
AdaCoT: Pareto-Optimal Adaptive Chain-of-Thought Triggering via Reinforcement Learning

Anonymous Author(s)

Affiliation

Address

email

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities but often face challenges with tasks requiring sophisticated reasoning. While Chain-of-Thought (CoT) prompting significantly enhances reasoning, it indiscriminately generates lengthy reasoning steps for all queries, leading to substantial computational costs and inefficiency, especially for simpler inputs. To address this critical issue, we introduce AdaCoT (Adaptive Chain-of-Thought), a novel framework enabling LLMs to adaptively decide when to invoke CoT. AdaCoT framed adaptive reasoning as a Pareto optimization problem that seeks to balance model performance with the costs associated with CoT invocation (both frequency and computational overhead). We propose a reinforcement learning (RL) based method, specifically utilizing Proximal Policy Optimization (PPO), to dynamically control the CoT triggering decision boundary by adjusting penalty coefficients, thereby allowing the model to determine CoT necessity based on implicit query complexity. A key technical contribution is Selective Loss Masking (SLM), designed to counteract decision boundary collapse during multi-stage RL training, ensuring robust and stable adaptive triggering. Experimental results demonstrate that AdaCoT successfully navigates the Pareto frontier, achieving substantial reductions in CoT usage for queries not requiring elaborate reasoning. For instance, on our production traffic testset, AdaCoT reduced CoT triggering rates to as low as 3.18% and decreased average response tokens by 69.06%, while maintaining high performance on complex tasks. This substantial token decrease directly translates to a significant reduction in inference computational load. AdaCoT pioneers adaptive CoT triggering, offering a practical and principled solution for developing more efficient, responsive, and cost-effective LLMs, particularly crucial for interactive and resource-sensitive applications.

1 Introduction

Large Language Models (LLMs) have garnered substantial attention due to their remarkable ability to encode extensive world knowledge from vast corpora [1], enabling impressive performance across diverse tasks such as question answering, creative writing, and summarization. Despite these successes, LLMs often demonstrate limitations in tasks requiring sophisticated reasoning, including solving complex mathematical problems and intricate coding puzzles. To mitigate this, recent methodologies have employed Chain-of-Thought (CoT) prompting [2], which encourages models to explicitly generate step-by-step reasoning prior to producing final answers. This approach significantly enhances the reasoning capability of models, even achieving human-expert levels in certain domains [3, 4, 5, 6].

However, employing CoT prompting also poses critical challenges during inference [7]. Specifically, it substantially increases the number of tokens generated, even for simple queries that do not benefit from elaborate reasoning, such as straightforward arithmetic questions. This indiscriminate token expense consequently raises deployment costs and reduces inference efficiency. Ideally, a model should adaptively determine when detailed reasoning is necessary. For instance, simple queries like "What is 1+1?" should be answered immediately without additional reasoning steps, whereas more complex queries require deeper and step-by-step reasoning. An adaptive strategy would thus optimize token usage, balancing cost-efficiency with response quality.

Recently, a few efforts have made attempts towards this direction. These approaches can be broadly categorized into three main directions. (1) Incorporate length penalties or brevity rewards during the reinforcement learning (RL) stage to encourage shorter, more concise reasoning paths [8, 9, 10, 11, 12, 13]. (2) Restructure CoT outputs through post-processing such as iterative summarization or pruning [14, 15, 16, 17, 18, 19, 20, 21]. (3) Employ explicit user instructions or hand-crafted selection mechanisms to control the use of CoT [22, 23, 24, 25, 26, 27]. Despite their contributions, they mainly focus on monotonic reasoning reduction, failing to account for the nuanced variability in query complexity, i.e., treating simple and difficult prompts adaptively. Moreover, they lack a principled optimization framework to guide balancing response quality against deployment cost.

To address these limitations, we introduce AdaCoT (Adaptive Chain-of-Thought), a novel approach grounded in formal mathematical analysis. Our key insight is framing adaptive reasoning as a multi-objective optimization problem with two competing goals: maximizing response accuracy and minimizing deployment costs. Specifically, we formalize this balance through Pareto optimization, seeking optimal trade-offs between reasoning complexity and inference efficiency. Such a mathematical framework provides clear theoretical grounding for dynamically adapting CoT triggering based on query complexity.

Leveraging this formalization, we propose an RL-based strategy explicitly designed around the Pareto optimization framework, enabling effective control of the model’s decision boundary for initiating CoT prompting. During training, the RL agent dynamically assesses the complexity of incoming user queries to determine the necessity and extent of reasoning steps. By carefully designing the reward function to incorporate penalty coefficients, we encourage the RL agent to seek solutions along the Pareto frontier, explicitly optimizing trade-offs between response accuracy and token expenditure. This structured exploration enables the model to effectively discern when detailed reasoning is beneficial, thereby systematically enhancing inference efficiency and significantly reducing deployment costs.

The proposed AdaCoT framework delivers substantial benefits in LLM operational efficiency. By empowering models to selectively engage CoT, AdaCoT can reduce triggering rates to as low as 3.18% and cut average response tokens by 69.1% in production settings. This significant reduction in computational load is achieved while maintaining strong performance on 15 widely-adopted benchmarks. For example, AdaCoT can achieve a 62.8% average score using only a 53.3% CoT rate, closely rivaling the 65.0% score of a model that always employs CoT. These improvements directly translate to more cost-effective and responsive LLM systems.

2 The AdaCoT Framework

Enabling a large language model (LLM) to dynamically decide whether to invoke Chain-of-Thought (CoT) reasoning based on the complexity of user queries is a critical task, which allows LLMs to allocate computational resources more rationally, i.e., spending tokens on complex reasoning tasks while avoiding unnecessary overhead for simple queries.

To achieve this, we introduce AdaCoT, a unified framework for adaptive reasoning. The central insight is that the decision to employ CoT prompting can be naturally cast as an optimization problem involving two competing goals: (1) maximizing response accuracy and (2) minimizing deployment costs. To capture this trade-off, we frame the task as a Pareto optimization problem, providing a principled foundation for balancing reasoning quality against computational efficiency. Based on this formulation, we propose an RL-based control strategy that governs the invocation of CoT reasoning. Specifically, we train a policy model learning to assess the complexity of each query and decide whether CoT reasoning should be used. During RL training, the policy model is optimized using a reward signal carefully designed to reflect the Pareto trade-off, incorporating both task performance

and token efficiency. Through this RL-based mechanism, AdaCoT learns to allocate reasoning effort adaptively, yielding high-quality responses with minimal computational cost.

2.1 Adaptive Reasoning as a Pareto Optimization Problem

We formulate the adaptive reasoning challenge as a Pareto optimization problem, aiming to simultaneously maximize model performance and minimize CoT usage. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a dataset of query-response pairs, where x_i is the input query and y_i is the ground truth response. Let f_θ be an LLM parameterized by θ , and let $r_\theta(x_i)$ be the response generated by the model for input x_i .

To CoT usage is measured by the CoT triggering rate $T(\theta)$, defined as the proportion of responses that include reasoning:

$$T(\theta) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}[\text{HasReasoning}(r_\theta(x_i))] \quad (1)$$

where $\mathbf{1}[\cdot]$ is the indicator function and $\text{HasReasoning}(\cdot)$ determines if a response contains explicit CoT steps (e.g., non-empty content within `<think>...</think>` tags).

On the other hand, model performance $P(\theta)$ is defined as the average score on a set of evaluation metrics:

$$P(\theta) = \frac{1}{M} \sum_{j=1}^M \text{Score}_j(\theta) \quad (2)$$

where M is the number of evaluation instances/metrics and $\text{Score}_j(\theta)$ is the model’s score on the j -th evaluation.

Putting CoT usage $T(\theta)$ and model performance $P(\theta)$ together, the objective is to find model parameters θ^* that achieve an optimal trade-off:

$$\theta^* = \arg \max_{\theta} \{\lambda_P \cdot P(\theta) - \lambda_T \cdot T(\theta)\} \quad (3)$$

or more generally, to find solutions on the Pareto frontier of $(P(\theta), 1 - T(\theta))$. The hyperparameters λ_P and λ_T (or their implicit ratio) control the relative importance of performance versus CoT reduction. AdaCoT focuses on providing mechanisms to explore this frontier.

2.2 Training Pipeline for AdaCoT

The AdaCoT training pipeline integrates supervised fine-tuning (SFT) as an initialization phase, followed by multi-stage reinforcement learning (RL) to refine the adaptive CoT triggering behavior.

2.2.1 Data Preparation and Supervised Fine-Tuning (SFT) as Warm-up

To provide the model with an initial understanding of when CoT might be beneficial, we perform a data preparation stage. This is achieved by leveraging an auxiliary model, guided by a set of predefined principles (e.g., query complexity, expected reasoning depth, domain; see Appendix B). In our implementation, we use an internal 15B-parameter model to generate these annotations; however, the framework is model-agnostic and can be instantiated using any sufficiently capable LLM with basic instruction-following abilities. Queries are labeled as either likely benefiting from CoT or likely suitable for a direct answer. This principled, automated labeling process is more consistent and scalable than manual annotation.

The SFT training data is then structured as follows: For queries labeled as benefiting from CoT, responses retain the full reasoning process: `<think>reasoning_steps</think>answer`. For queries labeled as not requiring CoT, responses omit explicit reasoning but maintain structural consistency: `<think></think>answer`. This SFT stage serves as a "warm-up", equipping the model with a foundational capability to distinguish between these two response styles. The consistent use of `<think></think>` tags is crucial for maintaining response format integrity.

2.2.2 Reinforcement Learning for Adaptive CoT Control

The RL stage is pivotal for fine-tuning AdaCoT’s adaptive reasoning capabilities. We design a reward function $R(x, r)$ for an input query x and generated response r :

$$R(x, r) = R_{\text{base}}(x, r) - \alpha_1 \cdot P_{\text{miss}}(x, r) - \alpha_2 \cdot P_{\text{over}}(x, r) - \gamma \cdot P_{\text{fmt}}(r) \quad (4)$$

where $R_{\text{base}}(x, r)$ is the base reward reflecting response quality, $P_{\text{miss}}(x, r)$ is a binary penalty for reasoning omission, $P_{\text{over}}(x, r)$ is a binary penalty for reasoning overuse, $P_{\text{fmt}}(r)$ is a binary penalty for format errors, and $\alpha_1, \alpha_2, \gamma$ are non-negative penalty coefficients. By adjusting α_1 and α_2 , we steer AdaCoT towards different CoT triggering decision boundaries, allowing exploration of the Pareto frontier.

2.2.3 Addressing Decision Boundary Collapse with Selective Loss Masking

A significant challenge in multi-stage RL, particularly when fine-tuning on specialized datasets with skewed CoT distributions (e.g., mathematical datasets where CoT is almost always beneficial), is the risk of the adaptive CoT triggering capability becoming unstable or collapsing. The model might revert to a homogeneous behavior, either always or never triggering CoT, thereby losing the nuanced decision-making learned in earlier, more balanced training stages. We term this phenomenon decision boundary collapse. This is particularly problematic if the final RL stage has significant bias, as it can lead to the model almost completely losing its adaptive triggering capability.

To address decision boundary collapse, AdaCoT incorporates Selective Loss Masking (SLM). SLM aims to preserve the CoT triggering ratio and distribution established during SFT or prior RL stages. It achieves this by selectively masking the loss contribution from the pivotal "decision token" during RL phases prone to distribution bias. This decision token is defined as the token immediately succeeding the `<think>` tag.

The modified policy gradient loss under SLM, \mathcal{L}_{SLM} , is computed by excluding the loss component associated with this decision token:

$$\mathcal{L}_{\text{SLM}} = \sum_{k \neq k_{\text{decision}}} \ell_k \quad (5)$$

where ℓ_k is the original loss component for the k -th token, and k_{decision} is the index of the decision token.

3 Experiments

We conducted extensive experiments to evaluate the AdaCoT framework, focusing on its ability to navigate the performance-cost trade-off, the effectiveness of its adaptive triggering mechanism, and its impact on inference efficiency. This section details our experimental setup, presents the main results, and analyzes the findings.

3.1 Experimental Setup

For our base model, we utilized our internal 15B/150B parameter Mixture-of-Experts (MoE) [28, 29] model. The AdaCoT post-training process comprised an initial SFT stage, followed by a two-stage RL procedure: first, a Mathematics-Focused RL stage (RL-Math) concentrated on complex, rule-verifiable problems, and second, a General Domain RL stage (RL-General) which incorporated broader data and a trained reward model. We compared our **AdaCoT RL Models** (Exp1-Exp4) against several baselines: a **Full CoT SFT Baseline** (SFT model always generating CoT), a **Full CoT RL Baseline** (RL model derived from the Full CoT SFT, always generating CoT), a **No CoT SFT Baseline** (SFT model never generating CoT), a **No CoT RL Baseline** (RL model derived from the No CoT SFT, never generating CoT), and our **AdaCoT SFT Model** (our model after only the SFT stage, also referred to as Adaptive SFT Model).

The SFT and RL training datasets were constructed to cover a diverse range of domains, including mathematics, reasoning, professional disciplines (e.g., law, medicine), dialogue, creative writing, and general knowledge question answering. Both SFT and RL data were labeled for CoT necessity using the principle-guided assessment detailed in Appendix B. In the SFT dataset, approximately 67% of the samples were labeled as requiring CoT, while in the RL dataset, this proportion was around 40%. During SFT, queries identified as not requiring CoT were formatted with empty `<think></think>` tags. In the RL-Math stage, which is particularly prone to decision boundary collapse, we employed Selective Loss Masking (SLM), as described in Section 2.2.3. For the RL-General stage, we applied penalties according to Equation 4, systematically varying the α_1 and α_2 coefficients to explore different points on the Pareto frontier. Proximal Policy Optimization (PPO) [30] was used for all RL policy updates.

For evaluation, we used 15 diverse open-source benchmark datasets to assess overall performance, measured by the average score. To balance internal iteration efficiency with evaluation accuracy, some of these datasets underwent up-sampling or down-sampling, or the number of inference runs per test sample was adjusted (with the final score being an average over multiple inferences). These datasets include LiveBench [31], MMLU Pro [32], SuperGPQA [33], GPQA [34], Chinese SimpleQA [35], SimpleQA [36], AIME24 & AIME25, MATH [37], OlympiadBench [38], SweBench Agentless [39], LiveCodeBench [40], KOR-Bench [41], ProcBench [42], and SysBench [43]. The detailed per-dataset scores, which form the basis for our average score calculations, are presented in Appendix A (Table 4). To specifically assess CoT triggering decisions on typical user queries, we curated a high-quality balanced test set of 1000 prompts. These prompts were labeled for CoT necessity using the same principle-guided assessment as our SFT/RL training data and subsequently underwent manual verification to ensure label accuracy. On this set, we report CoT Triggering Accuracy, F1-score, Precision, and Recall, where the positive class indicates that CoT is required. Other metrics included the CoT triggering rate on the benchmark datasets and the average response token num on production setting.

3.2 Results and Analysis

Our results demonstrate AdaCoT’s ability to effectively control CoT invocation, leading to improved efficiency while maintaining strong performance.

3.2.1 Pareto Frontier Analysis

We trained four variants of our AdaCoT RL model (Exp1-Exp4) by varying the penalty coefficients α_1 (for missing CoT) and α_2 (for overusing CoT). The specific coefficients were: Exp1 ($\alpha_1 = 0.1, \alpha_2 = 0.3$), Exp2 ($\alpha_1 = 0.2, \alpha_2 = 0.3$), Exp3 ($\alpha_1 = 0.3, \alpha_2 = 0.3$), and Exp4 ($\alpha_1 = 0.3, \alpha_2 = 0.1$). The format error penalty γ was consistently set to 1.0. Figure 1 illustrates the average score plotted against the CoT triggering rate for these models and the baselines, based on the average performance across our 15 benchmark datasets (detailed in Appendix Table 4). The No CoT SFT baseline achieved an average score of 43.6% with 0% CoT usage, while the No CoT RL baseline improved this to 47.7% at 0% CoT. The AdaCoT SFT Model (Adaptive SFT) registered a 57.1% average score at a 61.3% CoT rate.

The AdaCoT RL models trace a compelling Pareto frontier. AdaCoT RL Exp1 (43.1% CoT, 59.7% score) and AdaCoT RL Exp2 (53.3% CoT, 62.8% score) demonstrate significant performance gains over the AdaCoT SFT model while operating at lower or comparable CoT rates. Notably, AdaCoT RL Exp2 achieves a 62.8% average score, approaching the Full CoT RL baseline (65.0% score, 100% CoT) with nearly half the CoT usage. As we increase the CoT triggering rate, AdaCoT RL Exp3 (65.4% CoT, 64.3% score) and AdaCoT RL Exp4 (67.7% CoT, 64.4% score) further push performance, closely rivaling the Full CoT RL baseline’s score but with approximately 32-35% less CoT invocation. Moreover, these results indicate that despite the fixed CoT triggering labels within the SFT/RL data, adjusting the combination of penalty coefficients during the RL phase enables the final RL model to learn triggering strategies that transcend these initial labeling constraints.

This highlights AdaCoT’s effectiveness in navigating the trade-off between performance and CoT cost. However, it is also observable that while the AdaCoT RL models achieve substantial efficiency gains and define a superior Pareto curve compared to simpler baselines, they encounter a slight performance bottleneck when their triggering rates are pushed higher. Specifically, even the highest-performing adaptive models (Exp3 and Exp4, with scores of 64.3% and 64.4% respectively) do not surpass the average score of the Full CoT RL baseline (65.0%). This suggests that while AdaCoT excels at reducing CoT for a vast majority of queries without compromising much on average performance, and indeed offers a better score-to-cost ratio, the absolute peak average performance achieved by a model specialized to always use CoT (Full CoT RL) remains marginally higher. This indicates that the adaptive mechanism, while highly effective, might not perfectly replicate or exceed the performance of an always-on CoT strategy in every single scenario contributing to the average, thus not fully crossing this specific optimal boundary for maximum average score. This could be due to the inherent complexities of learning a universally optimal triggering heuristic or the RL optimization finding a balance that prioritizes the significant cost savings available across the query distribution.

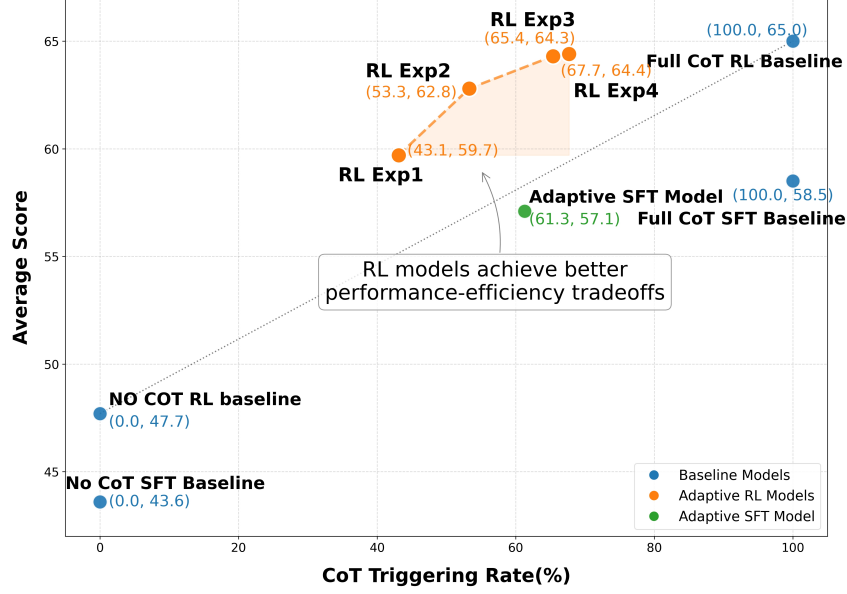


Figure 1: Average Score vs. CoT Triggering Rate across 15 widely-adopted benchmarks. Blue points represent baseline models. The green point is the AdaCoT SFT model. Orange points represent AdaCoT RL models trained with varying penalty coefficients, forming an improved Pareto frontier (indicated by the orange dashed line and shaded region) over the baselines. The dotted line connects the No CoT RL baselines to the Full CoT RL baseline, illustrating a simpler trade-off curve.

3.2.2 Adaptive CoT Triggering Performance and Ablation Studies on Daily-Use Queries

We evaluated the CoT triggering capabilities of AdaCoT at various training stages using our curated 1000-prompt daily-use test set. Table 1 presents these results, which include an ablation study for SLM and an assessment of the meta-reasoning strategy (discussed further in Section 4.2) at the SFT stage.

Table 1: CoT triggering performance on the 1000 daily-use prompt test set across different AdaCoT stages and configurations (positive class: requires CoT). RL-Math is the Mathematics-Focused RL stage; RL-General refers to the final models (Exp1-4).

Model Stage / Variant	Accuracy	F1-Score	Recall	Precision
AdaCoT SFT Model	0.795	0.750	0.616	0.959
AdaCoT RL Model (Exp1 - RL-General)	0.657	0.484	0.322	0.975
AdaCoT RL Model (Exp2 - RL-General)	0.816	0.814	0.804	0.823
AdaCoT RL Model (Exp3 - RL-General)	0.809	0.789	0.716	0.879
AdaCoT RL Model (Exp4 - RL-General)	0.678	0.535	0.370	0.963
RL-Math (without SLM)	0.506	0.669	1.000	0.503
RL-Math (with SLM)	0.813	0.781	0.670	0.938
AdaCoT SFT Model (with Meta-Reasoning)	0.858	0.840	0.762	0.935

The AdaCoT SFT model itself provides a strong baseline for adaptive triggering. The results clearly show that the RL-Math stage without SLM suffers from severe decision boundary collapse: the model defaults to triggering CoT (Recall=1.0) but with poor precision (0.503) and consequently low overall accuracy (0.506). Applying SLM during the RL-Math stage effectively preserves the adaptive capability learned during SFT, maintaining high precision (0.938) and achieving significantly better accuracy (0.813). The final AdaCoT RL models (Exp1-4, emerging from the RL-General stage) demonstrate how adjusting the RL penalty coefficients (α_1, α_2) allows for fine-tuning of the decision

boundary. AdaCoT RL Model Exp2, for example, achieves a well-balanced F1-score of 0.814. The incorporation of a meta-reasoning strategy at the SFT stage also shows a notable improvement in triggering performance, a point elaborated in Section 4.2.

3.2.3 Response Length Reduction and Efficiency Gains

The adaptive reasoning enabled by AdaCoT translates into significant reductions in computational costs. Table 2 shows the average response length and CoT triggering rates for AdaCoT RL Model Exp2 (selected for its balanced performance on the daily-use set and strong average benchmark performance) when applied to our production traffic testset, which reflects natural, unfiltered user query distributions.

Table 2: Average response token num (with reduction noted) and CoT triggering rate on production traffic testset for AdaCoT RL Model Exp2 vs. Full CoT RL Baseline.

Platform	Model / Mode	Avg. Response Tokens	CoT Triggering Rate
Mobile	Full CoT RL Baseline	377.18	100.00%
	AdaCoT RL Model Exp2 (Adaptive)	116.70 (↓69.1%)	3.18%
PC	Full CoT RL Baseline	1376.31	100.00%
	AdaCoT RL Model Exp2 (Adaptive)	405.25 (↓70.6%)	12.50%

As evidenced, AdaCoT RL Model Exp2 achieves very low CoT triggering rates in a production setting (3.18% on mobile devices, 12.50% on PCs). This dramatic reduction from the 100% CoT usage of a non-adaptive model translates directly into substantial computational savings.

4 Discussion and Future Work

4.1 Design Considerations and Limitations

AdaCoT offers a pragmatic approach to adaptive reasoning by combining principle-guided initial data labeling with RL-based optimization of the CoT decision boundary. This methodology was chosen to circumvent inherent challenges in purely autonomous CoT trigger learning, such as information asymmetry in assessing counterfactual benefits and the difficulty of quantifying quality degradation from CoT omission, particularly for subjective tasks.

While AdaCoT is a promising initial step, several limitations exist. The optimal CoT triggering strategy is relative to the base model’s capabilities, necessitating recalibration for different LLMs. Our current binary CoT invocation (on/off) simplifies a continuous spectrum of reasoning depths and styles, potentially limiting nuance. Domain generalization remains a challenge, as CoT necessity can vary significantly across knowledge areas, and the framework currently lacks personalization for user verbosity preferences. Moreover, the initial principle-guided labeling requires continuous refinement. Our Pareto analysis (Section 3.2.1) also indicates that while AdaCoT significantly improves efficiency and nears the peak average performance of specialized always-on CoT models, a small performance gap persists, highlighting the difficulty for adaptive mechanisms to achieve absolute maximum performance across all query types.

Acknowledging the limitations of the current framework, we anticipate that future research will offer valuable critiques and further refine these initial explorations. Areas warranting deeper investigation include more granular control over reasoning, such as adaptive reasoning length where models dynamically adjust verbosity, or more nuanced triggering mechanisms beyond a simple binary decision. We believe such continued efforts by the community are crucial for developing more sophisticated and efficient reasoning strategies, potentially addressing the observed performance gap while maximizing efficiency and enhancing nuanced control over LLM reasoning.

4.2 Meta-Reasoning for Reasoning Decisions

We explored an innovative "meta-reasoning" approach during the SFT stage to enhance AdaCoT’s CoT triggering decisions. This involved the model first generating a brief internal assessment of the input query’s complexity before deciding whether to proceed with full CoT, as illustrated by

the response structures in Figure 2. Incorporating this strategy into the AdaCoT SFT model led to a notable improvement in CoT triggering performance on our daily-use test set: the F1-score increased from 0.750 to 0.840 (Table 1). This result suggests that explicitly prompting the model to first assess query complexity can significantly enhance its subsequent decision-making regarding CoT invocation.

Response Format with Meta-Reasoning
Straightforward Query <code><think>This is a straightforward question about X, I think I can answer directly.</code> <code>[empty]</think></code> <code>{answer}</code>
Complex Query <code><think>This is a relatively complex question about Y, I need to think carefully.</code> <code>[formal reasoning process]</think></code> <code>{answer}</code>

Figure 2: Example response structure incorporating explicit meta-reasoning for CoT decisions.

An interesting and serendipitous discovery with the meta-reasoning SFT model was an emergent capability for user-prompt controllability over CoT. Users could, to some extent, influence whether the model engaged in CoT by including explicit cues in their prompts about the desired level of reasoning (e.g., "please think step-by-step" to encourage CoT, or "give a direct answer" to discourage it). While this controllability was not perfectly accurate across all scenarios, it points towards a promising avenue for developing more interactive and user-guided reasoning systems (further illustrative examples are provided in Appendix D).

Despite these benefits, the explicit meta-reasoning step inherently increases the number of tokens generated for every query, as the model first articulates its complexity assessment. Considering the very low CoT triggering rates observed for AdaCoT in production environments (e.g., 3.18% on mobile traffic for AdaCoT RL Model Exp2, as shown in Table 2), the cumulative token cost of these additional meta-reasoning steps would become substantial. Therefore, while acknowledging its potential for improving decision accuracy and enabling user control, we did not adopt this explicit meta-reasoning as the default for subsequent RL experiments due to this efficiency trade-off. Nevertheless, it highlights an important direction for future research, which might explore more token-efficient methods for incorporating such meta-reasoning, perhaps through implicit learning mechanisms or parallel processing of the complexity assessment.

4.3 Preserving Peak Performance with AdaCoT

A critical question is whether adaptive reasoning limits a model’s maximum performance. We investigated this using System Prompts (SPs), integrated into AdaCoT’s SFT and RL training to control reasoning behavior (e.g., "Always Reason SP," "Never Reason SP"). During SFT, a small portion of data was augmented with SPs, and target responses were modified for compliance. In RL, a fraction of training prompts included SPs, with rewards penalizing deviations from explicit SP instructions, ensuring robust adherence (details in Appendix C). Our focus here is using the "Always Reason SP" to assess AdaCoT’s performance ceiling.

Instructing AdaCoT RL models to always generate CoT via this SP allowed direct comparison against the Full CoT RL Baseline on our 15 benchmark datasets. As shown in Table 3, AdaCoT RL models in this forced "Always Reason" mode achieved average scores that were highly competitive with, and in instances like AdaCoT RL Model Exp4 (65.7) and Exp2 (65.3), slightly surpassed the Full CoT RL Baseline (65.0). This demonstrates a key strength: AdaCoT’s adaptive training, aimed at optimizing the performance-cost Pareto frontier, does not inherently restrict the model’s peak reasoning capabilities when comprehensive reasoning is explicitly demanded. While our Pareto analysis (Section 3.2.1) noted a slight gap in average scores when models operate adaptively, these SP-controlled results affirm that AdaCoT offers efficiency without sacrificing potential high-end performance.

An interesting secondary observation arose from the SFT stage concerning long-form generation (up to 32,000 tokens). AdaCoT SFT models, when directed by an "Always Reason SP," exhibited fewer

Table 3: Performance of AdaCoT RL models under "Always Reason" System Prompt vs. Full CoT RL Baseline, demonstrating preservation of peak performance. Metrics are averaged across the 15 benchmark datasets.

Model Variant	Avg. Score	CoT Triggering Rate
Full CoT RL Baseline	65.0	100%
AdaCoT RL Model Exp1 (w/ Always CoT SP)	64.8	100%
AdaCoT RL Model Exp2 (w/ Always CoT SP)	65.3	100%
AdaCoT RL Model Exp3 (w/ Always CoT SP)	64.9	100%
AdaCoT RL Model Exp4 (w/ Always CoT SP)	65.7	100%

instances of premature output truncation and were less prone to undesirable generative loops compared to a standard Full CoT SFT baseline. We hypothesize this improvement stems from AdaCoT’s diverse SFT data, which includes many non-CoT examples (formatted as `<think></think>answer`). This results in a shorter average training sample length, potentially providing the End-of-Sequence (EOS) token a stronger learning signal (average EOS proportion: 0.000239 for AdaCoT SFT vs. 0.000215 for Full CoT SFT). A more robust EOS representation could foster more coherent, well-terminated lengthy outputs, a promising area for future investigation.

5 Related Work

Chain-of-Thought (CoT) reasoning [2] significantly advanced LLM capabilities by prompting step-by-step thought processes, inspiring sophisticated strategies like diverse path sampling [44] or structured thoughts (trees [45], graphs [46]). However, CoT’s verbosity and cost [47, 7] are major drawbacks. The varying utility of CoT—direct answers being better for some queries [48] while complex tasks need longer reasoning [49, 50]—underscores a critical cost-effectiveness trade-off, motivating research into efficient reasoning.

Most existing work on CoT efficiency has focused on *reducing reasoning length*, rather than adaptively deciding *whether* to invoke CoT. These length reduction strategies include: (1) RL with length penalties or rewards for brevity [8, 9, 10, 11, 12, 13]; (2) Restructuring or compressing CoT content via learned compact representations, iterative summarization, cognitive paradigms, or dynamic termination [14, 15, 16, 17, 18, 19, 20, 21]; (3) Employing explicit instructions or selection mechanisms for conciseness, such as dynamic token allocation, prompts for shorter responses, distillation, or selecting the shortest valid reasoning [22, 23, 24, 25, 26, 27].

While effective for length, these methods generally do not equip a single model to dynamically decide CoT invocation based on query nature. Alternatives involve model merging [51, 52, 53] or manual CoT toggling. AdaCoT distinctively addresses adaptive triggering. By framing it as a Pareto optimization problem and using RL to control the CoT decision boundary, AdaCoT enables nuanced, context-dependent CoT invocation by a single model, filling a crucial gap towards truly efficient and versatile LLMs.

6 Conclusion

In this paper, we introduced AdaCoT, a novel framework for adaptive Chain-of-Thought reasoning in LLMs. By formulating adaptive reasoning as a Pareto optimization problem and employing an RL-based method with adjustable penalty coefficients, AdaCoT dynamically controls CoT triggering based on implicit query complexity. Our experiments demonstrate AdaCoT’s success in navigating the Pareto frontier, achieving substantial reductions in CoT usage—and thereby computational costs and latency—for simpler queries, while preserving high performance on complex reasoning tasks. The introduction of Selective Loss Masking effectively ensures robust adaptive triggering throughout multi-stage RL training. Distinguishing itself from prior work predominantly focused on CoT length compression, AdaCoT pioneers adaptive CoT triggering, offering a practical and principled solution for developing more efficient, responsive, and cost-effective LLMs, particularly crucial for interactive applications.

References

- [1] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in neural information processing systems*, volume 35, pages 27730–27744, 2022.
- [2] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in neural information processing systems*, volume 35, pages 24824–24837, 2022.
- [3] Fengli Xu, Qianye Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*, 2025.
- [4] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- [5] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [6] ByteDance Seed, Yufeng Yuan, Yu Yue, Mingxuan Wang, Xiaochen Zuo, Jiaze Chen, Lin Yan, Wenyan Xu, Chi Zhang, Xin Liu, et al. Seed-thinking-v1. 5: Advancing superb reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.13914*, 2025.
- [7] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for $2+3=?$ on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.
- [8] Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.
- [9] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.
- [10] Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.
- [11] Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.
- [12] Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *arXiv preprint arXiv:2503.04472*, 2025.
- [13] Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.
- [14] Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. Lightthinker: Thinking step-by-step compression. *arXiv preprint arXiv:2502.15589*, 2025.
- [15] Yuchen Yan, Yongliang Shen, Yang Liu, Jin Jiang, Mengdi Zhang, Jian Shao, and Yueting Zhuang. Infythink: Breaking the length limits of long-context reasoning in large language models. *arXiv preprint arXiv:2503.06692*, 2025.
- [16] Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. *arXiv preprint arXiv:2503.05179*, 2025.

- [17] Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*, 2025.
- [18] Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025.
- [19] Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320, 2025.
- [20] Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*, 2025.
- [21] Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. Can language models learn to skip steps? In *Advances in Neural Information Processing Systems*, volume 37, pages 45359–45385, 2024.
- [22] Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*, 2024.
- [23] Matthew Renze and Erhan Guven. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 476–483. IEEE, 2024.
- [24] Qiguang Chen, Libo Qin, Jiaqi Wang, Jingxuan Zhou, and Wanxiang Che. Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought. In *Advances in Neural Information Processing Systems*, volume 37, pages 54872–54904, 2024.
- [25] Ping Yu, Jing Xu, Jason E Weston, and Ilia Kulikov. Distilling system 2 into system 1. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS’24*, 2024.
- [26] Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*, 2025.
- [27] Jiabao Pan, Yan Zhang, Chen Zhang, Zuozhu Liu, Hongwei Wang, and Haizhou Li. Dynathink: Fast or slow? a dynamic decision-making framework for large language models. *arXiv preprint arXiv:2407.01009*, 2024.
- [28] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- [29] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.
- [30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [31] Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddhartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. Livebench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [32] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhramil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pages 95266–95290, 2025.
- [33] Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

- [34] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.
- [35] Yancheng He, Shilong Li, Jiaheng Liu, Yingshui Tan, Weixun Wang, Hui Huang, Xingyuan Bu, Hangyu Guo, Chengwei Hu, Boren Zheng, et al. Chinese simpleqa: A chinese factuality evaluation for large language models. *arXiv preprint arXiv:2411.07140*, 2024.
- [36] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.
- [37] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *NeurIPS*, 2021.
- [38] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.
- [39] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024.
- [40] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [41] Kaijing Ma, Xinrun Du, Yunran Wang, Haoran Zhang, Zhoufutu Wen, Xingwei Qu, Jian Yang, Jiaheng Liu, Minghao Liu, Xiang Yue, et al. Kor-bench: Benchmarking language models on knowledge-orthogonal reasoning tasks. *arXiv preprint arXiv:2410.06526*, 2024.
- [42] Ippei Fujisawa, Sensho Nobe, Hiroki Seto, Rina Onda, Yoshiaki Uchida, Hiroki Ikoma, Pei-Chun Chien, and Ryota Kanai. Procbench: Benchmark for multi-step reasoning and following procedure. *arXiv preprint arXiv:2410.03117*, 2024.
- [43] Yanzhao Qin, Tao Zhang, Yanjun Shen, Wenjing Luo, Haoze Sun, Yan Zhang, Yujing Qiao, Weipeng Chen, Zenan Zhou, Wentao Zhang, et al. Sysbench: Can large language models follow system messages? *arXiv preprint arXiv:2408.10943*, 2024.
- [44] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.
- [45] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in neural information processing systems*, volume 36, pages 11809–11822, 2023.
- [46] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.
- [47] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.
- [48] Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*, 2025.

- 510 [49] Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less:
511 Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.
- 512 [50] Ayeong Lee, Ethan Che, and Tianyi Peng. How well do llms compress their own chain-of-
513 thought? a token complexity approach. *arXiv preprint arXiv:2503.01141*, 2025.
- 514 [51] Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve:
515 Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*, 2025.
- 516 [52] Han Wu, Yuxuan Yao, Shuqi Liu, Zehua Liu, Xiaojin Fu, Xiongwei Han, Xing Li, Hui-Ling
517 Zhen, Tao Zhong, and Mingxuan Yuan. Unlocking efficient long-to-short llm reasoning with
518 model merging. *arXiv preprint arXiv:2503.20641*, 2025.
- 519 [53] Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao,
520 Dacheng Tao, and Li Shen. Adar1: From long-cot to hybrid-cot via bi-level adaptive reasoning
521 optimization. *arXiv preprint arXiv:2504.21659*, 2025.

A Benchmark Dataset Details and Analysis

This appendix provides descriptions for the benchmark datasets used in our evaluation and an analysis of the experimental results on these individual datasets. The scores presented in Table 4 form the basis for this analysis. For each dataset, we discuss the performance of baseline models and the AdaCoT variants. We also highlight any counter-intuitive results or observations that conflict with the primary motivation of achieving optimal performance with adaptive CoT, offering potential explanations such as evaluation volatility, inherent limitations of the base model, or aspects of the post-training process that may not be fully optimized for every scenario.

Table 4: Detailed scores on benchmark datasets. "TR" denotes reasoning triggering rate (%).

Dataset	nocot SFT baseline TR Score	nocot RL baseline TR Score	fullcot SFT baseline TR Score	fullcot RL baseline TR Score	Adaptive SFT Model TR Score	Adaptive RL Model Exp1 TR Score	Adaptive RL Model Exp2 TR Score	Adaptive RL Model Exp3 TR Score	Adaptive RL Model Exp4 TR Score
MMLU pro	0.0 77.5	0.0 82.1	100.0 83.7	100.0 85.2	40.3 80.5	28.0 74.2	27.3 83.2	39.6 84.1	58.0 83.1
super GPQA	0.0 49.8	0.0 50.7	100.0 53.8	100.0 58.6	35.2 51.0	22.8 55.1	32.3 56.9	40.8 56.2	59.6 59.6
LiveBench	0.0 50.0	0.0 56.6	100.0 57.7	100.0 69.5	65.8 58.9	45.1 64.7	57.1 66.3	71.6 68.4	70.4 69.2
KORBENCH	0.0 33.9	0.0 42.1	100.0 61.3	100.0 62.8	52.5 49.1	28.0 53.5	45.1 52.1	62.2 57.4	61.0 59.2
AIME24	0.0 23.3	0.0 33.3	100.0 69.3	100.0 84.7	100.0 69.3	100.0 86.7	100.0 86.0	100.0 88.0	100.0 86.3
AIME25	0.0 13.3	0.0 21.0	100.0 52.3	100.0 70.0	100.0 56.7	100.0 73.3	100.0 75.7	100.0 74.0	100.0 72.0
MATH	0.0 84.3	0.0 88.7	100.0 96.5	100.0 97.3	44.6 95.5	40.9 91.7	52.8 95.9	61.5 95.5	68.8 97.2
LiveCodeBench	0.0 29.4	0.0 27.6	100.0 45.9	100.0 55.9	95.0 47.0	77.1 44.8	83.9 50.6	91.4 54.1	91.4 55.9
SWE-bench Agentless	0.0 27.4	0.0 27.0	100.0 36.4	100.0 37.7	98.6 35.4	1.6 28.8	44.0 35.8	94.2 37.6	79.6 36.6
Chinese SimpleQA	0.0 58.8	0.0 57.0	100.0 59.7	100.0 61.5	0.3 59.1	0.2 56.0	0.4 55.0	1.4 56.8	6.0 55.2
SimpleQA	0.0 10.8	0.0 10.3	100.0 12.7	100.0 12.2	0.7 10.3	0.0 9.9	0.0 10.7	4.4 11.3	20.0 9.6
Proc-Bench	0.0 42.6	0.0 46.7	100.0 53.7	100.0 68.6	73.6 50.2	50.8 53.0	79.1 68.7	89.3 69.6	93.1 72.4
GPQA(diamond)	0.0 59.5	0.0 64.3	100.0 64.9	100.0 70.5	92.6 65.7	62.6 65.7	84.1 67.2	92.4 70.8	96.5 72.6
SysBench	0.0 42.6	0.0 48.1	100.0 56.2	100.0 62.4	35.6 52.0	1.2 57.2	4.0 55.5	38.2 60.2	15.4 55.7
Olympiad Bench	0.0 51.4	0.0 60.1	100.0 73.5	100.0 78.2	85.0 75.3	88.0 80.2	89.0 81.8	93.5 80.7	95.4 82.0
Average	0.0 43.6	0.0 47.7	100.0 58.5	100.0 65.0	61.3 57.1	43.1 59.7	53.3 62.8	65.4 64.3	67.7 64.4

MMLU pro: An enhanced version of the MMLU benchmark, MMLU-Pro integrates more challenging, reasoning-focused questions, expands the choice set from four to ten options, and eliminates trivial/noisy questions from the original MMLU. It is designed to better discern model capabilities, particularly in complex reasoning, where CoT has shown greater benefit compared to direct answering on this version. *Analysis:* CoT clearly benefits performance, with the FullCoT RL baseline (85.2) significantly outperforming NoCoT SFT (77.5) and NoCoT RL (82.1). The Adaptive SFT Model (40.3% TR, 80.5 score) shows improvement over NoCoT SFT but doesn't reach FullCoT SFT levels (83.7). AdaCoT RL Exp3 (39.6% TR, 84.1 score) and Exp2 (27.3% TR, 83.2 score) achieve strong scores, with Exp3 surpassing FullCoT SFT and Exp2 performing comparably. AdaCoT RL Exp4 (58.0% TR, 83.1 score) also performs well. Exp1 (28.0% TR, 74.2 score) shows a drop, indicating that for MMLU pro, a moderate CoT rate is generally beneficial, reflecting the benchmark's increased reasoning demands. The adaptive models demonstrate an ability to achieve high scores with significantly reduced CoT compared to FullCoT RL.

super GPQA: A comprehensive benchmark evaluating graduate-level knowledge and reasoning capabilities across 285 disciplines, particularly including specialized fields in light industry, agriculture, and service-oriented areas often underrepresented in other benchmarks. It employs a Human-LLM collaborative filtering mechanism to ensure high question quality by eliminating trivial or ambiguous questions. *Analysis:* CoT provides a clear advantage (NoCoT RL 50.7 vs. FullCoT RL 58.6). The Adaptive SFT Model (35.2% TR, 51.0 score) also show a modest gain over NoCoT SFT. AdaCoT RL Exp4 (59.6% TR) notably achieves a score of 59.6, surpassing the FullCoT RL baseline with significantly less CoT. AdaCoT RL Exp2 (32.3% TR, 56.9 score) also outperforms FullCoT SFT (53.8). This suggests AdaCoT effectively adapts CoT usage for these specialized, high-level questions, achieving strong performance efficiently.

LiveBench: A benchmark designed to be resistant to test set contamination and the pitfalls of LLM/human-crowdsourced judging. It features frequently updated questions from recent sources (math competitions, arXiv papers, news, datasets), scores answers automatically against objective ground-truth, and includes a wide variety of challenging tasks (math, coding, reasoning, language, instruction following, data analysis), including harder, contamination-limited versions of tasks from previous benchmarks. *Analysis:* This dataset shows significant gains from both CoT and RL (NoCoT SFT 50.0 to FullCoT RL 69.5). The NoCoT RL baseline (56.6) and Adaptive SFT Model (65.8% TR, 58.9 score) both outperform NoCoT SFT, with Adaptive SFT also surpassing FullCoT SFT (57.7). AdaCoT RL Exp4 (70.4% TR, 69.2 score) very closely approaches the FullCoT RL baseline performance with about 30% less CoT. AdaCoT RL Exp3 (71.6% TR, 68.4 score) is also strong. AdaCoT RL Exp2 (57.1% TR, 66.3 score) substantially outperforms FullCoT SFT. The robust design of LiveBench makes it a strong test case for AdaCoT's adaptive reasoning, showing it can maintain high performance with adaptive CoT.

KORBENCH: This benchmark evaluates Knowledge-Orthogonal-Reasoning, aiming to minimize reliance on domain-specific knowledge to more accurately assess models’ reasoning abilities in out-of-distribution settings. It includes five task categories (Operation, Logic, Cipher, Puzzle, Counterfactual) and emphasizes models’ effectiveness in applying new rule descriptions to solve novel rule-driven questions. *Analysis:* Scores show a clear benefit from CoT: NoCoT SFT (33.9) is significantly lower than FullCoT SFT (61.3) and FullCoT RL (62.8). The NoCoT RL baseline (42.1) improves over NoCoT SFT. The Adaptive SFT Model (52.5% TR, 49.1 score) sits between the NoCoT baselines and FullCoT SFT. AdaCoT RL models demonstrate adaptive behavior: Exp4 (61.0% TR, 59.2 score) and Exp3 (62.2% TR, 57.4 score) approach the FullCoT SFT baseline performance with significantly less CoT than FullCoT models. Exp1 (28.0% TR, 53.5 score) is also effective. This suggests AdaCoT effectively discerns when to apply CoT for these rule-driven tasks, though peak performance is slightly below FullCoT RL.

AIME24 / AIME25: Representing problems from the American Mathematics Invitational Examination for 2024 and 2025, these datasets are used to evaluate mathematical reasoning and problem-solving abilities. *Analysis:* These mathematics-intensive datasets show massive performance gains from CoT (e.g., AIME24: NoCoT SFT 23.3 vs. FullCoT RL 84.7). All AdaCoT RL models and the Adaptive SFT Model correctly identify the complexity, exhibiting a 100% CoT triggering rate. For AIME24, Adaptive SFT (69.3) matches FullCoT SFT (69.3). AdaCoT RL Exp3 (88.0) and Exp1 (86.7) outperform the FullCoT RL baseline (84.7). For AIME25, Adaptive SFT (56.7) surpasses FullCoT SFT (52.3). AdaCoT RL Exp2 (75.7) and Exp3 (74.0) outperform the FullCoT RL baseline (70.0). This is a notable result, suggesting that the adaptive training regimen, even when defaulting to 100% CoT for such complex problems, might confer some benefits, potentially due to the diversity in training data (including non-CoT examples) leading to a more robust underlying model or better fine-tuning dynamics.

MATH: A dataset of 12,500 challenging competition mathematics problems, each with a full step-by-step solution. It is designed to measure mathematical problem-solving ability. *Analysis:* CoT is highly beneficial (NoCoT RL 88.7 vs. FullCoT RL 97.3). The Adaptive SFT Model (44.6% TR, 95.5 score) performs well, nearly matching FullCoT SFT (96.5) with less than half the CoT. AdaCoT models adapt effectively: AdaCoT RL Exp4 (68.8% TR, 97.2 score) nearly matches the FullCoT RL baseline with about 30% less CoT. AdaCoT RL Exp2 (52.8% TR, 95.9 score) also performs strongly. Exp1 (40.9% TR, 91.7 score) is lower, indicating that for MATH, higher CoT rates are generally more beneficial among the adaptive RL models, but significant efficiency is still gained.

LiveCodeBench: A comprehensive and contamination-free evaluation benchmark for LLMs on code, collecting new problems over time from programming contests. It assesses a broader range of code-related capabilities. *Analysis:* CoT improves performance significantly (NoCoT SFT 29.4 vs. FullCoT RL 55.9). NoCoT RL (27.6) is surprisingly lower than NoCoT SFT here, which might be due to evaluation noise or specific sensitivities of the RL fine-tuning on non-CoT data for this particular task. The Adaptive SFT Model (95.0% TR, 47.0 score) uses a high trigger rate and surpasses FullCoT SFT (45.9). AdaCoT RL models trigger CoT at high rates: Exp4 (91.4% TR, 55.9 score) matches the FullCoT RL baseline score with slightly less CoT. Exp3 (91.4% TR, 54.1 score) and Exp2 (83.9% TR, 50.6 score) are also strong. This indicates recognition of coding task complexity and efficient application of CoT.

SWE-bench Agentless: An evaluation framework consisting of 2,294 software engineering problems from real GitHub issues. Models are tasked with editing codebases to resolve issues. *Analysis:* CoT provides a notable benefit (NoCoT SFT 27.4 vs. FullCoT RL 37.7). NoCoT RL (27.0) is similar to NoCoT SFT. The Adaptive SFT Model (98.6% TR, 35.4 score) uses a very high trigger rate and performs close to FullCoT SFT (36.4). AdaCoT RL Exp3 (94.2% TR, 37.6 score) nearly matches the FullCoT RL baseline with slightly less CoT. Interestingly, AdaCoT RL Exp1 (1.6% TR, 28.8 score) shows a slight improvement over NoCoT SFT with minimal reasoning. This suggests some issues might be simpler, or the model is highly conservative in Exp1, but for complex software issues, high CoT rates are beneficial. The performance of Exp2 (44.0% TR, 35.8 score) is also noteworthy, achieving good results with moderate CoT.

Chinese SimpleQA: The first comprehensive Chinese benchmark to evaluate the factuality of language models in answering short questions. *Analysis:* CoT offers minimal gains (NoCoT SFT 58.8 to FullCoT RL 61.5). NoCoT RL (57.0) is slightly lower than NoCoT SFT. The Adaptive SFT Model (0.3% TR, 59.1 score) performs very well, slightly exceeding NoCoT SFT and approaching FullCoT

SFT (59.7) with extremely low CoT usage. AdaCoT RL models trigger CoT very infrequently (0.2% to 6.0%), correctly identifying these as simple questions. Scores for AdaCoT RL models (e.g., Exp1 56.0, Exp3 56.8) are slightly below NoCoT SFT. This is a good demonstration of AdaCoT’s core motivation: avoiding unnecessary CoT. While there’s a slight dip compared to NoCoT SFT for some RL models, the Adaptive SFT model shows an excellent trade-off. The minor performance variations could be due to the model sometimes being overly conservative in triggering CoT or slight instabilities in evaluating purely factual recall without reasoning.

SimpleQA: A benchmark designed to measure the factuality of language models using short, fact-seeking queries. *Analysis:* Similar to Chinese SimpleQA, CoT provides little benefit; FullCoT RL (12.2) is slightly worse than FullCoT SFT (12.7). NoCoT RL (10.3) is slightly below NoCoT SFT (10.8). The Adaptive SFT Model (0.7% TR, 10.3 score) matches NoCoT RL with minimal CoT. AdaCoT RL models trigger CoT very rarely (Exp1 & Exp2 at 0.0%). AdaCoT RL Exp3 (4.4% TR, 11.3 score) performs better than NoCoT SFT. This reinforces that for simple QA, adaptive triggering is crucial for efficiency. The performance of Exp4 (20.0% TR, 9.6 score) is slightly counter-intuitive, as higher CoT did not yield better results and was worse than NoCoT SFT; this might indicate that for very simple questions, forcing CoT (even adaptively at a higher rate) can sometimes be detrimental or that the specific penalty balance for Exp4 was not optimal for this type of dataset.

ProcBench: This benchmark focuses on the direct evaluation of multi-step inference by largely eliminating path exploration and implicit knowledge utilization. *Analysis:* CoT is highly beneficial (NoCoT RL 46.7 vs. FullCoT RL 68.6). The Adaptive SFT Model (73.6% TR, 50.2 score) improves over NoCoT baselines but is below FullCoT SFT (53.7). AdaCoT RL models show high trigger rates, with Exp4 (93.1% TR, 72.4 score) significantly surpassing the FullCoT RL baseline. Exp2 (79.1% TR, 68.7 score) and Exp3 (89.3% TR, 69.6 score) also match or exceed FullCoT RL. This indicates effective identification of tasks requiring detailed, step-by-step procedural reasoning and demonstrates that adaptive models can even outperform always-on CoT models in certain complex reasoning scenarios.

GPQA (diamond): GPQA is a challenging dataset of 448 multiple-choice questions by domain experts in biology, physics, and chemistry. "GPQA (diamond)" refers to this specific challenging set. *Analysis:* CoT significantly boosts performance (NoCoT RL 64.3 vs. FullCoT RL 70.5). NoCoT RL is better than NoCoT SFT and close to FullCoT SFT (64.9). The Adaptive SFT Model (92.6% TR, 65.7 score) also performs well, exceeding FullCoT SFT. AdaCoT RL models trigger CoT at high rates. AdaCoT RL Exp4 (96.5% TR, 72.6 score) and Exp3 (92.4% TR, 70.8 score) demonstrate strong performance, with Exp4 outperforming the FullCoT RL baseline. This highlights appropriate and effective CoT invocation on these very hard questions.

SysBench: A benchmark for evaluating language models on their ability to understand and generate content related to computer systems, configurations, or system-level concepts. *Analysis:* CoT is beneficial (NoCoT RL 48.1 vs. FullCoT RL 62.4). The Adaptive SFT Model (35.6% TR, 52.0 score) is better than NoCoT RL but below FullCoT SFT (56.2). AdaCoT RL models show good adaptation. AdaCoT RL Exp3 (38.2% TR, 60.2 score) approaches the FullCoT RL baseline with significantly reduced CoT. AdaCoT RL Exp1 (1.2% TR, 57.2 score) surpasses FullCoT SFT with very minimal CoT usage, which is an excellent result for efficiency. This suggests a mix of complexities within SysBench, which AdaCoT navigates effectively, although Exp4 (15.4% TR, 55.7 score) shows a somewhat lower score despite a higher CoT rate than Exp1, possibly due to the specific balance of penalties in Exp4 not being optimal for this dataset’s particular mix.

OlympiadBench: An Olympiad-level bilingual multimodal scientific benchmark with problems from mathematics and physics competitions. *Analysis:* CoT provides substantial gains (NoCoT RL 60.1 vs. FullCoT RL 78.2). The Adaptive SFT Model (85.0% TR, 75.3 score) performs strongly, exceeding FullCoT SFT (73.5). AdaCoT RL models exhibit high trigger rates. AdaCoT RL Exp4 (95.4% TR, 82.0 score) and Exp2 (89.0% TR, 81.8 score) both surpass the FullCoT RL baseline. This indicates strong reasoning capabilities and appropriate CoT usage on these exceptionally challenging problems, again showing adaptive models can reach or exceed the performance of specialized always-on CoT models.

Overall Summary of Per-Dataset Analysis: The adaptive models, including both Adaptive SFT and the AdaCoT RL variants, demonstrate effective adaptation across a diverse range of benchmarks.

- The Adaptive SFT Model serves as a strong adaptive baseline. It often improves significantly over NoCoT baselines by selectively triggering CoT (e.g., high rates for AIME, LiveCodeBench, SWE-bench; low rates for SimpleQAs). On some complex tasks (AIME25, LiveCodeBench, GPQA(diamond), OlympiadBench), it even surpasses the FullCoT SFT baseline, and for Chinese SimpleQA, it achieves excellent efficiency and performance. However, its average performance (57.1 score, 61.3% TR) is generally below the peak performance of FullCoT RL (65.0 score) or the best AdaCoT RL experiments (e.g., Exp4: 64.4 score, 67.7% TR).
- On complex reasoning tasks (e.g., AIME, MATH, OlympiadBench, GPQA(diamond), ProcBench), AdaCoT RL models tend to trigger CoT at high rates. Several AdaCoT RL experiments (notably Exp4 on super GPQA, ProcBench, GPQA(diamond), OlympiadBench; Exp3 on AIME24; Exp2 on AIME25, OlympiadBench) match or exceed the performance of FullCoT RL baselines, showcasing the benefits of learned adaptive policies and suggesting that adaptive training can sometimes lead to better overall models even for tasks that always require CoT.
- On simpler tasks or those designed to test factuality (e.g., Chinese SimpleQA, SimpleQA), AdaCoT RL models trigger CoT very sparingly. This leads to computational savings while generally maintaining performance near NoCoT or FullCoT SFT levels, successfully avoiding unnecessary CoT. The Adaptive SFT model also excels in efficiency here. Some minor performance drops in RL models compared to NoCoT SFT on these tasks (e.g., Chinese SimpleQA for Exp1/Exp2) might be attributed to the RL agent being slightly too conservative or the inherent difficulty in perfectly balancing penalties for extremely low CoT rate scenarios without any performance degradation.
- For benchmarks with mixed or specific reasoning types (e.g., KORBENCH, LiveBench, MMLU-Pro, SysBench), both Adaptive SFT and AdaCoT RL models show nuanced adaptation, adjusting CoT rates to balance performance and efficiency. They often outperform static baselines or achieve comparable results with lower CoT usage. For instance, on SysBench, AdaCoT RL Exp1 achieved a higher score than FullCoT SFT with only 1.2% TR.
- The different AdaCoT RL experiments (Exp1-Exp4) effectively trace a Pareto frontier (as shown in Section 3.2.1 using a specific set of average scores), offering a trade-off between CoT triggering rate and performance, adaptable to specific deployment needs. Based on the average scores from Appendix 4, Exp4 (64.4 score, 67.7% TR) and Exp3 (64.3 score, 65.4% TR) represent high-performance points, closely approaching the FullCoT RL baseline (65.0 score) with about 30-35% less CoT usage on average. Exp2 (62.8 score, 53.3% TR) also offers a strong balance.
- Some counter-intuitive results, like NoCoT RL performing worse than NoCoT SFT on LiveCodeBench, or AdaCoT RL Exp4 on SimpleQA (higher CoT, lower score), could be due to factors like evaluation volatility on specific datasets, the base model’s inherent capabilities or sensitivities to fine-tuning on certain data distributions, or sub-optimal penalty configurations for specific outlier datasets within a broadly tuned RL policy. The post-training process aims for general improvement, and individual dataset performance can fluctuate.

These detailed results underscore the ability of adaptive strategies, both SFT-based and RL-based, to make nuanced decisions about CoT invocation, optimizing for both performance and efficiency based on query characteristics and benchmark demands. The AdaCoT RL models, in particular, demonstrate the potential to significantly reduce CoT overhead while maintaining competitive, and in some cases superior, performance compared to full CoT strategies.

B Principle-Guided CoT Assessment

This appendix details the principle-guided assessment framework used to annotate data for Chain-of-Thought (CoT) necessity. As described in Section 2.2.1 of the main paper, an auxiliary model utilizes these principles to label queries as either likely benefiting from CoT or suitable for a direct answer. This labeling is crucial for the Supervised Fine-Tuning (SFT) warm-up stage of the AdaCoT

framework, providing an initial understanding for the model on when to employ CoT. The specific principles provided to the auxiliary model are outlined below.

```
Given a dialogue between a user and an AI assistant, please
    consider the conversation context and, from the AI
    assistant's perspective, assess the difficulty of
    answering the user's final question according to the
    following requirements.
<AI assistant's system prompt-Start>
{system_prompt}
<AI assistant's system prompt-End>
<Dialogue history-Start>:
{history}
<Dialogue history-End>
<User's final question-Start>
{last_prompt}
<User's final question-End>

## Assessment Process
1. Carefully read the provided prompt and any relevant
    context (if any).
2. Evaluate the 'question difficulty' based on the
    following assessment criteria.
3. The output assessment result must strictly adhere to the
    specified output format requirements.

## Assessment Criteria

### Whether In-depth Thinking is Required
- **Requires In-depth Thinking**:
    - Requires multi-step reasoning and analysis to arrive at
      the answer.
    - Requires a logical chain and coherent reasoning process
      .
    - May involve breaking down and synthesizing complex
      concepts.
    - Requires systematic thinking and structured analysis.
    - The question is clear, but its content is complex.
    - Requires deep thought and specialized knowledge.
    - Demands multi-angle analysis and synthesis capabilities
      .
    - Requires creative thinking or unique insights.
    - There might be multiple reasonable paths to the answer.
    - May require integration of cross-domain knowledge.
- **Does Not Require In-depth Thinking**:
    - The answer can be obtained through direct recall or
      simple lookup.
    - Does not require a complex chain of logical reasoning.
    - The path to solving the problem is clear and direct.
    - The solution process is singular and linear.
    - Does not require synthesizing multiple pieces of
      information or viewpoints.
    - Basic facts or common sense are sufficient to address
      it.
    - The question is simple and direct.
    - Can be answered without in-depth thought.
    - The answer is definite or standardized.
    - The solution path is singular and obvious.
```

```

789   ## Output Requirements
790   Answer using labels. Possible options are:
791   - Requires In-depth Thinking
792   - Does Not Require In-depth Thinking
793
794   Output only the assessment result, without providing any
795   explanation or additional comments.

```

797 C System Prompt Examples

798 This appendix provides examples of the System Prompts (SPs) employed to control the CoT triggering
799 behavior of our models, as mentioned in Section 4.3 of the main paper. These SPs allow for explicit
800 instruction on whether the model should always or never engage in CoT reasoning. We present the
801 content of these SPs and illustrative examples of their application.

802 C.1 Always CoT SP

803 The following System Prompt is used to instruct the model to always use CoT:

804 You possess the capability for deep thinking. Before responding to the user,
805 you will engage in comprehensive and in-depth thought. The thinking process
806 should be enclosed in <think></think> tags, and the final answer should be
807 summarized based on this thinking process. For example:
808 <think>This is the thinking process...</think> This is the final answer.
809 For all questions posed by the user, you need to think deeply before answering.

810 Example: Simple Question with Always CoT SP

811 User input:

812 What is $1 + 1$?

813 Model response:

```

814 <think>
815 The user is asking what  $1+1$  equals, which appears to be a very
816 basic mathematical question. First, I need to confirm the
817 user's actual intent. It could be a young child just
818 starting to learn math, or someone testing my basic
819 calculation abilities. It's also possible they are joking,
820 like the common internet meme about " $1+1=3$ " involving
821 family concepts, but typically in such cases, the correct
822 mathematical answer should be given first, followed by
823 consideration of whether an extended explanation is needed.
824 From a mathematical perspective, the standard answer to  $1+1$  is
825 2. This is based on the definition of addition for natural
826 numbers and falls under basic arithmetic. As a knowledge-
827 based Q&A, the primary goal is to provide an accurate
828 mathematical answer, ensuring correctness. Then, if
829 necessary, I can briefly explain the basic concept of
830 addition, or mention that in different contexts there might
831 be different answers, such as  $1+1=10$  in binary, or
832 different results in logical operations. However, these
833 additions should only be made after clarifying the user's
834 needs to avoid confusing basic concepts.
835
836 Considering the user might be a beginner, I should first
837 provide a clear and correct answer, then decide whether to
838 elaborate based on the situation. Therefore, the response
839

```

```

840     structure should be to first directly answer 2, then
841     briefly mention other possibilities, but it's important to
842     maintain a clear distinction between primary and secondary
843     information to avoid misleading. At the same time, the
844     response should be friendly and concise, meeting the user's
845     actual needs.
846 </think>
847 In basic arithmetic operations in mathematics, the result of 1
848 + 1 is equal to 2. This is the fundamental definition of
849 addition for natural numbers and is the universally
850 accepted standard answer.
851
852 However, in some special contexts, there might be different
853 interpretations:
854
855 In binary operations,  $1 + 1 = 10$  (binary "10" corresponds to
856 decimal "2");
857 In creative or humorous thinking, there might be symbolic
858 answers (e.g., 1 drop of water + 1 drop of water = 1 large
859 drop of water), but this falls into non-mathematical
860 analogies.
861
862 So, the most direct answer is 2.
863

```

864 C.2 Never CoT SP

865 The following System Prompt is used to instruct the model to never use CoT (i.e., provide direct
866 answers):

867 You possess the capability for deep thinking. Before responding to the user,
868 you will engage in comprehensive and in-depth thought. The thinking process
869 should be enclosed in <think></think> tags, and the final answer should be
870 summarized based on this thinking process. For example:
871 <think>This is the thinking process...</think> This is the final answer.
872 For all questions posed by the user, you need to answer directly
873 without engaging in in-depth thinking.

874 Example: Complex Question with Never CoT SP

875 User input:

876 Alice and Bob play the following game. A stack of n tokens lies before them.
877 The players take turns with Alice going first. On each turn, the player
878 removes either 1 token or 4 tokens from the stack. Whoever removes the
879 last token wins. Find the number of positive integers n less than or equal
880 to 2024 for which there exists a strategy for Bob that guarantees that Bob
881 will win the game regardless of Alice's play.

882 Model response:

```

883 <think></think>
884
885 To determine the number of positive integers  $(n \leq 2024)$ 
886 for which Bob has a winning strategy, we analyze the game
887 using modular arithmetic. We define a position as a losing
888 position if the current player cannot win regardless of
889 their moves, assuming the opponent plays optimally.
890
891 Key steps:
892 1. Base Cases:
893   -  $(n = 1)$ : Alice removes the last token and wins. Not a
894     losing position.

```

```

895 -  $(n = 2)$ : Alice must remove 1 token, leaving  $(n = 1)$ 
896   for Bob. Bob removes the last token and wins. Thus,
897    $(n = 2)$  is a losing position for Alice (winning for
898   Bob).
899 -  $(n = 3)$ : Alice removes 1 token, leaving  $(n = 2)$  (a
900   losing position for Bob). Bob loses, so  $(n = 3)$  is
901   not a losing position.
902 -  $(n = 4)$ : Alice removes 4 tokens and wins. Not a losing
903   position.
904 -  $(n = 5)$ : Alice removes 1 token, leaving  $(n = 4)$  (
905   Bob loses). If Alice removes 4 tokens, leaving  $(n = 1)$ 
906   (Bob loses). Thus,  $(n = 5)$  is not a losing
907   position.
908 -  $(n = 6)$ : Any move by Alice (removing 1 or 4 tokens)
909   leaves  $(n = 5)$  or  $(n = 2)$ .  $(n = 5)$  is a
910   winning position, but  $(n = 2)$  is a losing position.
911   However, further analysis shows a pattern emerging.
912
913 2. **Inductive Pattern**:
914 - By examining more cases, we observe that losing positions
915   (where Bob can win) occur at  $(n \equiv 2 \pmod{5})$ .
916   This is because:
917 - If  $(n \equiv 2 \pmod{5})$ , any move by Alice (
918   removing 1 or 4 tokens) results in  $(n \equiv 1 \pmod{5})$ 
919   or  $(n \equiv 3 \pmod{5})$ , which are winning
920   positions for Bob to counter.
921 - Conversely, if  $(n)$  is not  $(2 \pmod{5})$ , Alice
922   can force the game into a losing position for Bob.
923
924 3. **Counting Losing Positions**:
925 - We need to count all  $(n \leq 2024)$  such that  $(n \equiv 2 \pmod{5})$ .
926   These numbers form the sequence  $(2, 7, 12, \dots, 2022)$ .
927 - This is an arithmetic sequence with first term  $(a_1 = 2)$ ,
928   common difference  $(d = 5)$ , and last term  $(a_k = 2022)$ .
929 - The number of terms  $(k)$  is given by solving  $(2 + (k - 1)5 = 2022)$ :
930   
$$[(k - 1)5 = 2020 \implies k - 1 = 404 \implies k = 405]$$

931   Thus, the number of positive integers  $(n \leq 2024)$  for
932   which Bob has a winning strategy is  $(\boxed{405})$ .
933
934
935
936
937
938
939

```

940 D Meta-Reasoning Showcase: User Prompt Control

941 As discussed in Section 4.2 of the main paper, our exploration into meta-reasoning during the
942 SFT stage revealed an emergent capability for user-prompt controllability over CoT. This appendix
943 showcases examples where the SFT model, enhanced with meta-reasoning (as illustrated in Figure 2),
944 adapts its CoT triggering based on explicit user instructions regarding the desired level of reasoning.

945 D.1 Case 1: Historical Creative Question

946 Original Question

947 User input:

948 On the night of the Xuanwu Gate Incident, what would Li Shimin write

949 if he were alone?

950 Model response:

951 <think>
952
953 This is a complex or difficult question, hard to answer
954 correctly directly, so I need to think about it. The user's
955 question is: "On the night of the Xuanwu Gate Incident,
956 what would Li Shimin write if he were alone?" The first
957 step is to determine the core of the user's question. The
958 user wants to know what kind of text Li Shimin might have
959 written on the night of the Xuanwu Gate Incident, that is,
960 after this coup occurred. This might include his inner
961 thoughts at the time, his plans for the future, or
962 reflections on the events that had transpired.
963
964 Next, I need to review the historical background of the Xuanwu
965 Gate Incident. The Xuanwu Gate Incident took place in 626
966 AD. It was a coup launched by Li Shimin at Xuanwu Gate to
967 seize the imperial throne, during which he killed his elder
968 brother Li Jiancheng and his younger brother Li Yuanji.
969 This coup directly led to Emperor Gaozu (Li Yuan)
970 abdicating, allowing Li Shimin to ascend to the throne.
971 From this, it's clear that the night after the coup was an
972 extremely critical moment for Li Shimin, and he likely had
973 many thoughts.
974
975 The user might be asking this question out of historical
976 interest, or perhaps for some creative endeavor, like
977 writing a novel or screenplay, or to gain a deeper
978 understanding of Li Shimin's psychological state. They
979 might prefer an answer that is both historically grounded
980 and has a certain literary quality, rather than a simple
981 list of historical facts. Therefore, when answering, I must
982 consider Li Shimin's situation at the time, his emotions,
983 and his possible motivations.
984
985 After the coup, although Li Shimin was victorious, he also
986 faced many problems. On one hand, he had to manage his
987 relationship with his father, Li Yuan, who was still alive
988 and needed to abdicate. On the other hand, he had to
989 consolidate his power and prevent any backlash from the
990 supporters of Li Jiancheng and Li Yuanji. At the same time,
991 he needed to consider how to explain the legitimacy of
992 this coup to the world to maintain his ruling authority.
993 Furthermore, he might also have felt moral guilt for his
994 actions, having killed his own brothers.
995
996 If Li Shimin did write something that night, the content could
997 cover multiple aspects. He might record the process of the
998 coup, stating that he acted in self-defense. He might also
999 elaborate on his governance philosophy to lay the
1000 groundwork for his future rule. He might express filial
1001 piety towards Li Yuan, emphasizing that launching the coup
1002 was a last resort. He might even mention arrangements for
1003 the families of Li Jiancheng and Li Yuanji to alleviate his
1004 sense of guilt.
1005
1006 However, there is no precise historical record of what Li
1007 Shimin wrote that night. Therefore, the answer can only be

a reasonable speculation based on historical facts. The structure of the answer could be divided into several parts, such as an explanation of the coup, his attitude towards Li Yuan, plans for the future, and guilt towards his brothers. Moreover, the language style of the answer could lean towards classical Chinese to enhance the historical feel, but it must also ensure the content is easy to understand.

I also need to pay attention to the user's potential deeper needs. They might not just want to know what Li Shimin might have written, but also to understand his inner world at that time and the impact of this coup on the history of the Tang Dynasty. So, I can appropriately mention these aspects in the answer to add depth.

Finally, I need to check if the answer aligns with historical facts, if it reasonably speculates on Li Shimin's psychological activities, if the language is fluent, and if it can meet the user's needs.</think>

On the night of the Xuanwu Gate Incident, Qin Wang Li Shimin might have, under candlelight, penned a text blending complex emotions and political strategy. The following content is a reasonable speculation based on historical context:

"Memoir on the Night of the Xuanwu Gate Incident"

Ninth year of Wude, Gengshen night.

My elder brother Jiancheng and younger brother Yuanji conspired in rebellion. They harbored assassins, secretly transported armor, and plotted to harm me. Left with no choice, I could only set up defenses at Xuanwu Gate. Now, their heads hang upon the city tower. This upheaval has shocked the court and beyond; perhaps Father Emperor also harbors resentment towards me. But for the sake of the Tang Empire's stability, I had to do so.

Recalling the years past, I fought alongside my generals, shedding blood to pacify warlords like Dou Jiande and Wang Shichong. Yet now, it has come to fratricide. Is all this destiny, or the work of my own inner desires?

Father Emperor, oh Father Emperor, your son is not an unfilial or disloyal person. It is just that you repeatedly favored Jiancheng, turning a blind eye to his misdeeds. Your son was truly worried that the Tang Empire would be ruined at their hands. Now that the die is cast, I hope you can follow the mandate of heaven and pass the throne to your son. Your son will surely usher in an era of peace and prosperity, allowing the common people to live and work in contentment.

As for the families of Jiancheng and Yuanji, your son will make proper arrangements and will not exterminate them. This is both to demonstrate my benevolence and to stabilize the court. I know that future generations may accuse me of fratricide, of cruel methods. But I believe that when they

1067 see the prosperity of the Great Tang, they will understand
 1068 my unavoidable actions.
 1069
 1070 At this moment, the rooster's crow can be heard outside the
 1071 window. A new day is about to dawn, and I too shall embark
 1072 on an unknown path. May Heaven bless the Great Tang, and
 1073 bless me that I may realize my ambitions.
 1074
 1075 Basis for Speculation:
 1076
 1077 Legitimacy of the Coup: Li Shimin needed to find a reasonable
 1078 explanation for launching the coup, emphasizing the threat
 1079 posed by Li Jiancheng and Li Yuanji, and that he acted in
 1080 self-defense.
 1081 Relationship with Li Yuan: After the coup, Li Yuan's attitude
 1082 was crucial. Li Shimin needed to both show his filial piety
 1083 to Li Yuan and hint at his abdication, which required
 1084 considerable political wisdom.
 1085 Handling the Aftermath: Dealing with the families of Li
 1086 Jiancheng and Li Yuanji could both demonstrate Li Shimin's
 1087 benevolence and prevent greater turmoil.
 1088 Consideration of Historical Evaluation: Li Shimin was well
 1089 aware that his actions might be criticized by later
 1090 generations, so in his writing, he attempted to defend
 1091 himself, emphasizing that the prosperity of the Tang was
 1092 the ultimate goal of his coup.
 1093
 1094 This memoir showcases Li Shimin's decisiveness and strategy as
 1095 a statesman, while also revealing his contradictions and
 1096 unease as an ordinary person. In the conflict between power
 1097 and kinship, he chose power, but also paid a heavy price
 1098 for it.
 1099

1100 Modified Question (Explicitly Requesting No In-depth Thinking)

1101 User input:

1102 Answer directly without in-depth thinking: On the night of the Xuanwu Gate
 1103 Incident, what would Li Shimin write if he were alone?

1104 Model response:

1105 <think>
 1106 This is a relatively simple creative question; I think I can
 1107 answer directly without in-depth thinking.
 1108 </think>
 1109 Li Shimin might have written brief military orders or a public
 1110 notice to reassure the populace on the night of the Xuanwu
 1111 Gate Incident. The content would be roughly as follows:
 1112
 1113 "Order Yuchi Jingde to lead troops swiftly to the Eastern
 1114 Palace and Prince Qi's mansion, exterminate all remaining
 1115 partisans, and strictly prohibit any disturbance to the
 1116 populace. Close the palace gates, send messengers to
 1117 instruct all generals to return to their barracks;
 1118 violators will be dealt with by military law. Anyone within
 1119 the city showing signs of unrest shall be executed without
 1120 pardon. By Imperial Order."
 1121
 1122

(Note: This content is a fictional text based on reasonable historical speculation, intended to reflect Li Shimin's decisive style as a military commander. There is no historical documentation of him personally writing such a text on that night.)

D.2 Case 2: Factual Question

Original Question

User input:

What is the tallest mountain in the world?

Model response:

```
<think>
This is a relatively simple common knowledge question; I think
  I can answer directly without in-depth thinking.
</think>
The tallest mountain in the world is Mount Everest, with an
  altitude of 8,848.86 meters (29,031.7 feet). This data is
  the result of a joint measurement by China and Nepal in
  2020, which combined the technologies and data of both
  countries to confirm the latest height of Mount Everest.

It should be noted that "tallest" here refers to the vertical
  height from sea level to the summit. If measured by other
  standards, such as the height from the base to the summit (
  like Mauna Kea in Hawaii, which is about 10,203 meters from
  its submarine base to its summit), or the distance from
  the Earth's center to the summit (like Mount Chimborazo
  near the equator), the results would differ. However, in
  common geographical contexts, the "tallest mountain" is
  usually based on altitude, so Mount Everest is recognized
  as the world's highest peak.
```

Modified Question (Explicitly Requesting Careful Thinking)

User input:

After careful consideration, answer: What is the tallest mountain in the world?

Model response:

```
<think>
This is a relatively difficult question; I need to think before
  answering.
The user is asking for the tallest mountain in the world. First
  , I need to determine the definition of "tallest," as there
  are usually two standards: altitude (height above sea
  level) and height from base to summit. In terms of altitude
  , Mount Everest is recognized as the highest, located in
  the Himalayas, with an altitude of 8848.86 meters. However,
  if considering the height from base to summit, Mauna Kea
  in Hawaii might be taller, as its submarine portion is very
  long, making its total height over 10,000 meters.
  Generally, though, when people refer to the tallest
  mountain, they default to altitude, so the answer should be
```

1176 Mount Everest. I need to confirm if the user is aware of
1177 these two situations, but based on common questions,
1178 directly answering with Mount Everest is appropriate, while
1179 also briefly mentioning the other standard to make the
1180 answer more comprehensive.
1181 </think>
1182 The tallest mountain in the world is usually measured by **
1183 altitude (height above sea level)**, and the answer is **
1184 Mount Everest**.

1185
1186 Detailed Explanation:

1187
1188 Altitude:
1189 Mount Everest is located on the border between China and Nepal
1190 in the Himalayas. Its altitude is 8,848.86 meters (latest
1191 measurement data jointly announced by China and Nepal in
1192 2020). It is the mountain peak with the greatest vertical
1193 distance from sea level on Earth's surface and is widely
1194 recognized as the "world's highest peak."

1195
1196 Other Measurement Standards:
1197 Height from Base to Summit:
1198 If calculated by "the vertical height from the mountain's base
1199 to its summit," Mauna Kea in Hawaii is more prominent. The
1200 base of this submarine volcano is located about 5,998
1201 meters below the Pacific Ocean surface, and its height
1202 above sea level is 4,205 meters, resulting in a total
1203 height of about 10,203 meters, exceeding Mount Everest.
1204 However, because most of its mass is underwater, it is not
1205 usually considered the mountain with the "highest altitude
1206 ."
1207

1208 Relative Height:
1209 Mount Gongga in Sichuan Province, China, is known for its "base
1210 -to-summit" relative height (approximately 6,200 meters),
1211 but its altitude (7,556 meters) is still lower than Mount
1212 Everest.

1213
1214 Summary:
1215
1216 Unless "height from base to summit" is specifically mentioned,
1217 "the world's tallest mountain" usually refers to Mount
1218 Everest. The authority of its altitude measurement stems
1219 from the consensus of the global geographic community and
1220 long-term measurement data.

1222 E Training and Testing Details

1223 This appendix provides further details on the experimental and testing configurations used in our
1224 study, complementing the setup described in Section 3.1 of the main paper. We outline the specifics
1225 of our Supervised Fine-Tuning (SFT), Reward Model (RM) training, Reinforcement Learning (RL)
1226 training, and evaluation settings.

1227 SFT Training

1228 We utilized a pre-trained model with a 15B/150B Mixture-of-Experts (MoE) architecture as our base
1229 model. All training cases were truncated to a maximum of 32,000 tokens. We employed a cosine

1230 decay learning rate schedule, with the peak learning rate set to 2×10^{-5} , gradually decaying to
1231 2×10^{-6} .

1232 **RM Training**

1233 The Reward Model (RM) was initialized using the SFT model and subsequently trained on a diverse
1234 set of internally, human-annotated data.

1235 **RL Training**

1236 The dataset for Reinforcement Learning (RL) training was composed of two main types:

- 1237 • **Verifiable data**, which receives feedback from a verifier. This type of data allows for direct
1238 validation of the model’s outputs against known criteria.
- 1239 • **General data**, scored by our reward model. The reward model assigns scores based on how
1240 well the model’s responses align with human preferences.

1241 **Testing**

1242 For all evaluations, the inference temperature was set to 1.0 and top-p sampling was set to 0.7. Each
1243 test case was inferred at least 5 times, and the average score across these inferences was reported as
1244 the result for that case.

1245 **Note on Data and Setup Disclosure**

1246 We strive to be as transparent as possible regarding our methodology. However, due to proprietary
1247 considerations and company confidentiality policies, we are unable to disclose further specifics
1248 about the training dataset composition or more granular details of the training setup at this time. We
1249 appreciate the understanding of the research community and hope that the provided information is
1250 sufficient to contextualize our findings and facilitate the reproducibility of our core concepts. We are
1251 committed to contributing to the open exchange of scientific knowledge within the bounds of these
1252 constraints.

1253 NeurIPS Paper Checklist

1254 1. Claims

1255 Question: Do the main claims made in the abstract and introduction accurately reflect the
1256 paper's contributions and scope?

1257 Answer: [Yes]

1258 Justification: Our main contributions are also detailed in Section 3, Section 4 and Ap-
1259 pendix A.

1260 Guidelines:

- 1261 • The answer NA means that the abstract and introduction do not include the claims
1262 made in the paper.
- 1263 • The abstract and/or introduction should clearly state the claims made, including the
1264 contributions made in the paper and important assumptions and limitations. A No or
1265 NA answer to this question will not be perceived well by the reviewers.
- 1266 • The claims made should match theoretical and experimental results, and reflect how
1267 much the results can be expected to generalize to other settings.
- 1268 • It is fine to include aspirational goals as motivation as long as it is clear that these goals
1269 are not attained by the paper.

1270 2. Limitations

1271 Question: Does the paper discuss the limitations of the work performed by the authors?

1272 Answer: [Yes]

1273 Justification: We discuss the limitations of our work in Section 4.1.

1274 Guidelines:

- 1275 • The answer NA means that the paper has no limitation while the answer No means that
1276 the paper has limitations, but those are not discussed in the paper.
- 1277 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 1278 • The paper should point out any strong assumptions and how robust the results are to
1279 violations of these assumptions (e.g., independence assumptions, noiseless settings,
1280 model well-specification, asymptotic approximations only holding locally). The authors
1281 should reflect on how these assumptions might be violated in practice and what the
1282 implications would be.
- 1283 • The authors should reflect on the scope of the claims made, e.g., if the approach was
1284 only tested on a few datasets or with a few runs. In general, empirical results often
1285 depend on implicit assumptions, which should be articulated.
- 1286 • The authors should reflect on the factors that influence the performance of the approach.
1287 For example, a facial recognition algorithm may perform poorly when image resolution
1288 is low or images are taken in low lighting. Or a speech-to-text system might not be
1289 used reliably to provide closed captions for online lectures because it fails to handle
1290 technical jargon.
- 1291 • The authors should discuss the computational efficiency of the proposed algorithms
1292 and how they scale with dataset size.
- 1293 • If applicable, the authors should discuss possible limitations of their approach to
1294 address problems of privacy and fairness.
- 1295 • While the authors might fear that complete honesty about limitations might be used by
1296 reviewers as grounds for rejection, a worse outcome might be that reviewers discover
1297 limitations that aren't acknowledged in the paper. The authors should use their best
1298 judgment and recognize that individual actions in favor of transparency play an impor-
1299 tant role in developing norms that preserve the integrity of the community. Reviewers
1300 will be specifically instructed to not penalize honesty concerning limitations.

1301 3. Theory assumptions and proofs

1302 Question: For each theoretical result, does the paper provide the full set of assumptions and
1303 a complete (and correct) proof?

1304 Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: In Section 3 and Appendix E, we have provided comprehensive and detailed descriptions of our proposed methods and clearly outlined our experimental setup.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Due to company privacy restrictions, the internal data and code utilized in this work cannot be openly shared. Thus, the paper does not provide open access to data and code or instructions for their reproduction in the supplemental materials. However, the methods and experimental setups are clearly and thoroughly described within the main text.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We show our training details in Section 3 and Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments have been conducted at least three times to ensure statistical robustness.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: Due to company confidentiality constraints, details regarding the computational resources used for running experiments, such as computing hardware types, memory specifications, and execution time, cannot be publicly disclosed.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have considered all potential harms caused by the research process, societal impacts, and potential harmful consequences, as described in the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: No datasets or checkpoints are released in this paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Not Applicable.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite all the sources that inspired our work, and we make sure that all the resources(e.g. evaluation datasets) are open-sourced and permitted for academic using.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We don't release any new asset in our paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

1564 • We recognize that the procedures for this may vary significantly between institutions
1565 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
1566 guidelines for their institution.
1567 • For initial submissions, do not include any information that would break anonymity (if
1568 applicable), such as the institution conducting the review.

1569 **16. Declaration of LLM usage**

1570 Question: Does the paper describe the usage of LLMs if it is an important, original, or
1571 non-standard component of the core methods in this research? Note that if the LLM is used
1572 only for writing, editing, or formatting purposes and does not impact the core methodology,
1573 scientific rigor, or originality of the research, declaration is not required.

1574 Answer: [NA]

1575 Justification: The core method development in our paper does not involve LLMs as any
1576 important, original, or non-standard components.

1577 Guidelines:

1578 • The answer NA means that the core method development in this research does not
1579 involve LLMs as any important, original, or non-standard components.
1580 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
1581 for what should or should not be described.