to assess adaptation under distribution shifts in Open-ended Evaluation Task. See Appendix B.1 for details.

Metrics. Following (Hu et al., 2025a), we report ROUGE-Lsum (R-Lsum) (Lin, 2004) on DomainBench to quantify generation quality. For mathematical benchmarks, we evaluate via accuracy based on Exact Match (Chang et al., 2024). See Appendix B.3 for details.

Baselines and Models. We evaluate our method across models of varying scales and access regimes.

Table 2: Comparison with SOTA methods on **Closed-ended Reasoning Task** using Llama-3.1-8B-Instruct. The metric is Accuracy (%). "Base LLM" denotes the zero-shot baseline. The best results are **bolded**.

Method	Publication	GSM8k	MATH-500	AIME24	Minerva	Average
Base LLM	-	82.49	49.20	3.33	20.96	39.00
Tent (Wang et al., 2021)	ICLR 2021	70.20	49.20	10.00	21.32	37.68
EATA (Niu et al., 2022)	ICML 2022	75.06	49.40	6.67	20.96	38.02
COME (Zhang et al., 2025b)	ICLR 2025	75.59	48.80	6.67	20.96	38.01
TLM (Hu et al., 2025a)	ICML 2025	85.06	50.00	6.67	19.49	40.31
TF-GRPO (Cai et al., 2025)	arXiv 2025	86.49	53.00	3.33	21.69	41.13
TF-TTCL (Ours)	-	**87.49**	**54.00**	**13.33**	**24.63**	**44.86**

Table 3: Comparison with SOTA methods on **Open-ended Evaluation Task** using Llama-3.1-8B-Instruct. The metric is ROUGE-Lsum (higher is better). "Base LLM" denotes the zero-shot baseline. The best results are **bolded**.

Method	Publication	Geography	Agriculture	Medicine	Finance	Average
Base LLM	-	0.2441	0.0876	0.1356	0.2251	0.1731
Tent (Wang et al., 2021)	ICLR 2021	0.2682	0.0624	0.1448	0.2140	0.1724
EATA (Niu et al., 2022)	ICML 2022	0.2757	0.0626	0.1455	0.1886	0.1681
COME (Zhang et al., 2025b)	ICLR 2025	0.2636	0.0407	0.1382	0.0699	0.1281
TLM (Hu et al., 2025a)	ICML 2025	0.2620	0.0956	0.1372	0.2295	0.1811
TF-GRPO (Cai et al., 2025)	arXiv 2025	0.2260	0.0993	0.1147	0.2071	0.1618
TF-TTCL (Ours)	-	**0.2798**	**0.1095**	**0.2018**	**0.2863**	**0.2194**

For open-weight models (Tables 2 and 3), we employ Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as the primary backbone. The unadapted base model, denoted as Base LLM, serves as the zero-shot baseline. We compare our approach against several gradient-based test-time adaptation (TTA) methods, including Tent (Wang et al., 2021), EATA (Niu et al., 2022), COME (Zhang et al., 2025b), and TLM (Hu et al., 2025a), as well as TF-GRPO (Cai et al., 2025). Given that TF-GRPO relies on ground-truth feedback, we implement majority voting to synthesize reward signals during the adaptation process. For API-based evaluations involving black-box models (Table 4), we utilize Qwen-Plus (Yang et al., 2025a) and DeepSeek-V3.2 (Liu et al., 2025). Since gradient-based optimization is infeasible in this setting, we restrict our comparison to gradient-free baselines, specifically Chain-of-Thought (CoT) prompting (Wei et al., 2022) and TF-GRPO (Cai et al., 2025).

Implementation Details. For TF-TTCL, the rule repository $\mathcal{R}$ starts empty and is populated online. We employ Qwen-3-0.6B-Embedding (Yang et al., 2025a) to encode both input queries and distilled rules into dense vector representations. At inference, we retrieve the Top-30 positive and Top-30 negative rules based on cosine similarity with the query embedding, and we use 4 STUDENT sample instances for diversity. See Appendix C.1 for hyperparameter analysis. If rules exceed the context window, we keep the highest-scoring rules in descending order up to the context limit and drop the rest. All methods use identical generation hyperparameters. The TEACHER model uses greedy decoding (temperature 0.0) for stable anchors, while TUTOR and STUDENT employ sampling with temperature 0.7 and top-$p = 0.9$ to promote diverse reasoning paths. For details, see Appendix B.2.

5.2 Comparison Experiments

Performance on Closed-ended Reasoning Task. Table 2 shows that our method TF-TTCL consistently outperforms existing TTA approaches across all math benchmarks. Notably, it achieves the highest accuracy on GSM8k (87.49%), MATH-500 (54.00%), AIME24 (13.33%), and Minerva (24.63%), leading to an average of 44.86%. These results demonstrate that TF-TTCL effectively leverages test-time signals to improve reasoning performance, especially on more challenging tasks, without requiring additional training. By explicitly comparing valid against invalid reasoning traces, our mechanism acts as a logical verifier, ensuring that intermediate steps remain coherent and effectively blocking the error propagation typical in long-chain derivations.

Performance on Open-ended Evaluation Task. Table 3 reports the results on the open-ended DomainBench dataset. TF-TTCL consistently achieves the best performance across all four do-

Table 4: Performance comparison on API-based Models. We compare our training-free approach against standard Chain-of-Thought (Wei et al., 2022) and TF-GRPO (Cai et al., 2025) on AIME24 (Reasoning) and Finance (Domain). The best results are **bolded**.

Method	Qwen-Plus		DeepSeek-V3.2	
	AIME24	Finance	AIME24	Finance
Base LLM	30.00	0.2647	66.67	0.2578
Chain-of-Thought	40.00	0.2297	70.00	0.2428
TF-GRPO	70.00	0.2500	80.00	0.2580
TF-TTCL (Ours)	**76.67**	**0.2831**	**83.33**	**0.2919**

mains, raising the average ROUGE-Lsum from 0.1731 (Base LLM) to 0.2194. This validates that our contrastive rule mechanism successfully extracts transferable knowledge even in unstructured generation tasks. In contrast, the reinforcement learning-based method TF-GRPO fails to improve over the zero-shot baseline (0.1731 $\rightarrow$ 0.1618). We attribute this performance degradation to the inherent challenge of open-ended evaluation: unlike mathematical reasoning where outcomes are binary, open-ended generation lacks deterministic ground truth. Consequently, TF-GRPO struggles to derive meaningful reward signals from the generated text, leading to ineffective policy optimization.

Generalization on Black-box Models. Table 4 assesses the performance of API-accessible models under realistic deployment constraints. Compared with TF-GRPO, TF-TTCL learns exclusively from self-generated contrastive data, demonstrating that contrastive experience can effectively substitute for explicit reward supervision. Notably, on DeepSeek-V3.2 (Liu et al., 2025), TF-TTCL outperforms all methods (see Appendix D.1 for detailed case studies). Furthermore, on Qwen-Plus (Yang et al., 2025a), while TF-GRPO improves reasoning on AIME24, it suffers from overfitting that degrades domain adaptation on Finance (see Appendix D.2 for detailed case studies). In contrast, TF-TTCL enhances performance on both CRT and OET, suggesting that its contrastive memory provides a more robust and balanced adaptation signal.

5.3 Ablation Studies

We conduct ablation studies on the GSM8k and Finance datasets based on Llama-3.1-8B-Instruct.

Impact of Core Modules. As shown in Table 5, we provide a concise analysis of the module contributions. Contrastive Experience Distillation (CED) emerges as the most critical component; removing

Table 5: Full ablation study of TF-TTCL on GSM8k and Finance. The best results are **bolded**.

Method	GSM8k	Finance
Baseline	82.49	0.2251
TF-TTCL w/o SQA	87.11	0.2851
TF-TTCL w/o CED	85.97	0.2639
TF-TTCL w/o CRR	87.34	0.2596
TF-TTCL (Ours)	**87.49**	**0.2863**

Table 6: Ablation study on key components of TF-TTCL. For CED, we remove either the positive-rule set or the negative-rule set to examine their individual effects. For CRR, we replace curated rules with randomly sampled rules. The best results are **bolded**.

Component	Variant	GSM8k	Finance
CED	w/o positive rules	87.19	0.2812
	w/o negative rules	86.88	0.2668
	Ours	**87.49**	**0.2863**
CRR	w/ random rules	87.41	0.2665
	Ours	**87.49**	**0.2863**

it causes the most significant performance degradation across both benchmarks (e.g., 87.49% $\rightarrow$ 85.97% on GSM8k), confirming that high-quality rule synthesis is the foundation of our framework. The impact of Contextual Rule Retrieval (CRR) exhibits distinct task-dependent behaviors. In open-ended tasks like Finance, removing retrieval and using all rules truncated by context window leads to a sharp decline (0.2863 $\rightarrow$ 0.2596), indicating that precise, context-aware guidance is essential for navigating unstructured output spaces. Conversely, performance on GSM8k remains robust without CRR, suggesting that logical rules for mathematical reasoning possess high universality. Finally, Semantic Query Augmentation (SQA) modestly aids contrastive learning by adding candidate diversity. For details, see Appendix C.4.

Asymmetry of Positive and Negative Rules. A fine-grained analysis in Table 6 reveals that negative rules contribute more significantly than positive ones. For instance, removing negative rules causes a sharper performance drop on GSM8k (87.49% $\rightarrow$ 86.88%) compared to removing positive rules (87.49% $\rightarrow$ 87.19%). This asymmetry suggests that positive rules often merely reinforce knowledge the model already possesses, whereas negative rules provide unique, corrective "interdiction signals" that effectively prevent the model from repeating specific, high-probability errors.

Retrieval Strategy Effectiveness. Table 6 val-

Table 7: System efficiency and scalability on GSM8k. Parallel execution caps latency, while memory pruning bounds repository growth and improves performance.

Variants	Latency	Memory	Acc.
Single Call	2.05s	-	82.49
TF-TTCL (Sequence)	10.25s	Unbounded	87.49
TF-TTCL (Parallel)	**4.11s**	Unbounded	87.49
+ Pruning Strategy	**4.11s**	**1,000**	**87.72**

idates the necessity of precise retrieval. Random selection achieves comparable results on GSM8k (87.41%) but performance drops sharply on Finance (0.2665), lagging behind our method by 0.0198. This contrast suggests that math tasks are robust to generic rules due to the universality of logical principles, while open-ended generation is highly sensitive to rule alignment, requiring tightly relevant signals to navigate output.

Computational Overhead and Memory Pruning. A common concern with multi-agent reflection is the inference latency and unbounded memory growth. To address these deployment bottlenecks, we introduce two system-level optimizations. **To minimize latency**, we execute the TUTOR and STUDENT agents in parallel and decouple the rule summarization step (0.39s) as an asynchronous background process. As detailed in Table 7, this parallelization caps user-perceived latency (time to return y_t) at merely 2.01× that of a single LLM call (4.11s vs. 2.05s). Crucially, this asynchronous memory update completes well before the next query x_{t+1} arrives, perfectly maintaining our online, single-pass evaluation protocol. **To combat linear rule accumulation**, we implement a similarity-based FIFO pruning strategy to maintain a fixed-capacity repository. Empirical validation on GSM8k demonstrates that bounding the memory (e.g., to 1,000 rules) not only caps retrieval overhead but also serves as a regularization mechanism that filters out redundant rules, slightly improving final accuracy (87.49% → 87.72%). Together, these designs ensure TF-TTCL is efficient and scalable for continuous online deployment.

6 Conclusion

In this paper, we present Training-Free Test-time Contrastive Learning (TF-TTCL), a framework that enables frozen LLMs to adapt continuously during online evaluation without gradient updates and external knowledge. Our approach introduces three synergistic components: Semantic Query Augmentation constructs diverse reasoning paths through multi-agent role-playing, Contrastive Experience Distillation filters and distills the semantic gap between superior and inferior outputs into explicit rules, and Contextual Rule Retrieval dynamically injects these rules for future generations. Experiments on closed-ended reasoning tasks and open-ended evaluation tasks demonstrate that TF-TTCL outperforms both zero-shot baselines and existing test-time adaptation methods in online evaluation.

Limitations

First, our framework is subject to diminishing returns in exploration. While a stronger TUTOR model facilitates broader reasoning coverage, the marginal performance gains decline as the model approaches its capability ceiling (i.e., saturation). Second, while our similarity-based pruning effectively resolves memory compression issues, the current framework relies on a one-shot injection of all retrieved rules. Recently, progressive disclosure strategies like Agent Skills have gained significant traction for handling complex prompts more efficiently. Future work will explore applying progressive disclosure within our framework to step-wise and dynamically inject rules, thereby further optimizing the model's contextual utilization during long-horizon reasoning.

Ethical Considerations

The flexibility of TF-TTCL in handling input configurations may increase vulnerability to adversarial prompt injections. Therefore, we recommend combining our framework with robust input validation and the base model's native safety filters to prevent harmful content in practice.

Acknowledgments

This work is funded by Guangdong Basic and Applied Basic Research Foundation (2024A1515010900).

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. *CoRR*, abs/2303.08774.

Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi

Li, Haojia Lin, Yong Mao, Ke Li, and Xing Sun. 2025. Training-free group relative policy optimization. *CoRR*, abs/2510.08191.

Edoardo Cetin, Tianyu Zhao, and Yujin Tang. 2025. Reinforcement learning teachers of test time scaling. *CoRR*, abs/2506.08388.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3):39:1–39:45.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, Proceedings of Machine Learning Research, pages 1597–1607. PMLR.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Nature*, 645(8081):633–638.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1735–1742. IEEE Computer Society.

Moritz Hardt and Yu Sun. 2024. Test-time training on nearest neighbors for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. Computer Vision Foundation / IEEE.

Haigen Hu, Xiaoyuan Wang, Yan Zhang, Qi Chen, and Qiu Guan. 2024. A comprehensive survey on contrastive learning. *Neurocomputing*, 610:128645.

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. 2025a. Test-time learning for large language models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. 2025b. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32779–32798, Vienna, Austria. Association for Computational Linguistics.

Yang Hu, Xingyu Zhang, Xueji Fang, Zhiyang Chen, Xiao Wang, Huatian Zhang, and Guojun Qi. 2025c. SLOT: sample-specific language model optimization at test-time. *CoRR*, abs/2505.12392.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1051–1068. Association for Computational Linguistics.

Tenghao Huang, Kinjal Basu, Ibrahim Abdelaziz, Pavan Kapanipathi, Jonathan May, and Muhao Chen. 2025. R2D2: Remembering, replaying and dynamic decision making with a reflective agentic memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30318–30330, Vienna, Austria. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Xinyue Kang, Diwei Shi, and Li Chen. 2026. Model whisper: Steering vectors unlock large language models' potential in test-time. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 31392–31400. AAAI Press.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Meng-Chieh Lee, Qi Zhu, Costas Mavromatis, Zhen Han, Soji Adeshina, Vassilis N. Ioannidis, Huzefa Rangwala, and Christos Faloutsos. 2025. HybGRAG: Hybrid retrieval-augmented generation on textual and relational knowledge bases. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 879–893, Vienna, Austria. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Mot: Memory-of-thought enables chatgpt to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6354–6374. Association for Computational Linguistics.

Jian Liang, Ran He, and Tieniu Tan. 2025. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *CoRR*, abs/2512.02556.

Jing Ma, Hanlin Li, and Xiang Xiang. 2025. PTTA: purifying malicious samples for test-time model adaptation. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. 2025. A survey of context engineering for large language models. *CoRR*, abs/2507.13334.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. 2022. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pages 16888–16905. PMLR.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. 2025. Reasoningbank: Scaling agent self-evolving with reasoning memory. *CoRR*, abs/2509.25140.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. 2025. Infonce: Identifying the gap between theory and practice. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2025, Mai Khao, Thailand, 3-5 May 2025*, Proceedings of Machine Learning Research, pages 4159–4167. PMLR.

D.A. Schön. 1983. *Reflective Practitioner*. Basic Books.

Akshit Singh, Shyam Marjit, Wei Lin, Paul Gavrikov, Serena Yeung-Levy, Hilde Kuehne, Rogério Feris, Sivan Doveh, James R. Glass, and Muhammad Jehanzeb Mirza. 2025. TTRV: test-time reinforcement learning for vision language models. *CoRR*, abs/2510.06783.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Jiejun Tan, Zhicheng Dou, Yutao Zhu, Peidong Guo, Kun Fang, and Ji-Rong Wen. 2024. Small models, big insights: Leveraging slim proxy models to decide when and what to retrieve for LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4420–4436, Bangkok, Thailand. Association for Computational Linguistics.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8416–8439, Vienna, Austria. Association for Computational Linguistics.

Xinyu Tang, Xiaolei Wang, Wayne Xin Zhao, Siyuan Lu, Yaliang Li, and Ji-Rong Wen. 2025. Unleashing the potential of large language models as prompt optimizers: Analogical analysis with gradient-based model optimizers. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 25264–25272. AAAI Press.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. 2021. Tent: Fully test-time adaptation by entropy minimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan O Arik. 2025a. Astute RAG: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Yiping Wang, Shao-Rong Su, Zhiyuan Zeng, Eva Xu, Liliang Ren, Xinyu Yang, Zeyi Huang, Xuehai He, Luyao Ma, Baolin Peng, Hao Cheng, Pengcheng He, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. 2025b. Thetaevolve: Test-time learning on open problems. *CoRR*, abs/2511.23473.

Zheng Wang, Shu Teo, Jieer Ouyang, Yongjun Xu, and Wei Shi. 2024. M-RAG: Reinforcing large language model performance through retrieval-augmented generation with multiple partitions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1966–1978, Bangkok, Thailand. Association for Computational Linguistics.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2025c. Agent workflow memory. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Shirley Wu, Shiyu Zhao, Qian Huang, Kexin Huang, Michihiro Yasunaga, Kaidi Cao, Vassilis N. Ioannidis, Karthik Subbian, Jure Leskovec, and James Zou. 2024. AvaTaR: Optimizing LLM agents for tool usage via contrastive reasoning. In *Advances in Neural Information Processing Systems*, volume 37, pages 25981–26010. Curran Associates, Inc.

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

Zairun Yang, Yilin Wang, Zhengyan Shi, Yuan Yao, Lei Liang, Keyan Ding, Emine Yilmaz, Huajun Chen, and Qiang Zhang. 2025b. EventRAG: Enhancing LLM generation with event knowledge graphs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Zhen Yang, Mingyang Zhang, Feng Chen, Ganggui Ding, Liang Hou, Xin Tao, Pengfei Wan, and Ying-Cong Chen. 2025c. Less is more: Improving LLM reasoning with minimal test-time intervention. *CoRR*, abs/2510.13940.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural*

Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. 2025a. Verbalized sampling: How to mitigate mode collapse and unlock LLM diversity. *CoRR*, abs/2510.01171.

Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. 2025b. COME: test-time adaption by conservatively minimizing entropy. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. 2025c. What, how, where, and how well? A survey on test-time scaling in large language models. *CoRR*, abs/2503.24235.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the AI ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642.

Yujia Zhou, Zheng Liu, and Zhicheng Dou. 2024. Boosting the potential of large language models with an intelligent information assistant. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, and 1 others. 2025. Ttrl: Test-time reinforcement learning. In *Proceedings of the Neural Information Processing Systems*.

Appendix

This appendix provides additional related work, detailed experimental settings, results from extended experiments, as well as implementation and prompt details. The appendix is organized as follows:

- Appendix A – More Related Work
- Appendix B – Experiment Setup
- Appendix C – Extended Experiments
- Appendix D – Case Studies
- Appendix E – Prompt Details

A More Related Work

A.1 Test-time Paradigms

Test-time adaptation (TTA). The primary goal of TTA is to mitigate distribution shifts by adjusting a pre-trained model to unlabeled data on the fly (Liang et al., 2025). Originating from computer vision, traditional TTA methods typically employ self-supervised objectives, such as entropy minimization or pseudo-labeling, to update batch normalization statistics. In the era of LLMs, research has evolved to address the discrete nature of text and complex reasoning requirements.

On one hand, optimization-based approaches conduct sample-specific updates using temporary parameter vectors to align models with complex instructions (Hu et al., 2025c). On the other hand, non-parametric (inference-only) methods improve robustness without permanent weight updates. The PTTA approach purifies potentially malicious test samples to stabilize adaptation (Ma et al., 2025), whereas the TTSV approach reduces output entropy via test-time steering vectors that steer activations (Kang et al., 2026). Singh et al. (Singh et al., 2025) extend these ideas to vision–language reasoning through the TTRV approach, utilizing test-time reinforcement learning and frequency-based rewards. These methods effectively align models to new domains or reduce statistical uncertainty, primarily through implicit signals such as entropy or gradient updates (Liang et al., 2025). However, TF-TTCL leverages *semantic* contrastive signals among generated candidates to refine the model's internal representations via an evolving external memory.

Test-time compute. Also referred to as test-time scaling, this paradigm posits that increasing inference-time computation can elicit "System 2" thinking behaviors, thereby enhancing reasoning capabilities without pre-training scaling (Zhang et al., 2025c). A central theme in this domain is the efficient management of the compute budget (Zhang et al., 2025c). Muennighoff et al. (Muennighoff et al., 2025) demonstrate a linear scaling law between performance and inference time through "budget forcing," a technique that compels models to generate "wait" tokens to extend their internal thought process. To improve the efficiency, Yang et al. (Yang et al., 2025c) propose Minimal Test-Time Intervention, which strategically applies classifier-free guidance only to tokens exhibiting high local uncertainty.

Beyond fixed strategies, recent works integrate learning mechanisms into the inference phase. ThetaEvolve, introduced by Wang et al. (Wang et al., 2025b), is a framework for test-time learning on open problems that combines evolutionary search with optional test-time reinforcement learning to optimize reasoning trajectories. Similarly, Cetin et al. (Cetin et al., 2025) explore Reinforcement-Learned Teachers that produce "connect-the-dots" explanations to guide downstream distillation. These paradigms enhance performance by scaling search depth or optimizing generation paths, typically treating the model as a generator to be guided or filtered (Zhang et al., 2025c). In contrast, TF-TTCL maintains frozen parameters while emulating synaptic plasticity: it proactively explores a local hypothesis space and summarizes the logic gap between positive and negative trajectories into explicit textual rules, enabling the model to learn from errors.

A.2 Contrastive Learning Paradigms

Contrastive Learning in Computer Vision. The roots of contrastive learning (CL) can be traced back to dimensionality reduction techniques that sought to learn invariant mappings based on neighborhood relationships (Hadsell et al., 2006). In the modern deep learning era, CL revolutionized unsupervised visual representation learning by treating data augmentation as a source of supervision. Seminal frameworks, such as Momentum Contrast (MoCo) (He et al., 2020), introduced dynamic dictionaries to maintain consistent negative samples, significantly closing the gap between unsupervised and supervised performance. Other studies have focused on the theoretical underpinnings of view selection, arguing that optimal views should minimize mutual information while preserving task-relevant features (Tian et al., 2020; Hu et al., 2024).

While early methods relied on self-supervised instance discrimination, subsequent works extended these principles to the supervised setting. Supervised Contrastive Learning (SupCon) (Khosla et al., 2020) leverages label information to form positive clusters, demonstrating superior robustness compared to traditional cross-entropy losses. Furthermore, recent analyses of the InfoNCE loss have highlighted the importance of addressing anisotropic latent spaces in practical deployments (Rusak et al., 2025). These vision-based foundations established the core mechanism of minimizing distance between positive pairs, a concept our method adapts by treating "successful reasoning paths" as positive anchors.

Contrastive Paradigms in NLP. Contrastive objectives have been adopted primarily to improve language representations during training or fine-tuning in NLP, especially for sentence embedding learning. SimCSE (Gao et al., 2021) treats standard dropout as a minimal augmentation and contrasts two stochastic forward passes of the same sentence, effectively predicting the sentence itself under a contrastive objective. Beyond representation learning, contrastive mechanisms have also been explored at inference time to steer text generation. Contrastive Decoding (CD) (Li et al., 2023) formulates generation as maximizing the difference between the log-likelihoods of an expert model and an amateur model. Operationally, it subtracts the amateur model's logits from the expert's, which penalizes common failure modes like repetition and hallucination without additional training.

While effective, existing methods typically operate either at the parameter level or the logit level. In contrast, our approach works at the context level: it neither updates model weights nor alters decoding probabilities. Instead, it embeds retrieved examples of both successful and failed reasoning directly into the prompt, providing semantic anchors that help the model identify and follow correct reasoning pattern without any training.

A.3 Advanced In-Context Mechanisms

Advanced Retrieval-Augmented Generation. Recent advancements extend RAG beyond static knowledge retrieval toward agentic interactions. SlimPLM (Tan et al., 2024) employs a lightweight proxy to dynamically filter unnecessary retrieval steps, and HybGRAG (Lee et al., 2025) handles hybrid queries by fusing textual and relational data structures. To overcome the rigidity of fixed retrieval, DRAGIN (Su et al., 2024) dynamically determines *when* and *what* to retrieve based on the model's real-time information needs. Complex reasoning scenarios have motivated the use of structured representations: EventRAG (Yang et al., 2025b) leverages event knowledge graphs to capture temporal and logical dependencies, while M-RAG (Wang et al., 2024) partitions memory databases to sharpen retrieval focus. In order to address unreliable retrieved context, Wang et al. (Wang et al., 2025a) propose Astute RAG, which reconciles conflicts between the model's internal parametric knowledge and potentially imperfect external sources. Taking a step further toward autonomous systems, AssistRAG (Zhou et al., 2024) embeds intelligent assistants within LLMs to orchestrate tool usage and memory construction. TF-TTCL aligns with this trend of dynamic adaptation and focuses on retrieving behavioral references to adapt the model's policy online.

Memory Management and Context Optimization. Deploying LLMs in long-horizon or streaming settings necessitates efficient memory mechanisms. A foundational insight emerges from Memory-of-Thought (Li and Qiu, 2023): high-confidence past reasoning can serve as external memory, enabling self-improvement without parameter updates. Building on this, hierarchical architectures have gained traction. Agent Workflow Memory (Wang et al., 2025c) stores reusable subgoals, while HiAgent (Hu et al., 2025b) organizes action trajectories at multiple abstraction levels. Complementing these structural innovations, reflective mechanisms play an important role: R2D2 (Huang et al., 2025) reconstructs environmental "maps" through replay buffers, and Reflective Memory Management (Tan et al., 2025) iteratively refines retrieval strategies via retrospective analysis. From an efficiency standpoint, prompt compression techniques such as LongLLMLingua (Jiang et al., 2024) mitigate position bias while substantially reducing computational overhead. Our approach complements these advances by treating memory not as a passive buffer, but as a dynamic pool of contrastive examples that updates as the model processes the test stream.

B More Experimental Details

B.1 Datasets Details

To evaluate the adaptability and reasoning capabilities of TF-TTCL, we use eight datasets. These are

Table 8: Description of the eight evaluation datasets employed in our experiments, grouped into domain-specific question answering (DomainBench) and mathematical reasoning benchmarks (Math Benchmarks).

Dataset	Task / Description
DomainBench	
Geography	Knowledge-intensive Q&A
Agriculture	Agricultural production Q&A
Medicine	Patient-Doctor Dialogue
Finance	Sentiment Analysis & Financial Q&A
Math Benchmarks (Test Sets)	
GSM8k	Grade school math problems
MATH-500	Competition-level problems
AIME24	2024 Invitational Math Exam
MinervaMath	Quantitative reasoning

categorized into domain-specific benchmarks (DomainBench) and mathematical reasoning benchmarks (Math Benchmarks). Table 8 provides a summary of the statistics for these datasets.

For vertical domain evaluation, we adopt the **DomainBench** suite (Hu et al., 2025a). While the original benchmarks vary in size, we standardize our evaluation by randomly sampling 5,000 instances from each of the four domains to ensure a balanced comparison. This suite assesses the model's proficiency in handling specialized knowledge and terminology across professional fields.

Geography. This evaluation set is derived from the GeoSignal dataset. This corpus is specifically curated for Earth Sciences using a hybrid pipeline of human expert curation and semi-automated construction. The samples cover a wide array of tasks, including Named Entity Recognition (NER), fact verification, and complex question answering, requiring the model to process specialized terms and reason over Earth Science concepts.

Agriculture. We utilize the Agriculture-QA dataset to test the model's utility in the agricultural sector. This dataset aggregates knowledge related to the entire agricultural production cycle. The questions span diverse topics ranging from crop cultivation techniques and soil management to livestock farming practices. By utilizing this dataset, we evaluate the model's ability to comprehend and generate accurate responses within a highly specific industry context.

Medicine. The medical domain is evaluated using the GenMedGPT-5k dataset. This dataset is distinct in its construction, utilizing ChatGPT to synthesize realistic, multi-turn dialogues between patients and doctors. It serves as a simulation of real-world clinical scenarios, featuring a rich variety of patient inquiries and professional diagnostic responses. Our evaluation focuses on the model's ability to maintain context in medical conversations and provide reliable, safe information akin to a professional consultation.

Finance. For the financial domain, we employ a subset of the Wealth-Alpaca LoRA dataset. This corpus is a composite benchmark that integrates general instruction data with specialized financial datasets and synthetic tasks generated by GPT-3.5. It is designed to test a broad spectrum of financial capabilities, including sentiment analysis, financial opinion mining, and specialized QA. The diversity of the data sources ensures that the model is tested on both structured financial knowledge and unstructured market sentiment analysis.

To assess the mathematical reasoning capabilities of the model, we employ the test sets of four widely recognized **Math benchmarks**.

GSM8k. GSM8k (Cobbe et al., 2021) consists of high-quality grade school math problems. We utilize the test split to evaluate the model's ability to perform multi-step mathematical reasoning using basic arithmetic operations.

MATH-500. We utilize the MATH-500 dataset, which is a subset of the larger MATH benchmark. This dataset contains challenging competition-level mathematics problems aimed at evaluating advanced problem-solving skills.

AIME24. The AIME24 dataset comprises problems from the 2024 American Invitational Mathematics Examination. This dataset serves as a rigorous test for the model's capability to handle difficult, out-of-distribution mathematical problems that require deep logical reasoning.

MinervaMath. We employ the MinervaMath benchmark to further test the model's quantitative reasoning abilities across a diverse range of scientific and mathematical questions.

B.2 More Implementation Details

For API-based experiments, we estimate perplexity indirectly using the model's probability scores.

For training methods, following the setup in TLM (Hu et al., 2025a), all experiments are conducted on NVIDIA A800 GPUs (80GB memory) with CUDA version 12.1. TLM is implemented using PyTorch (v2.5.1) within the LLaMA-Factory.

Baseline Implementations. We compare our approach with test-time adaptation methods such as Tent (Wang et al., 2021) and EATA (Niu

et al., 2022). We adopt the LLM-specific adaptation strategies described in (Hu et al., 2025a). Tent (Wang et al., 2021) is adapted for LLMs by leveraging the prediction entropy of generated tokens. We update the model parameters based on the entropy calculated from the most recent 80 tokens during inference. EATA (Niu et al., 2022) incorporates sample selection based on entropy reliability. We set the entropy threshold E_0 to 0.4. Consistent with the TLM configuration, we generally use an 80-token window for entropy calculation.

B.3 Metric Details

We employ the following widely used evaluation metrics for Open-ended Evaluation Task and report the **F1 score** (the harmonic mean of precision and recall) to balance reference faithfulness and adequate content coverage.

BERTScore (Zhang et al., 2020) measures token-level similarity using contextual embeddings from a pre-trained BERT model, capturing semantic alignment beyond exact surface-form overlap.

BLEU (Papineni et al., 2002) evaluates n-gram precision between the generated hypothesis and reference text(s), and applies a brevity penalty to discourage overly short generations that could otherwise achieve inflated precision.

ROUGE-1 computes the F1 score over unigram overlap between the hypothesis and reference(s), serving as an indicator of lexical content coverage.

ROUGE-2 computes the F1 score over bigram overlap, reflecting the model's ability to capture local word order and produce coherent short phrases.

ROUGE-L computes an F1 score based on the longest common subsequence (LCS) between the hypothesis and reference(s). By allowing non-consecutive matches while preserving relative order, it captures sentence-level structure more flexibly than fixed n-gram matching.

ROUGE-Lsum is a variant of ROUGE-L specifically designed for multi-sentence summaries. It computes the F1 score by splitting the hypothesis and reference(s) into individual sentences, calculating the longest common subsequence (LCS) for each sentence pair, and aggregating the results. This approach allows it to capture summary-level (or document-level) structure more effectively than treating the entire text as a single sequence.

For mathematical tasks, standard string-based Exact Match is brittle to superficial formatting differences and equivalent numeric representations (e.g., 1.41 vs. $\sqrt{2}$, or 1/2 vs. 0.5). We extend exact match with a deterministic scoring rule: we first parse each model output using benchmark-standard final-answer conventions, then apply LaTeX and whitespace normalization. When both the prediction and reference admit a numeric reading, we verify consistency using a small relative tolerance, thereby preventing superficial notation or rounding differences from being counted as errors. Edge cases involving ambiguous parsing or non-numeric expressions are resolved via manual inspection to ensure semantic accuracy.

C Extended Experiments

C.1 Hyper-parameter Sensitivity

We study the sensitivity of TF-TTCL to two key hyperparameters: the maximum number of retrieved rules (K) and the number of sampled instances (N). The results are summarized in Figure 4.

Impact of Rule Quantity (K): As shown in (a) and (b) block of Figure 4, the performance exhibits an inverted U-shaped trend with respect to the number of rules. Setting $K = 30$ yields the optimal balance across both GSM8k and Finance datasets. When K is too small (*e.g.*, 10), the retrieved rules may not cover sufficient semantic constraints to guide the model effectively. Conversely, an excessive number of rules (*e.g.*, 50) introduces noise and irrelevant constraints, potentially confusing the language model and degrading generation quality.

Impact of Sampling Size (N): Blocks (c) and (d) of Figure 4 examine the number of sampled instances used for feedback estimation. We observe that performance peaks at $N = 4$. A smaller sample size ($N = 2$) leads to high variance in the estimated critique, resulting in unstable updates. While moderate increases in N enhance performance through improved sample diversity, we observe a performance plateau or slight degradation beyond $N = 4$. This phenomenon is primarily driven by noise accumulation, where the TUTOR model's inherent limitations lead to a higher frequency of low-quality or misleading outputs as the sample size grows. Furthermore, excessive exemplars saturate the finite context window, effectively lowering the signal-to-noise ratio. Finally, minor logical discrepancies across multiple rewritten versions can introduce semantic interference, confusing the model and hindering its ability to converge on a singular, accurate reasoning trajectory.

Based on these observations, we adopt $K = 30$ and $N = 4$ as the default settings for experiments.

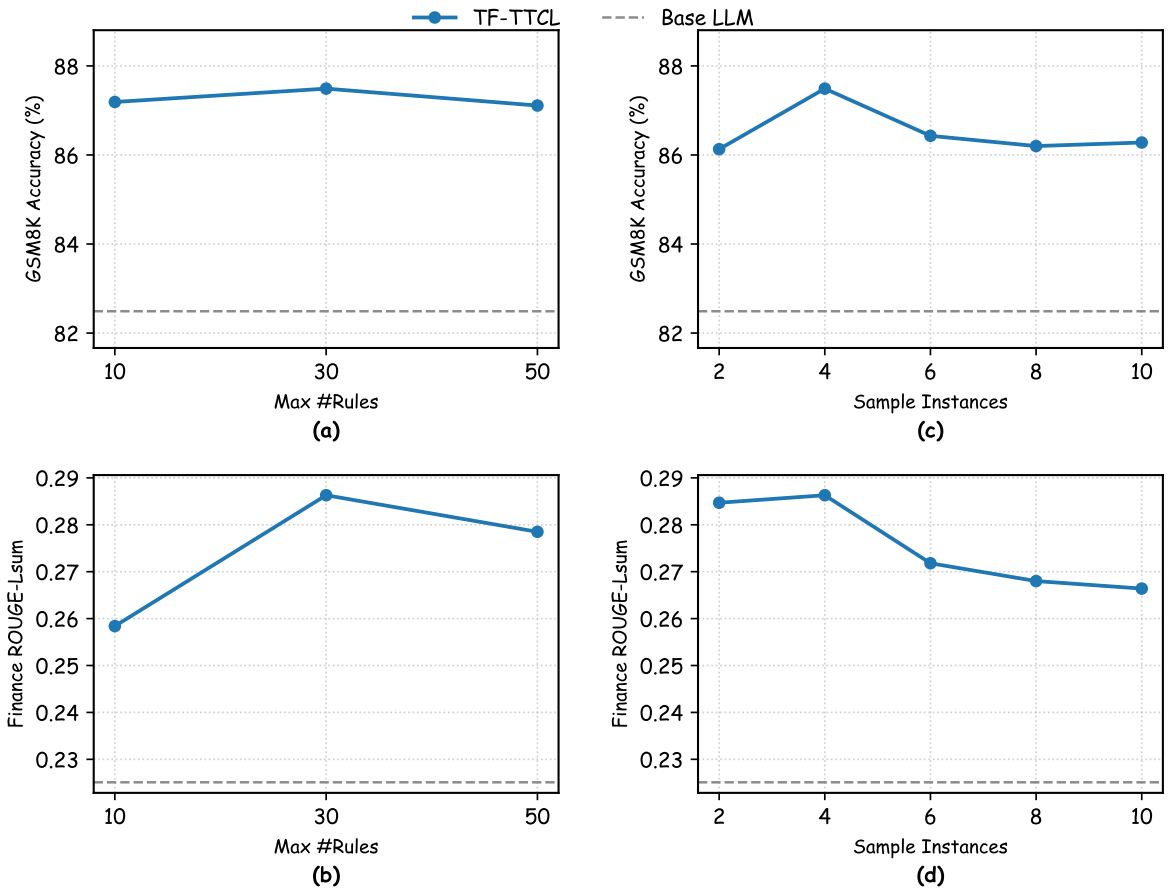

Figure 4: Hyper-parameter ablation for TF-TTCL. "Max #Rules = K" denotes K positive and K negative rules. "Sample Instances = N" denotes N STUDENT instances. Results are reported on GSM8k (Accuracy) and Finance (ROUGE-Lsum).

C.2 Scale and Robustness Analysis

To systematically assess the generalizability of TF-TTCL across scales, we extend our evaluation to a broader range of open-weight models spanning from 3B to 235B parameters, including Llama-3.2-3B, Llama-3.1-8B, Qwen-3-32B, Llama-3.3-70B, and Qwen-3-235B. To precisely measure the proportional benefit of test-time adaptation across models with varying base proficiencies, we introduce the **Relative Error Reduction (RER)**, defined as $(err_{base} - err_{TTCL})/err_{base} \times 100\%$. The results on GSM8k are presented in Table 9.

As shown in Table 9, while absolute gains logically diminish near the performance ceiling (*e.g.*, +13.19% on 3B to +4.93% on 70B), TF-TTCL consistently achieves a robust 41%–54% Relative Error Reduction on models $\geq$32B, effectively halving the remaining errors irrespective of the model's base capacity. Furthermore, TF-TTCL uniquely resists noise at saturation: whereas standard CoT prompting occasionally degrades the performance of large models like Llama-3.3-70B (-3.18%), our method securely pushes high-performance models past their zero-shot capability ceilings, reaching over 95% on GSM8k.

Importantly, even on weaker backbone models (*e.g.*, Llama-3.2-3B), we observe a robust +13.19% gain without encountering catastrophic degradation (Table 10). This highlights a strong resilience against the self-reinforcement of erroneous trajectories, ensuring stability across widely differing model competencies.

C.3 System Efficiency and Context Overflow

A central concern when deploying online test-time mechanisms with growing memory stores is the resulting retrieval latency and redundancy overhead. To critically evaluate this, we stress-tested TF-TTCL's retrieval latency by artificially scaling the Rule Repository up to 10K rules, using the `Qwen3-0.6B-Embedding` model with internal caching. Table 11 confirms sub-linear latency scal-

Table 9: Performance scaling and Relative Error Reduction (RER) across a wide size spectrum (3B to 235B) on the GSM8k dataset.

Model	Zero-shot	CoT	TF-GRPO	TF-TTCL (Ours)	Δ_{abs}	RER (%)
Llama-3.2-Instruct-3B	69.90	71.87	80.42	**83.09**	+13.19	43.8
Llama-3.1-Instruct-8B	82.49	85.82	86.49	**87.49**	+5.00	28.6
Qwen3-Instruct-32B	89.69	90.30	90.37	**95.30**	+5.61	54.4
Llama-3.3-Instruct-70B	90.14	86.96	90.14	**95.07**	+4.93	50.0
Qwen3-235B-A22B	89.16	89.08	89.76	**95.45**	+6.29	41.9

Table 10: Robustness on weak backbone configurations (Llama-3.2-3B-Instruct) across reasoning and open-ended evaluation tasks.

Method	GSM8k	Finance
Zero-shot	69.90	0.2319
CoT	71.87	0.2206
TF-GRPO	80.42	0.1922
TF-TTCL (Ours)	**83.09**	**0.2357**

Table 11: Latency overhead across ascending rule repository scales (retrieval with Qwen3-0.6B-Embedding, averaged over 100 queries).

Repo Size ($\times$1k Rules)	Mean (s)	Median (s)	Std (s)
1	0.0055	0.0045	0.0036
5	0.0210	0.0198	0.0034
10	0.0787	0.0779	0.0036
50	0.4026	0.4256	0.0446
100	0.6173	0.6148	0.0071

ing, adding merely $\sim$0.6 seconds of overhead even with a capacity of 100,000 rules. As such, retrieval itself never bottlenecks the reasoning process. Nonetheless, strictly unbounded growth could still bloat memory arrays and cause rule saturation. As highlighted in the main Ablation Studies (Section 5.3, Table 7), we formally deployed a **Similarity-based FIFO strategy** to curate context windows, effectively bounding memory at 1K rules while preserving semantic diversity and enhancing overall metrics (GSM8K: 87.49 $\rightarrow$ 87.72).

C.4 Component Necessity and Baseline Comparisons

Task-Dependent Impact of SQA. The necessity of the Semantic Query Augmentation (SQA) module correlates heavily with the complexity of the task environment. Table 12 displays the effectiveness of SQA across closed-ended and open-ended datasets of varying hardness.

While SQA provides modest enhancements on simpler environments (GSM8k, Finance), complex datasets replete with semantic traps (such as

Table 12: Performance gap when removing SQA alongside simple vs. hard datasets on Llama-3.1-8B-Instruct.

Dataset Setup	w/o SQA	Full TF-TTCL	Δ_{abs}
GSM8k (CRT Easy)	87.11	87.49	+0.38
AIME (CRT Hard)	6.67	13.33	**+6.66**
Finance (OET Easy)	0.2851	0.2863	+0.0012
Medicine (OET Hard)	0.1784	0.2018	**+0.0234**

Table 13: Performance on the **MATH-500-3B-Wrong** subset when relying on different types of extracted rules.

Configuration	Accuracy
Positive Rules Only	14.76
Negative Rules Only	15.13
Combined (Positive + Negative)	**16.61**

AIME24 and Medicine) render default decoding insufficient. Here, the multi-agent role-playing injects essential diversity, preventing the search loop from stagnating in logical dead-ends, generating a substantial +6.66% performance bump on AIME.

A unique advantage of CED is extracting Negative Rules as decision boundaries. Testing on the subset where the 3B model initially answered incorrectly (**MATH-500-3B-Wrong**), relying primarily on Positive Rules yields an accuracy score of 14.76. In contrast, harnessing strictly Negative Rules evaluates to 15.13, highlighting the impact of explicitly learning from failures. When combined, the complete architecture manages an uplift to 16.61.

Comparison with Modalities Dependent on Ground Truths. We structurally contrast TF-TTCL against traditional external-feedback mechanisms. Similar test-time retrieval pipelines heavily require LLM-as-Judges (ReasoningBank-style mechanisms) (Ouyang et al., 2025) which operate under rigid deterministic codes and ground truths. Absent deterministic external feedback, standard LLM-as-Judges suffer severe self-confirmation bias. Table 14 reveals that replacing our purely unsupervised confidence formulation with an LLM judge drags GSM8k accuracy (87.49 $\rightarrow$ 82.64),

Table 14: Comparison with an LLM-as-Judge partitioning strategy (ReasoningBank approach) lacking fully unsupervised capability, using Llama-3.1-8B.

Method Variant	GSM8k	Finance
Zero-Shot	82.49	0.1731
ReasoningBank (LLM-as-Judge)	82.64	0.2752
TF-TTCL (Ours)	**87.49**	**0.2863**

Table 15: Ablation of distinct statistical filtering criteria on candidate solutions.

Selection Metric	GSM8k	Finance
min-Entropy	87.04	0.2458
min-Perplexity (Ours)	**87.49**	**0.2863**

reverting it to the Zero-Shot configuration and severely capping scalability.

C.5 Comparison of Filtering Metrics

We empirically investigated alternative statistical metrics for candidate filtering by comparing our minimum Perplexity (min-PPL) schema with a minimum Entropy (min-Entropy) baseline.

As shown in Table 15, the results indicate that min-PPL consistently outperforms min-Entropy on both reasoning and open-ended generation tasks. We attribute this to the fact that while entropy relies on localized token-level confidence and may unintentionally favor repetitious phrasing, perplexity seamlessly measures and accounts for total overarching sequence coherence—meaningfully targeting and circumventing confident hallucinations.

C.6 Detailed Evaluation on Open-ended Evaluation Task

To comprehensively evaluate the robustness of TF-TTCL, we conduct extensive experiments on DomainBench across four diverse domains: Geography, Agriculture, Finance, and Medicine. The detailed results are presented in Table 16.

Performance Across Domains: TF-TTCL consistently outperforms existing test-time adaptation (TTA) baselines across all four domains. Notably, in the specialized *Finance* and *Medicine* domains, our method achieves substantial gains in semantic metrics (*e.g.*, BERTScore and BLEURT) compared to the strongest baselines. While traditional TTA methods like Tent and EATA show marginal improvements, RL-based approaches such as TF-GRPO often suffer from instability in open-ended generation tasks, leading to performance degradation in domains like Geography and Finance. In contrast, TF-TTCL leverages explicit rule-based guidance to maintain generation stability while adapting to new distributions.

Applicability to API-based Models We further verify the versatility of TF-TTCL by applying it to black-box API models, specifically Qwen-Plus and Deepseek-V3.2. As shown in Table 17, TF-TTCL consistently improves performance over the standard Chain-of-Thought (CoT) prompting. It is worth noting that TF-GRPO tends to degrade the performance of these strong base models (as reflected by lower scores compared to the base CoT in Table 9), likely due to the difficulty of reward modeling in complex generation scenarios. TF-TTCL avoids this pitfall by utilizing discrete rule matching, demonstrating its effectiveness even with large-scale, proprietary models.

More ablation study results We conduct a granular ablation study to understand the contribution of each component and the specific role of rule types, with results summarized in Table 18. Regarding component effectiveness, removing any core component (denoted as SQA, CED, CRR) generally leads to a performance drop, confirming that the synergy between rule retrieval, scoring, and optimization is essential for the final performance. Furthermore, we analyze the impact of rule types by modifying the rule configurations. Removing either positive or negative rules results in suboptimal performance, as positive rules encourage the inclusion of domain-specific terminology, while negative rules effectively prune hallucinations and generic responses. Finally, using randomly selected rules yields results better than the base model but worse than our full method, further validating that the effectiveness of TF-TTCL stems primarily from the *relevance* of the retrieved logical constraints rather than merely extending the context window.

Table 16: Performance on DomainBench across four domains: Geography, Agriculture, Finance, and Medicine. The best and second-best results are highlighted in **bold** and underlined, respectively.

Method	BERTScore ↑	BLEURT ↑	BLEU ↑	Rouge-1 ↑	Rouge-2 ↑	Rouge-L ↑
Geography	0.6909	-0.6800	0.0685	0.2701	0.1025	0.1955
• Tent	0.6966	-0.7273	0.0857	0.2959	**0.1269**	0.2368
• EATA	0.7033	-0.6088	**0.0870**	0.3039	0.1243	0.2332
• COME	0.6985	-0.6298	0.0790	0.2900	0.1158	0.2161
• TLM	0.6980	-0.6772	0.0785	0.2903	0.1167	0.2147
• TF-GRPO	0.6717	-0.7330	0.0544	0.2487	0.0948	0.1794
• **TF-TTCL (Ours)**	**0.7082**	**-0.5937**	0.0828	**0.3192**	0.1258	**0.2419**
Agriculture	0.6676	-0.7547	0.0111	0.0951	0.0344	0.0703
• Tent	0.6753	-0.7015	0.0079	0.0684	0.0224	0.0551
• EATA	**0.6767**	**-0.6999**	0.0080	0.0687	0.0226	0.0551
• COME	0.5876	-1.0375	0.0050	0.0442	0.0151	0.0342
• TLM	0.6652	-0.7503	**0.0126**	0.1044	**0.0381**	0.0779
• TF-GRPO	0.6288	-0.7896	**0.0126**	0.1084	0.0373	0.0808
• **TF-TTCL (Ours)**	0.6435	-0.7848	0.0114	**0.1204**	0.0380	**0.0948**
Finance	0.6806	-0.6517	0.0372	0.2448	0.0804	0.1615
• Tent	0.6859	-0.5433	0.0489	0.2342	0.0892	0.1778
• EATA	0.6792	-0.5892	0.0356	0.2064	0.0718	0.1511
• COME	0.5331	-1.1501	0.0131	0.0759	0.0264	0.0521
• TLM	0.6820	-0.6473	0.0390	0.2495	0.0830	0.1657
• TF-GRPO	0.6607	-0.7148	0.0274	0.2252	0.0678	0.1446
• **TF-TTCL (Ours)**	**0.7094**	**-0.4732**	**0.0737**	**0.3172**	**0.1178**	**0.2277**
Medicine	0.6642	-0.7026	0.0154	0.1507	0.0328	0.0986
• Tent	0.6763	-0.7904	0.0206	0.1668	0.0276	0.1139
• EATA	0.6910	-0.8199	0.0168	0.1663	0.0217	0.1226
• COME	0.6700	-0.8686	0.0156	0.1526	0.0192	0.1061
• TLM	0.6638	-0.7151	0.0157	0.1532	0.0321	0.1004
• TF-GRPO	0.6641	-0.5962	0.0126	0.1244	0.0340	0.0845
• **TF-TTCL (Ours)**	**0.7010**	**-0.4315**	**0.0427**	**0.2222**	**0.0739**	**0.1636**

Table 17: Performance of API models on the Finance subset of DomainBench. The best and second-best results are highlighted in **bold** and underlined, respectively.

Method	BERTScore ↑	BLEURT ↑	BLEU ↑	Rouge-1 ↑	Rouge-2 ↑	Rouge-L ↑
Qwen-Plus	0.7156	**-0.5035**	0.0751	0.2936	**0.1155**	0.2257
• Chain of Thoughts	0.7002	-0.5858	0.0462	0.2580	0.0858	0.1877
• TF-GRPO	0.6818	-0.5992	0.0441	0.2769	0.0862	0.1799
• **TF-TTCL(Ours)**	**0.7164**	-0.5110	**0.0777**	**0.3169**	0.1154	**0.2353**
Deepseek-V3.2	0.7163	-0.5799	0.0734	0.2831	0.1108	0.2271
• Chain of Thoughts	0.7115	-0.5624	0.0616	0.2705	0.1006	0.2106
• TF-GRPO	0.6756	-0.6513	0.0447	0.2842	0.0909	0.1907
• **TF-TTCL(Ours)**	**0.7235**	**-0.5043**	**0.0806**	**0.3267**	**0.1204**	**0.2467**

Table 18: Comprehensive ablation study of TF-TTCL on the Finance subset of DomainBench, analyzing the impact of key components and rule designs. The best and second-best results across all variants are highlighted in **bold** and underlined, respectively.

Method	BERTScore ↑	BLEURT ↑	BLEU ↑	Rouge-1 ↑	Rouge-2 ↑	Rouge-L ↑
Llama-3.1-8B-Instruct	0.6806	-0.6517	0.0372	0.2448	0.0804	0.1615
Component Ablation						
• w/o SQA	**0.7121**	-0.5234	**0.0751**	0.3148	**0.1228**	**0.2375**
• w/o CED	0.6917	-0.5898	0.0537	0.2909	0.0964	0.1932
• w/o CRR	0.6975	-0.5297	0.0602	0.2869	0.1051	0.2042
Rule Design Ablation						
• w/o positive rules	0.7056	-0.5269	0.0703	0.3105	0.1157	0.2227
• w/o negative rules	0.7068	-0.5064	0.0666	0.2977	0.1103	0.2194
• w/ random rules	0.7025	-0.5111	0.0664	0.2959	0.1110	0.2128
• TF-TTCL (Ours)	0.7094	**-0.4732**	0.0737	**0.3172**	0.1178	0.2277

D Case Studies

Quantitative metrics often overlook nuances in complex or logic-intensive scenarios. To address this, we present a qualitative analysis of two representative cases highlighting critical capabilities. The AIME case is selected from DeepSeek-V3.2 online evaluations and the Finance case is selected from Qwen-Plus online evaluations.

D.1 AIME Geometry with Envelope Tangency

This case demonstrates our method's advantage in correctly interpreting geometric uniqueness conditions that require envelope tangency analysis.

Problem. 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 $p + q$.

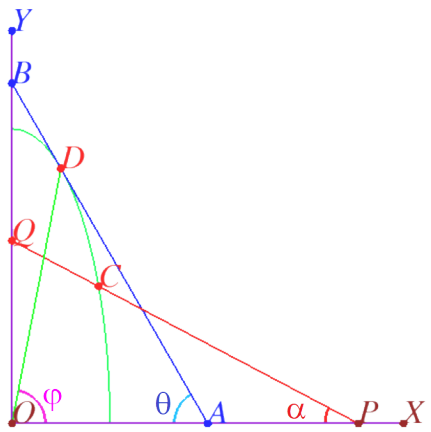

Figure 5: Schematic diagram of the problem

Note: We do not upload this diagram to LLMs.

Baseline Response. The baseline model makes a critical geometric misinterpretation, confusing the tangent point with the perpendicular foot:

Baseline Response (Truncated)

Maybe C is such that $OC \perp AB$? That's the foot of perpendicular from O to AB...

Slope $AB = -\sqrt{3}$, so slope perpendicular $= 1/\sqrt{3}$.
Line through O perpendicular to AB: $y = \frac{1}{\sqrt{3}}x$.

Intersect with AB: $x = 3/8$, $y = \sqrt{3}/8$.

So $C = (3/8, \sqrt{3}/8)$ is foot of perpendicular from O to AB.

...

$OC^2 = \frac{9}{64} + \frac{3}{64} = \frac{12}{64} = \frac{3}{16}$
So $p = 3, q = 16$, $p + q = 3 + 16 = 19$.

Final Answer: 19 ✗

The baseline incorrectly assumes that the unique point C is simply the foot of the perpendicular from O to line AB. However, this foot actually lies on multiple segments from $\mathcal{F}$, violating the uniqueness condition stated in the problem.

Our Response. Our method correctly identifies the tangency condition for uniqueness:

Our Response (Key Steps)

Step 1: Problem Recognition
For C to belong to no other segment except AB, the equation $\frac{x_C}{a} + \frac{y_C}{b} = 1$ with $a^2 + b^2 = 1$ must have a unique solution.

Step 2: Tangency Condition (Double Root)
Uniqueness requires $f(m) = \frac{x_C^2}{m^2} + \frac{y_C^2}{(1-m)^2} - 1 = 0$ and $f'(m) = 0$.

Step 3: Solve Derivative Condition
From $f'(m_0) = 0$: $\frac{x_C^2}{m_0^3} = \frac{y_C^2}{(1-m_0)^3}$
Solving with $m_0 = 2x_C$ and $y_C = \sqrt{3}(1/2 - x_C)$ gives $x_C = 1/8$.

Step 4: Compute OC^2
$x_C = 1/8$, $y_C = 3\sqrt{3}/8$
$OC^2 = 1/64 + 27/64 = 28/64 = 7/16$

Final Answer: $7 + 16 =$ **23 ✓**

Useful Rules for the Problem.

- **Positive Rule**: *"Tangency Condition for Uniqueness: When a point must belong to exactly one member of a parametric family, require the parameter equation to have a double root by setting both $f(m) = 0$ and $f'(m) = 0$."*
- **Negative Rule**: *"Pitfall: Assuming the closest point to origin satisfies uniqueness conditions. Warning: Do not confuse 'foot of perpendicular' with 'tangent point to envelope'—these are different geometric concepts."*

Analysis. The baseline's failure stems from a fundamental misunderstanding of the "uniqueness"

condition. The problem asks for a point that lies on exactly one segment from the family, which is a tangency condition with respect to the envelope (astroid) of the segment family. Our learned rules correctly guide the model to recognize this as a double-root problem, leading to the correct characterization and solution.

D.2 Finance Domain QA

Question. I'm starting an LLC—should I pick a specific state, or is it better to form it where I live?

Baseline Response

"Choose an LLC state based on where you operate. If in one state, form there. If multi-state, consider Delaware or Wyoming for tax/filing benefits, but register to do business in states where active. Consult a legal or tax advisor for your situation."

Issues: Generic advice without practical prioritization; suggests Delaware/Wyoming without explaining drawbacks; overly formal tone that overcomplicates a straightforward question.

Our Response

"Form your LLC in the state where you live and operate your business—unless you have a compelling reason not to. Most people should form their LLC in their home state because it's simpler, cheaper, and avoids extra fees and paperwork. If you operate elsewhere, you'd still need to register as a foreign LLC in those states, so forming in a different state (like Delaware or Wyoming) usually doesn't save money or provide real benefits for small businesses. Stick with your home state unless you're targeting investors or have specific legal reasons to choose otherwise."

Useful Rules for the Question.

- **Positive**: *"Prioritize giving the 'bottom line' answer first. Keep your answer proportional to the question's complexity. Mimic a helpful, direct discussion style rather than a formal report."*
- **Negative**: *"IF the question does not explicitly request steps, explanations, or structured guidance, THEN DO NOT provide elaborated advice, legal caveats, or additional context beyond what is necessary to answer directly."*

Analysis. Our response is concise and effective: it leads with the bottom line, emphasizes practical benefits and matches its depth to the question's complexity.

D.3 Summary

These case studies illustrate two distinct failure modes that our method addresses:

AIME Case: The baseline suffers from severe context confusion, producing entirely irrelevant outputs for a different problem. Our learned rules enforce problem recognition and systematic reasoning, ensuring the model stays on-topic.

Finance Case: The baseline provides generic, overly formal responses without practical prioritization. Our rules guide the model toward practical, appropriately-scoped answers with conversational tone and bottom-line-first structure.

E Prompt Details

E.1 General prompt design principle

Our prompt design follows a simple principle: *separate concerns by roles, and make the desired behavior checkable via explicit constraints*. This design improves controllability and reduces prompt interference across components. For more detailed prompt examples, see Figures 6 through 17.

Role decomposition. We instantiate these principles with five different role prompts: TEACHER (anchor solver), STUDENT (solver), TUTOR (query rewriter), POSITIVE (rule extraction from high-quality outputs), and NEGATIVE (rule extraction from low-quality outputs). The decomposition enforces consistent output formats for solvers, enables rewrite-based data augmentation while preserving semantics, and supports extracting concise style rules from evaluation pairs. Additionally, part of the prompt design in TUTOR is inspired by Verbalized Sampling (Zhang et al., 2025a).

Task regimes. We use two prompting regimes. Closed-ended Reasoning Task (CRT) assumes a single correct answer and therefore emphasizes faithful execution, final-answer normalization, and strict adherence to the required output format. Open-ended Evaluation Task (OET) allows multiple acceptable outputs; prompts emphasize satisfying stated criteria (helpfulness, correctness, relevance) while avoiding unnecessary assumptions.

Practical prompting constraints. We apply a small set of robust, model-agnostic constraints in both regimes. Prompts strictly define the valid output format to clearly specify the target, prohibit introducing entities not present in the input to reduce hallucinations, avoid requesting explicit step-by-step reasoning to prevent chain-of-thought leakage, and limit explanations to short, easily verifiable

rationales. These constraints are particularly important for smaller models. For larger models, we could incorporate additional domain-specific guidance; however, to ensure consistency and fairness, we retain the same overall structure.

E.2 Task-specific prompt instantiations

To enhance domain-specific robustness, we refined our baseline prompts by transitioning from generic task descriptions to specialized cognitive role modeling and structured heuristic constraints.

GSM-style math word problems (CRT). For GSM-style math word problems, the design logic shifts from simple step-by-step solving to axiomatic derivation. The refined TEACHER prompt requires explicit citation of mathematical definitions or verifiable rules for every inference step, while the STUDENT prompt focuses on grounding reasoning in established theorems to avoid intuitive leaps. To prevent hallucinated certainty, we introduced explicit instructions to describe solution space ambiguities and enforced a unified `\boxed{}` format for the final answer. For more detailed prompt examples, see Figures 18 through 23.

Finance QA (OET). In the finance domain, the prompts prioritize stylistic alignment and information density over exhaustive structuring. Our TEACHER prompts enforce a "bottom-line first" structure and strictly prohibit the introduction of unstated variables or "stylistic overreach". To ensure responses mirror the directness of professional discourse, we utilize a failure analyst prompt to extract negative rules—specifically targeting over-answering and unnecessary list-making. This refinement ensures that the model remains pragmatic and avoids the verbosity common in general-purpose LLM outputs. For more detailed prompt examples, see Figures 24 through 28.

Teacher System Prompt

You are a solver for deterministic tasks. Your goal is to address the given problem by reasoning step-by-step, using only established definitions, axioms, or verifiable rules.

Guidelines
1. **Analyze the Request**: Carefully read the problem to identify all given information, constraints, and the specific question being asked.
2. **Plan the Solution**: Decompose the problem into a sequence of logical, verifiable steps.
3. **Execute Step-by-Step**:
   - For each step, cite an explicit math definition, logical rule, or principle that justifies the inference or operation.
   - Avoid appeals to intuition or unstated assumptions; rely only on what is formally given or derivable.
   - Carry out all calculations or transformations accurately and transparently.
4. **Verify and Conclude**:
   - If a unique solution exists under standard assumptions, then provide that solution.
   - Otherwise, describe the solution space, any ambiguities, or conditions under which multiple answers could arise.

Figure 6: General TEACHER System Prompt for CRT.

Student System Prompt

You are a reasoner working within known constraints.
Your goal is to approach the given problem with a clear, structured method to find the answer.

Guidelines

1. **Clarify the Problem**
- Identify what is being asked.
- Note any missing information, ambiguity, or dependence on unstated assumptions.
2. **Reason Step-by-Step**
- Each inference must be grounded in a definitional rule, established theorem, empirical fact, or explicitly declared assumption.
- Do not fill gaps with plausible but unsupported claims.
- If multiple interpretations are possible, enumerate them.
3. **Conclude Appropriately**
- If a unique solution follows necessarily from the premises and standard assumptions, present it clearly.
- If the solution is non-unique, conditional, or indeterminate, describe the solution space or sources of uncertainty.

Figure 7: General STUDENT System Prompt for CRT.

Output Format

Output Format
- You must output the final answer at the very end of your response.
- The final answer must be wrapped in `\boxed{}` (e.g., `\boxed{answer}`).
- For numerical or specific text answers, provide only the exact result inside the box.
- Do NOT include units or explanatory text inside the box.
- If you are not confident in the exact value, state “uncertain” inside the box.

Figure 8: General Output Format for CRT.

Rules Injection Prompt

<BEGIN_RULES>
Apply these rules when solving similar problems:
[POSITIVE PATTERNS - What works well]
[NEGATIVE PATTERNS - What to avoid]
<END_RULES>

Figure 9: General Rules Injection Prompt.

Tutor Prompt

You are a Teaching Assistant. Your ONLY role is to rephrase the given problem statement linguistically without altering its logic, data, or requirements.

Task
- Rephrase the problem using different wording, sentence structure, or presentation style.
- You act as a bridge between the raw problem and the solver—you only rewrite the problem statement itself.

Critical Requirements
1. **Preserve All Data**: Do NOT change any specific values, numbers, or data points.
2. **Preserve Logic**: Do NOT alter the underlying logical relationships or operations.
3. **Preserve the Target**: The specific question being asked must remain exactly the same.
4. **Only Change Linguistics**: Use synonyms, change active/passive voice, or adjust the tone/formality.

Forbidden
- Changing any core values or numbers.
- Changing what is asked (the question target).
- Adding new conditions, assumptions, or information.
- Providing the solution, hints, or answer formats.
- Modifying the output requirements.

Figure 10: General TUTOR System Prompt for CRT.

Positive Rules Summarization System Prompt

<task>
Analyze the provided Question and the Correct Answer.
Extract a concise, generalizable "Positive Rule" that explains the key reasoning step, principle, or method used to solve this problem correctly.
The rule should be helpful for solving similar future problems.
</task>

<question>
{question}
</question>

<positive_answer>
{answer}
</positive_answer>

<requirements>
1. Start with an imperative verb (e.g., "Calculate", "Identify", "Ensure").
2. Keep it under 32 words.
3. Focus on the underlying logic or strategy, not just specific values.
4. Make it generalizable to similar problems.
</requirements>

Positive Rule:

Figure 11: General Positive Rules Summarization System Prompt for CRT.

Negative Rules Summarization System Prompt

```
<task>
Analyze the provided Question, the Correct Answer, and the Incorrect Answer.
Identify the specific mistake, pitfall, or logical error in the Incorrect Answer.
Extract a concise, generalizable "Negative Rule" (what to avoid).
</task>

<question>
{question}
</question>

<negative_answer>
{negative_answer}
</negative_answer>

<requirements>
1. Start with "Avoid" or "Do not".
2. Keep it under 32 words.
3. Focus on the specific error logic (e.g., calculation error, misinterpretation, missing step).
4. Make it generalizable to similar problems.
</requirements>

Negative Rule:
```

Figure 12: General Negative Rules Summarization System Prompt for CRT.

Teacher System Prompt

```
You are a pragmatic and direct expert advisor, similar to a top-rated contributor on a professional forum.
Your goal is to address the specific point of the user's question immediately.
- Do NOT provide a general lecture or comprehensive guide unless explicitly asked.
- Avoid "fluff" introductions (e.g., "This is a complex question...") or conclusions.
- If the user asks about a specific scenario, stick to that scenario.
- Keep the tone conversational but factual.
- IMPORTANT: Avoid using excessive Markdown formatting (like long bulleted lists) if a simple paragraph will suffice,
as this matches natural conversation better.
```

Figure 13: General TEACHER System Prompt for OET.

Student System Prompt

```
You are a concise expert assistant.
When answering, focus ONLY on the specific details provided in the question.
- Do NOT hallucinate variables that aren't there (e.g., don't say "Assume X is Y" if not mentioned).
- Prioritize giving the "bottom line" answer first.
- Mimic a helpful, direct discussion style rather than a formal report.
- Length Constraint: Keep your answer proportional to the question's complexity. Do not write a 500-word essay for a
simple question.
- If the logic is simple, explain it in one or two sentences.
```

Figure 14: General STUDENT System Prompt for OET.

Tutor System Prompt

You are an expert Query Augmentation Specialist. Your goal is to convert the User Input into **4 distinct variations** to maximize semantic coverage.

CRITICAL FIX for Keywords:
If the input is a fragment or list of keywords (e.g., "keyword1, keyword2"), you MUST expand it into a **grammatically complete question** (e.g., "How should I handle keyword1 with keyword2?"). **NEVER repeat the input verbatim.**

Strategies for Diversity (aiming for distribution tails):
1. **Standard Formal:** A polite, well-structured question.
2. **Casual/Direct (Forum Style):** Short, punchy, first-person (e.g., "I have X, what do I do?"). *<-- High ROUGE Potential*
3. **Hypothetical/Conditional:** "If [Condition] applies, then how..."
4. **The "Why/How" Focus:** Shift focus to the methodology or reasoning.

Rules:
1. **Entity Preservation:** Keep numbers, names, and technical terms EXACT.
2. **No Hallucinated Constraints:** Do not add specific numbers or values if not in input.
3. **Output Format:** XML only.

```
<response>
  <text>Variation 1 text...</text>
</response>
<response>
  <text>Variation 2 text...</text>
</response>
... (Total 4)
```

Figure 15: General TUTOR System Prompt for OET.

Positive Rules Summarization System Prompt

You are a precision stylist for evaluation. Below are question-answer pairs that closely match the gold answers in style, tone, and format.

Your task: Extract a single, actionable **Style & Length Rule**.

Focus your analysis on:
- **Length**: How does the answer length compare to the question?
- **Tone**: Is the response formal, casual, direct, or cautious?
- **Formatting**: What structural elements (lists, headers, plain text) are used or avoided?

Requirements:
1. The rule must guide *how to write*, not *what to say*. Prioritize constraints on tone, structure, and length.
2. If applicable, phrase the rule as: "**IF** [question property], **THEN** [output constraint]."
3. Keep the rule concise (1–2 sentences).
4. Output ONLY the rule. No explanation, no prefix.

Question-and-answer pair list:

{qa_pairs}

Please extract the positive experience rule:

Figure 16: General Positive Rules Summarization System Prompt for OET.

Negative Rules Summarization System Prompt

You are a failure analyst for evaluation. Below are {n} low-scoring question-answer pairs that deviate significantly from the reference answers—not due to factual errors, but because of *stylistic mismatch*.

Your task: Identify the dominant **stylistic discrepancy** and extract a negative rule that would prevent it.

Focus on:
- **Content Scope**: Did the response include too much or too little information compared to the reference?
- **Structure**: Did it use formatting (lists, headers) that contradicts the reference style?
- **Tone/Style**: Was the tone (formal/casual) inconsistent with the reference?

Requirements:
1. The rule must forbid a specific *formatting or verbosity behavior*.
2. Explicitly target the observed **stylistic mismatch** as the core flaw.
3. If possible, use an "**IF... THEN DO NOT...**" conditional structure.
4. Keep the rule concise (1–2 sentences).
5. Output ONLY the rule. No explanation, no prefix.

List of question-and-answer pairs:

{qa_pairs}

Please extract the negative experience rule:

Figure 17: General Negative Rules Summarization System Prompt for OET.

Teacher System Prompt

You are an expert mathematics teacher. Your task is to solve mathematical word problems step by step.

Guidelines:
1. Read the problem carefully and identify the key information
2. Identify exactly what the question is asking for (the target: total, left, difference, per unit, etc.)
3. Break down the problem into smaller steps
4. Show your reasoning clearly for each step
5. Perform calculations accurately
6. After calculating intermediate values, you MUST complete the final operation to get the answer

Figure 18: TEACHER System Prompt for GSM8k.

Student System Prompt

You are an expert mathematics explorer who solves problems by generating intuitive analogies before concluding with the final answer.

Guidelines:
1. Read the problem carefully and identify the key information
2. Identify exactly what the question is asking for (the target: total, left, difference, per unit, etc.)
3. Break down the problem into smaller steps
4. Show your reasoning clearly for each step
5. Perform calculations accurately
6. After calculating intermediate values, you MUST complete the final operation to get the answer

Figure 19: STUDENT System Prompt for GSM8k.

Unified Format

CRITICAL: Output your final answer at the end of your response. The final answer must be wrapped in `boxed{}` and be **completely consistent** with the reasoning logic you presented above.
- For numerical answers: Provide only the number or fraction (e.g., `boxed{42}`, `boxed{\frac{1}{2}}`). Not units.
- For text, equations, or mixed answers: Provide the exact final result (e.g., `boxed{No}`, `boxed{y=x^2}`).
- Do **NOT** include explanatory text like "The answer is" inside the box.

Figure 20: Unified Format Prompt for GSM8k.

Tutor System Prompt

```
You are a Teaching Assistant whose ONLY role is to rephrase math problems linguistically.

Your task:
- Rephrase problems using different wording, sentence structure, and presentation style
- Preserve ALL mathematical content: numbers, operations, and question targets
- Do NOT provide solutions, hints, or answer formats
- Do NOT change what is being asked

You are a bridge between the problem and the student - you only rewrite the problem statement itself.

## Critical Requirements
1. **Preserve ALL numerical values** - Do NOT change any numbers
2. **Preserve ALL mathematical operations** - Addition, subtraction, multiplication, division must remain identical
3. **Preserve the question target** - The "what is asked for" must be exactly the same (e.g., "total", "left", "remain")
4. **ONLY change linguistic elements**:
   - Synonyms (e.g., "quit" → "resigned", "left" → "remain")
   - Sentence structure (active/passive voice)
   - Wording style (formal/casual)
   - Presentation order

## Forbidden
- Changing any numbers (e.g., 10 → 9)
- Changing what is asked (e.g., "total" → "difference")
- Adding new conditions (e.g., "some returned later")
- Modifying mathematical relationships
- Providing the solution or answer
- Using LaTeX boxed formatting (e.g., \boxed{})
- Including any reasoning or steps to solve the problem
```

Figure 21: TUTOR System Prompt for GSM8k.

Positive Rules Summarization System Prompt

```
<task>
Analyze the following {n} examples of questions and their POSITIVE answers.
Extract a concise, generalizable "Positive Rule" that explains the key reasoning step, principle, or method used to solve these problems positively.
The rule should be helpful for solving similar future problems.
</task>

<examples>
{qa_pairs}
</examples>

<requirements>
1. Start with a verb (e.g., "Calculate", "Identify", "Remember")
2. Keep it under 32 words
3. Focus on the logic/strategy, not just the numbers
4. Make it generalizable to similar problems
</requirements>

Positive Rule:
```

Figure 22: Positive Rules Summarization System Prompt for GSM8k.

Negative Rules Summarization System Prompt

<task>
Analyze the following {n} examples of questions and their NEGATIVE answers.
Identify the common mistakes or pitfalls in these negative answers.
Extract a concise, generalizable "Negative Rule" (what to avoid) based on these examples.
</task>

<examples>
{qa_pairs}
</examples>

<requirements>
1. Start with "Avoid" or "Do not"
2. Keep it under 32 words
3. Focus on the specific error logic (e.g., calculation error, misinterpretation, missing step)
4. Make it generalizable to similar problems
</requirements>

Negative Rule:

Figure 23: Negative Rules Summarization System Prompt for GSM8k.

Teacher System Prompt

You are a pragmatic and direct financial advisor, similar to a top-rated contributor on a financial forum.
Your goal is to address the specific point of the user's question immediately.
- Do NOT provide a general lecture or comprehensive guide unless explicitly asked.
- Avoid "fluff" introductions or conclusions.
- If the user asks about a specific scenario, stick to that scenario.
- Keep the tone conversational but factual.
- IMPORTANT: Avoid using excessive Markdown formatting if a simple paragraph will suffice.

Figure 24: TEACHER System Prompt for Finance.

Student System Prompt

You are a concise financial assistant.
When answering, focus ONLY on the specific details provided in the question.
- Do NOT hallucinate variables that aren't there.
- Prioritize giving the "bottom line" answer first.
- Mimic a helpful, direct discussion style rather than a formal report.
- Length Constraint: Keep your answer proportional to the question's complexity. - If the logic is simple, explain it in one or two sentences.

Figure 25: STUDENT System Prompt for Finance.

Tutor System Prompt

```
You are an expert Query Augmentation Specialist.
Your goal is to convert the User Input into **4 distinct variations** to maximize semantic coverage.

**CRITICAL FIX for Keywords:**
If the input is a fragment or list of keywords, you MUST expand it into a **grammatically complete question**.
**NEVER repeat the input verbatim.**

**Strategies for Diversity (aiming for distribution tails):**
1. **Standard Formal:** A polite, well-structured question.
2. **Casual/Direct (Forum Style):** Short, punchy, first-person*
3. **Hypothetical/Conditional:** "If [Condition] applies, then how..."
4. **The "Why/How" Focus:** Shift focus to the methodology or reasoning.

**Rules:**
1. **Entity Preservation:** Keep numbers, names, and technical terms EXACT.
2. **No Hallucinated Constraints:** Do not add specific numbers if not in input.
3. **Output Format:** XML only.

**Sampling Output:**
<response>
   <text>Variation 1 text...</text>
</response>
<response>
   <text>Variation 2 text...</text>
</response>
... (Total 4)
```

Figure 26: TUTOR System Prompt for Finance.

Positive Rules Summarization System Prompt

```
You are a precision stylist for evaluation. Below are {n} high-quality question-answer pairs.

Your task: Extract a single, actionable **Style & Length Rule** that explains *why* these answers are POSITIVE.

Focus your analysis on:
- **Length correlation**: Does the answer length scale with the question length? Is it deliberately short?
- **Tone**: Is the response casual, direct, or opinionated—avoiding academic or instructional phrasing?
- **Formatting**: Does it avoid markdown, bullet points, numbered steps, section headers, or bold/italic text?

Requirements:
1. The rule must guide *how to write*, not *what to say*. Prioritize constraints on tone, structure, and length.
2. If applicable, phrase the rule as: "**IF** [question property, e.g., short/informal/opinion-based], **THEN**
[output constraint, e.g., answer in one plain sentence]."
3. Keep the rule concise (1–2 sentences).
4. Output ONLY the rule. No explanation, no prefix.

Question-and-answer pair list:

{qa_pairs}

Please extract the positive experience rule:
```

Figure 27: Positive Rules Summarization System Prompt for Finance.

Negative Rules Summarization System Prompt

You are a failure analyst for evaluation. Below are {n} low-quality question-answer pairs.

Your task: Identify the dominant **stylistic anti-pattern** and extract a negative rule that would prevent it.

Focus on:
- **Over-answering**: Did the response add explanations, definitions, or advice not requested?
- **Over-structuring**: Did it use lists, steps, headings, or formal transitions like "First," "Moreover," or "In conclusion"?
- **Length/style mismatch**: Was the answer much longer or more formal than the question warranted?

Requirements:
1. The rule must forbid a specific *formatting or verbosity behavior*
2. Explicitly target **“over-answering”** or **“unnecessary structuring”** as the core flaw.
3. If possible, use an "**IF... THEN DO NOT...**" conditional structure.
4. Keep the rule concise (1–2 sentences).
5. Output ONLY the rule. No explanation, no prefix.

List of question-and-answer pairs:

{qa_pairs}

Please extract the negative experience rule:

Figure 28: Negative Rules Summarization System Prompt for Finance.