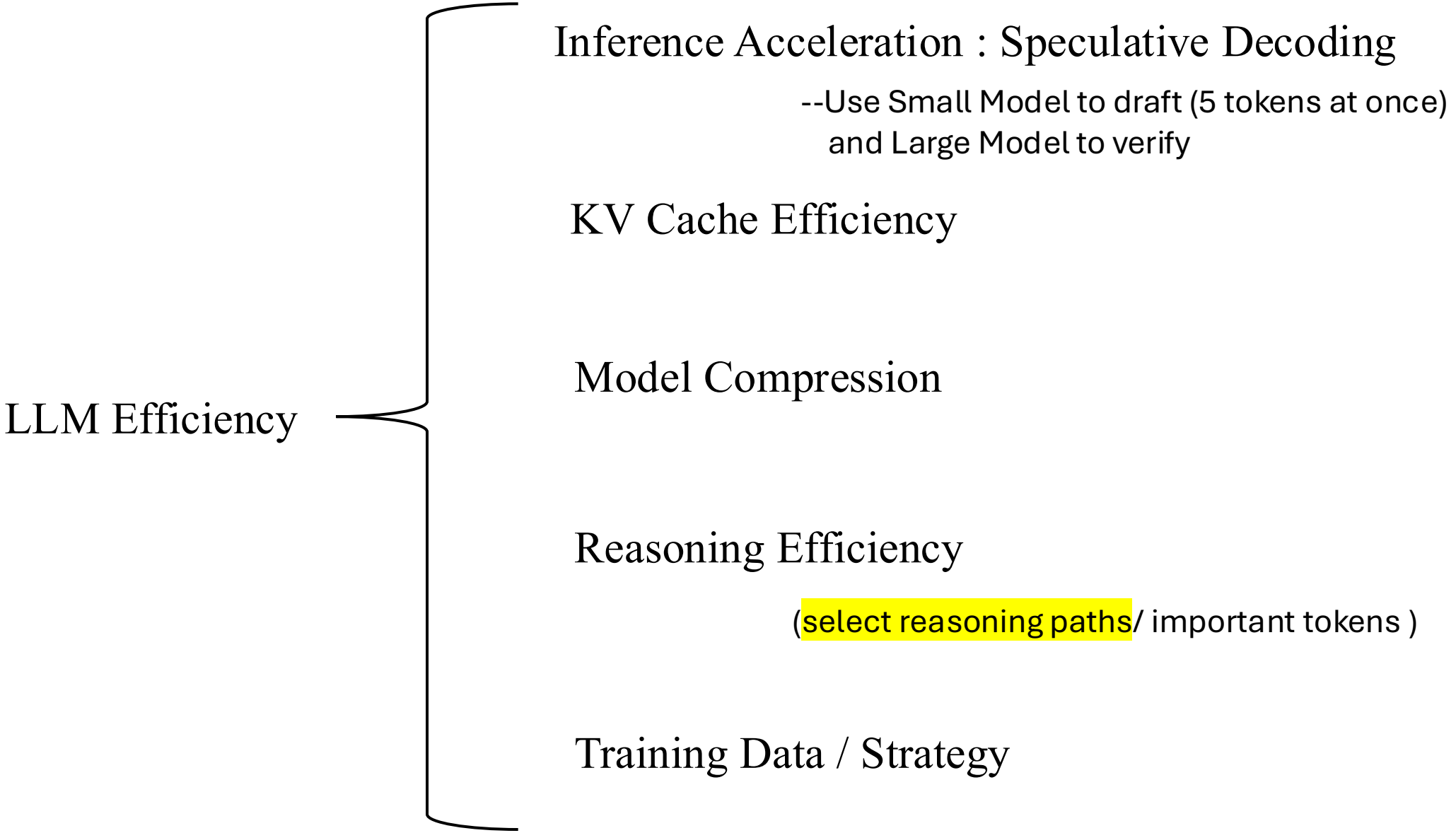
Fine-Grained Evaluation of LLM Reasoning

July 22, 2026


LLM Efficiency
Inference Acceleration : Speculative Decoding
--Use Small Model to draft (5 tokens at once)
and Large Model to verify
KV Cache Efficiency
Model Compression
Reasoning Efficiency
(select reasoning paths/ important tokens)
Training Data / Strategy

Introduction

- Coarse-Grained: get a reward score after the entire reasoning completed

  Fine-Grained: use signals to guide reasoning become more efficient and informative in the process.

- -- Entropy comes to mind since it evaluate uncertainty (LLM confidence), information gai ($IG(X, A) = H(X) - H(X|A)$) and is calculated by probability (LLM probability distribution after softmax).

  -- Could also consider knowledge boundary, multi-objective optimization…

ETR: Entropy Trend Reward for Efficient Chain-of-Thought Reasoning

Xuan Xiong[1] **Huan Liu**[2*] **Li Gu**[3] **Zhixiang Chi**[1] **Yue Qiu**[4] **Yuanhao Yu**[2] **Yang Wang**[3]

[1]University of Toronto [2]McMaster University [3]Concordia University [4]University of Ottawa

ACL 2026 Oral

Motivation

- CoT often generates long, repetitive, and self-contradicting traces;
- Existing methods shorten CoTs using length penalties (content-blind), or global entropy reduction (discourage useful backtracing);
- Their hypothesis is, the reasoning efficiency is governed by the trajectory of uncertainty, not absolute magnitude.

Motivation

They conduct empirical study to support the hypothesis:

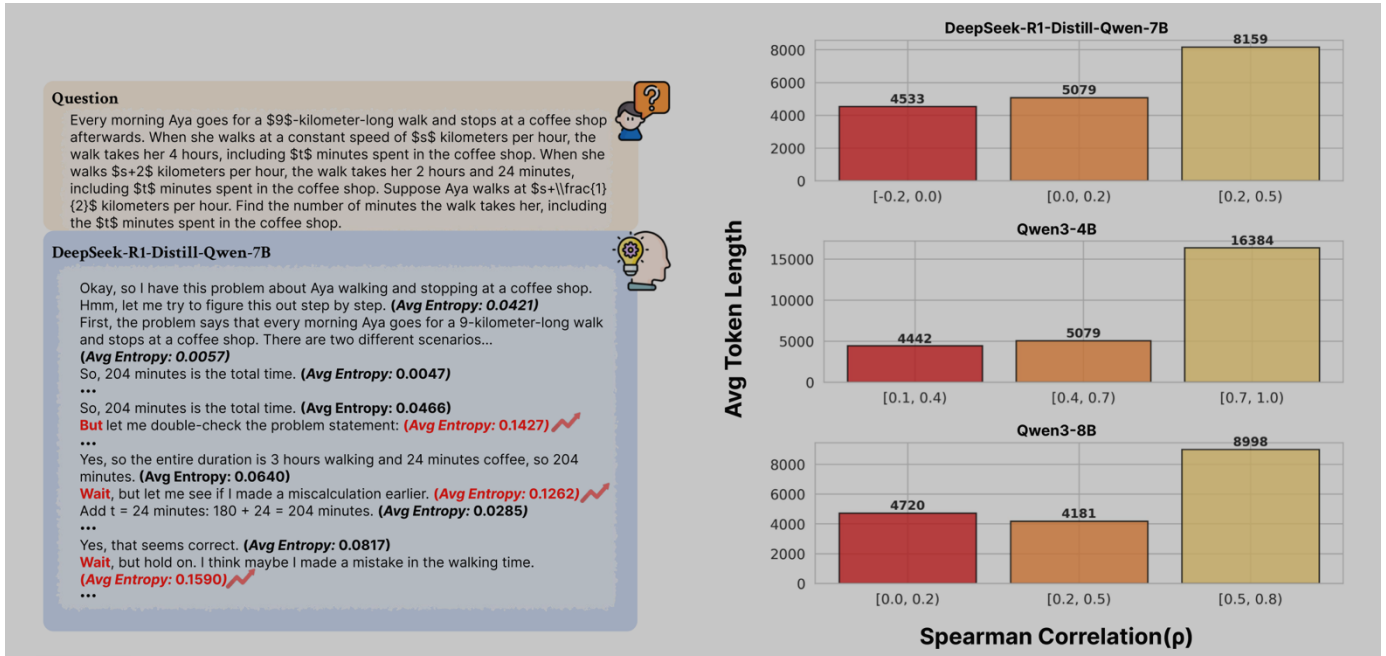


Figure 2: Step-wise entropy dynamics in generated CoTs on the MATH500 dataset (Hendrycks et al., 2024). (Left) A sample response illustrating high step-wise entropy during self-reflection, marked by uncertainty cues (e.g., but, wait, etc.). (Right) Across models, a higher Spearman rank correlation (ρ) between reasoning step index and step-wise entropy indicates persistent entropy growth during reasoning. This is consistently associated with longer outputs and greater token redundancy.

- The x-axis bin shows spearman correlation (a lower value indicates downward entropy trend with reasoning steps);
- The y-axis shows with a downward entropy trend, the reasoning tend to be short;
- Left shows that a high-entropy token tend to be a keyword for self-reflection.

Entropy Trend Reward

- The key idea - to encourage progressive uncertainty reduction along a CoT while preserve correctness;

- Also integrate it into GRPO as a scalar reward signal.

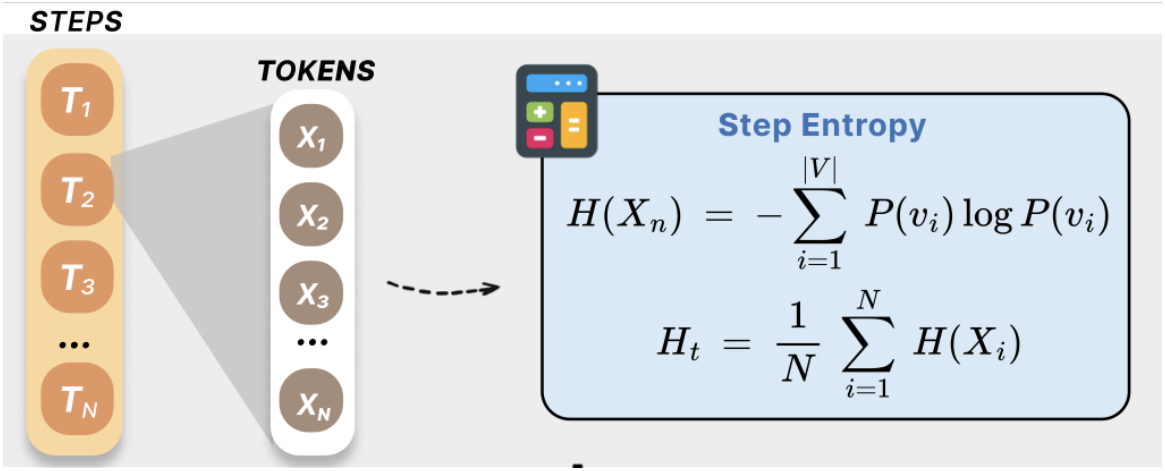
$$R(q,o) = \begin{cases} -1, & \text{if incorrect,} \\ 1 + \lambda R_{\text{entropy}}(o), & \text{if correct.} \end{cases}$$

Entropy Trend Reward

Step 1: Entropy Trajectory of Chain-of-Thought

1) Let $o = \{C_1, C_2, \ldots, C_T\}$ denotes a Chain-of-Thought response,where T represents each reasoning step.

2) Then, for each step t, they define a scalar entropy $H_t = H(p_\theta(\cdot \mid C_{1:t}))$



Entropy Trend Reward

Step 1: Entropy Trajectory of Chain-of-Thought

3) Finally, they define the step-wise entropy change as:

$$\Delta_t = H_{t-1} - H_t, \qquad t = 2, \ldots, T.$$

A positive Δ_t indicates uncertainty reduction.

Entropy Trend Reward

Step 2: Momentum-Based Entropy Trend Reward

Why this step is needed: A simple summation over step-wise change does not make sense:

$$R_{naive}(o) = \sum_{t=2}^{T} \Delta_t = (H_1 - H_2) + (H_2 - H_3) + \cdots + (H_{T-1} - H_T)$$

$$= H_1 - H_T$$

Entropy Trend Reward

Step 2: Momentum-Based Entropy Trend Reward

To address the above issue, they introduce a latent trend variable S_t recursively.

$$S_t = \gamma S_{t-1} + \Delta_t, \qquad S_1 = 0$$

The entropy reward for a CoT is then defined as $R_{\text{entropy}}(o) = \sum_{t=2}^{T} S_t$

Unrolling the recurrence yields $R_{\text{entropy}}(o) = \sum_{t=2}^{T} \alpha_t \, \Delta_t,$

$$\text{where } \alpha_t = \frac{1 - \gamma^{T-t+1}}{1 - \gamma}$$

Step 1: Unroll the recursion. For any $t \geq 2$, repeatedly substituting Eq. (14) gives

$$\begin{aligned} S_t &= \gamma S_{t-1} + \Delta_t \\ &= \gamma(\gamma S_{t-2} + \Delta_{t-1}) + \Delta_t \\ &= \gamma^2 S_{t-2} + \gamma \Delta_{t-1} + \Delta_t \\ &\vdots \\ &= \gamma^{t-2} S_2 + \sum_{k=3}^{t} \gamma^{t-k} \Delta_k. \end{aligned} \tag{16}$$

Since $S_2 = \gamma S_1 + \Delta_2 = \Delta_2$ and $S_1 = 0$, Eq. (16) simplifies to

$$S_t = \sum_{k=2}^{t} \gamma^{t-k} \Delta_k. \tag{17}$$

Step 2: Substitute into the reward and swap summations. Plugging Eq. (17) into Eq. (15) yields

$$\begin{aligned} R_{\text{entropy}}(o) &= \sum_{t=2}^{T} \sum_{k=2}^{t} \gamma^{t-k} \Delta_k \\ &= \sum_{k=2}^{T} \Delta_k \sum_{t=k}^{T} \gamma^{t-k}, \end{aligned} \tag{18}$$

where the last line follows by exchanging the order of summation.

Step 3: Evaluate the inner geometric series. Let $m = t - k$, so that $m = 0, \ldots, T - k$. Then

$$\sum_{t=k}^{T} \gamma^{t-k} = \sum_{m=0}^{T-k} \gamma^{m}. \tag{19}$$

For $\gamma \neq 1$, this is a finite geometric series:

$$\sum_{m=0}^{T-k} \gamma^{m} = \frac{1 - \gamma^{T-k+1}}{1 - \gamma}. \tag{20}$$

Substituting Eq. (20) into Eq. (18) gives

$$R_{\text{entropy}}(o) = \sum_{k=2}^{T} \alpha_k \Delta_k, \qquad \alpha_k = \frac{1 - \gamma^{T-k+1}}{1 - \gamma}. \tag{21}$$

Renaming the index $k \mapsto t$ yields Eq. (10) in the main text:

$$R_{\text{entropy}}(o) = \sum_{t=2}^{T} \alpha_t \Delta_t, \qquad \alpha_t = \frac{1 - \gamma^{T-t+1}}{1 - \gamma}. \tag{22}$$

Entropy Trend Reward

Step 2: Momentum-Based Entropy Trend Reward

A property: α_t strictly decreases in t.

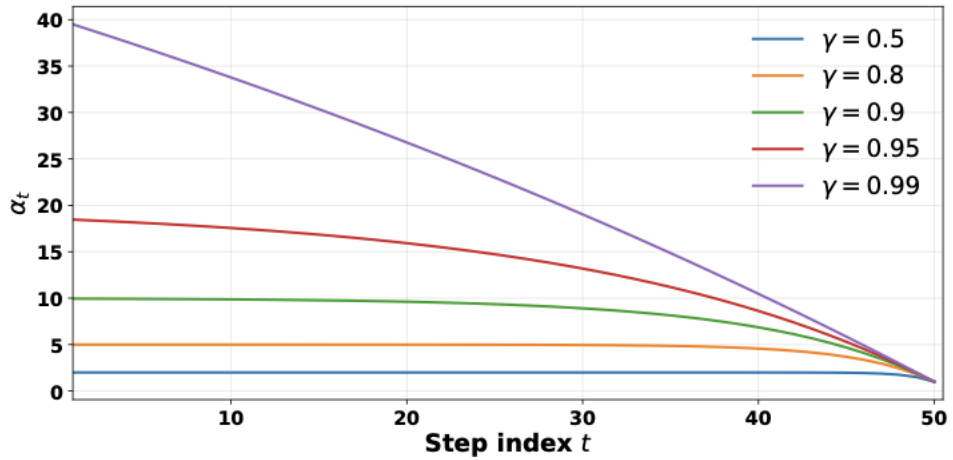


Figure 4: Evolution of the cumulative momentum weights α_t under different momentum coefficients γ.

- This means reductions in uncertainty that occur earlier in the reasoning trajectory are rewarded more;
- This encourages the model to converg its belief state rapidly rather than postponing critical deductions to late steps.

Entropy Trend Reward

A Property: Instance-Adaptive Termination via Entropy Trend Shaping.

ETR maintains a momentum-aggregated trend state $S_t = \gamma S_{t-1} + \Delta_t$ with $\Delta_t = H_{t-1} - H_t$, and the entropy reward accumulates $\sum_{t=2}^{T} S_t$. This means we only encourage steps with entropy descent.

This implicitly encourage shorter length for easy instances.

Trajectory –aware for longer (how?)

GRPO

It is a policy gradient approach that evaluates model outputs based on their relative performance within a group of sampled responses.

- For a given input question q, GRPA samples a set of G candidate responses $\{o_1, o_2, \ldots, o_G\}$ from the current policy.

- Then, the relative advantage is calculated as $\hat{A}_i = \frac{r_i - \frac{1}{G}\sum_{j=1}^{G} r_j}{\sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \mu_r)^2}}$.

- The GPRO objective is $\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G} \min \mathcal{L}_{i,t}(\theta) - \beta\, \text{KL}(\pi_\theta)\right]$

where $\mathcal{L}_{i,t}(\theta) = \left(\tau_i(\theta)\hat{A}_i,\ \text{clip}\left(\tau_i(\theta),\ 1-\epsilon,\ 1+\epsilon\right)\hat{A}_i\right)$

Integration with GRPO

$$R(q, o) = \begin{cases} -1, & \text{if incorrect,} \\ 1 + \lambda R_{\text{entropy}}(o), & \text{if correct.} \end{cases}$$

Ensures correctness is a hard constraint, while entropy shaping only affects the correct reasoning trajectories.

Experiments – Overall Performance

Method	AMC23			AIME24			MATH500			GPQA-D			Overall		
	Acc↑	Len↓	CR↓	Acc↑	Len↓	CR↓	Acc↑	Len↓	CR↓	Acc↑	Len↓	CR↓	Acc↑	Len↓	AES↑
DeepSeek-R1-Distill-7B															
Original	80.0	6.6k	100%	43.3	11.8k	100%	85.0	4.2k	100%	24.2	11.3k	100%	58.1	8.5k	0.00
DEER	85.0	4.9k	74.2%	50.0	9.9k	83.9%	85.8	2.3k	54.8%	22.7	7.6k	67.3%	60.9	6.2k	0.51
NoThink	77.5	3.8k	57.6%	56.7	8.3k	70.3%	81.4	1.7k	40.5%	22.2	2.0k	17.7%	59.5	4.0k	0.65
LCPO	87.5	3.5k	53.0%	50.0	6.8k	57.8%	85.4	2.2k	52.6%	11.61	2.8k	24.6%	58.6	3.8k	0.60
O1-Pruner	92.5	3.2k	48.4%	46.7	7.8k	66.1%	89.0	2.1k	50.0%	39.4	6.2k	54.9%	66.9	4.8k	1.18
PEAR	92.5	4.4k	66.04%	60.0	8.5k	72.2%	90.2	2.5k	59.9%	36.9	5.2k	45.9%	69.8	5.1k	1.41
ETR	**87.5**	**2.4k**	**36.4%**	**56.7**	**4.6k**	**39.0%**	**90.6**	**1.5k**	**35.7%**	**37.3**	**2.5k**	**22.1%**	**68.0**	**2.8k**	**1.53**
Qwen3-4B															
Original	90.0	7.6k	100%	53.3	11.7k	100%	90.6	5.0k	100%	43.9	10.4k	100%	69.5	8.7k	0.00
DEER	92.5	6.1k	80.3%	56.7	11.5k	98.3%	92.2	3.7k	74.0%	52.5	8.2k	78.8%	73.5	7.4k	0.44
NoThink	82.5	2.4k	31.6%	33.3	8.0k	68.4%	85.6	1.7k	34.0%	23.2	4.7k	45.2%	56.2	4.2k	-0.44
LCPO	90.0	6.0k	78.9%	50.0	8.2k	70.6%	90.8	4.5k	89.2%	47.5	3.8k	36.3%	69.6	5.6k	0.36
O1-Pruner	85.0	2.6k	34.2%	50.0	6.9k	59.0%	90.6	1.8k	36%	50.0	3.3k	31.7%	68.0	3.6k	0.54
PEAR	92.5	6.0k	34.2%	70.0	9.9k	59.0%	91.8	3.8k	36%	54.54	7.1k	31.7%	77.2	6.7k	0.79
ETR	**90.0**	**4.0k**	**52.6%**	**73.3**	**7.5k**	**64.1%**	**91.4**	**2.1k**	**42.0%**	**53.5**	**4.1k**	**39.4%**	**77.1**	**4.4k**	**1.03**
Qwen3-8B															
Original	90.0	8.0k	100%	63.3	12.2k	100%	90.6	5.4k	100%	52.0	9.9k	100%	74.0	8.9k	0.00
DEER	92.5	6.4k	80.0%	63.3	10.3k	84.4%	93.0	3.1k	57.4%	61.6	9.0k	90.9%	77.6	7.2k	0.43
NoThink	67.5	3.4k	41.3%	40.0	7.0k	57.4%	86.4	1.5k	27.8%	26.8	4.4k	44.4%	55.2	4.1k	-0.73
LCPO	90.0	5.7k	41.3%	60.0	7.0k	57.4%	93.6	4.8k	27.8%	54.6	4.0k	44.4%	74.5	5.4k	0.43
O1-Pruner	90.0	3.3k	41.3%	66.7	6.6k	54.1%	92.2	1.9k	35.2%	56.1	4.1k	41.4%	76.3	4.0k	0.70
PEAR	87.5	6.8k	84.7%	63.3	11.0k	90.1%	92.4	4.5k	83.5%	55.1	8.2k	83.3%	74.6	7.6k	0.18
ETR	**92.5**	**4.2k**	**53.8%**	**73.3**	**8.6k**	**75.4%**	**93.6**	**2.4k**	**44.4%**	**57.1**	**5.2k**	**52.5%**	**79.1**	**5.1k**	**0.77**

Table 1: Evaluation results on four benchmarks using greedy decoding (pass@1). **Acc** denotes accuracy, **Len** denotes token length, and **CR** denotes compression rate. ↑ and ↓ indicate higher- and lower-is-better, respectively. **AES** is a unified score that jointly accounts for accuracy and efficiency.

Evaluate
1) accuracy,
2) response length,
3) Compression Rate:

$$\frac{New\ Avg(response\ length)}{Original\ Avg\ (response\ length)}$$

4) Accuracy-Efficiency Trade-off Score (AES):

$$\text{AES} = \underbrace{\frac{L_{\text{base}} - L_{\text{model}}}{L_{\text{base}}}}_{\text{relative length reduction}} + \varsigma \underbrace{\frac{A_{\text{model}} - A_{\text{base}}}{A_{\text{base}}}}_{\text{relative accuracy chang}}$$

Empirical Analysis – Ablation Study

- Conduct an ablation study on entropy-based reward designs to isolate the effects of entropy magnitude, temporal aggregation, and correct supervision.

Reward	AMC23		AIME24		MATH500		GPQA-D		AES↑
	Acc ↑	Len ↓	Acc ↑	Len ↓	Acc ↑	Len ↓	Acc ↑	Len ↓	
Original	80.0	6.6k	43.3	11.8k	85.0	4.2k	24.2	11.3k	0.00
Min. H	80.0	2.1k	43.3	5.1k	88.2	1.3k	38.3	2.1k	1.06
Max. H	10.0	15.1k	0.0	16.4k	9.0	15.3k	1.5	16.0k	-5.4
No γ	87.5	4.9k	46.67	10.0k	87.8	3.6k	31.8	10.0k	0.61
No R_{corr}	65.0	1.2k	23.3	1.4k	78.6	0.7k	29.8	0.7k	0.11
Ours	**87.5**	**2.4k**	**56.7**	**4.6k**	**90.6**	**1.5k**	**37.4**	**2.5k**	**1.53**

Table 2: Ablation Study on Entropy-based Reward Designs. We compare our momentum-based reward against extreme constraints and simplified variants across four benchmarks. Our method demonstrates a superior balance between accuracy and efficiency.

Entropy minimization

Collapses with high entropy

(only relies on entropy differences between first and last sentences)

$$S_t = \gamma S_{t-1} + \Delta_t, \qquad S_1 = 0,$$

$$R(q, o) = \begin{cases} -1, & \text{if incorrect,} \\ 1 + \lambda R_{\text{entropy}}(o), & \text{if correct.} \end{cases}$$

Empirical Analysis – Entropy Trend

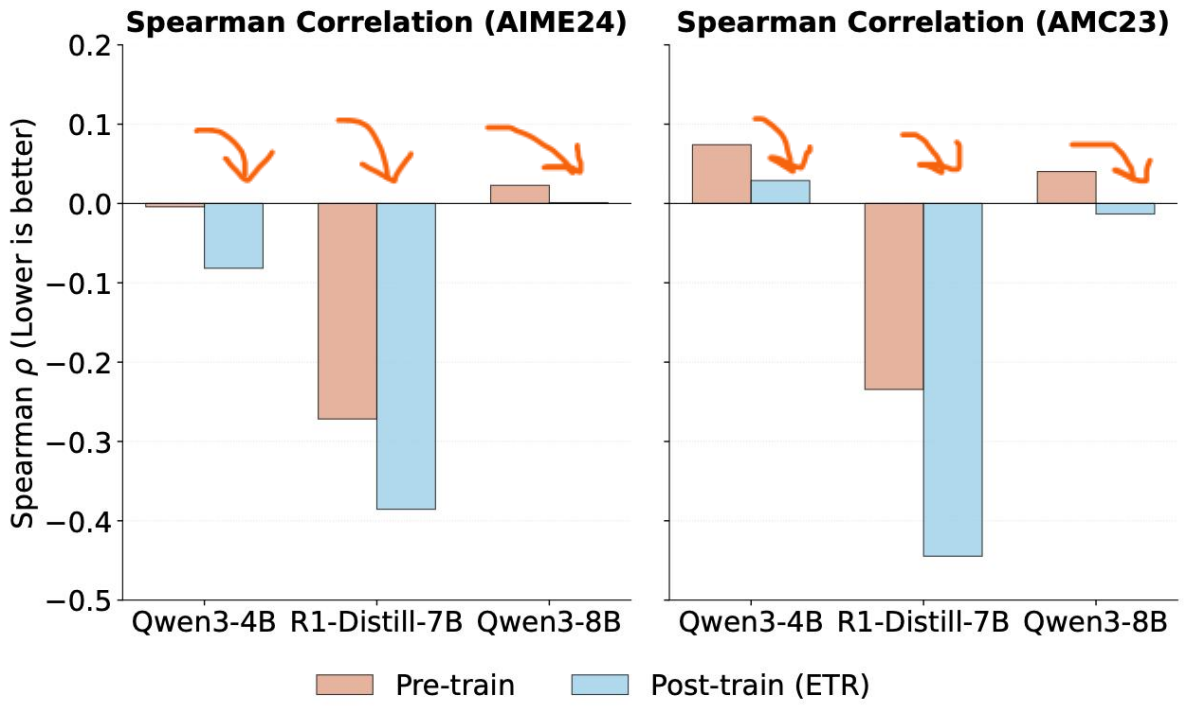


Figure 5: Spearman's rank correlation coefficient between reasoning steps and step entropy. The shift toward negative values indicates a more convergent reasoning trajectory after ETR training.

- Goal: to assess whether Entropy Trend Reward (ETR) promotes convergent reasoning.
- Method: compute the spearman rank correlation between reasoning step index and the entropy at each step, where a negative value indicates a stronger descent.

Empirical Analysis – Behavioral View of CoT Reduction

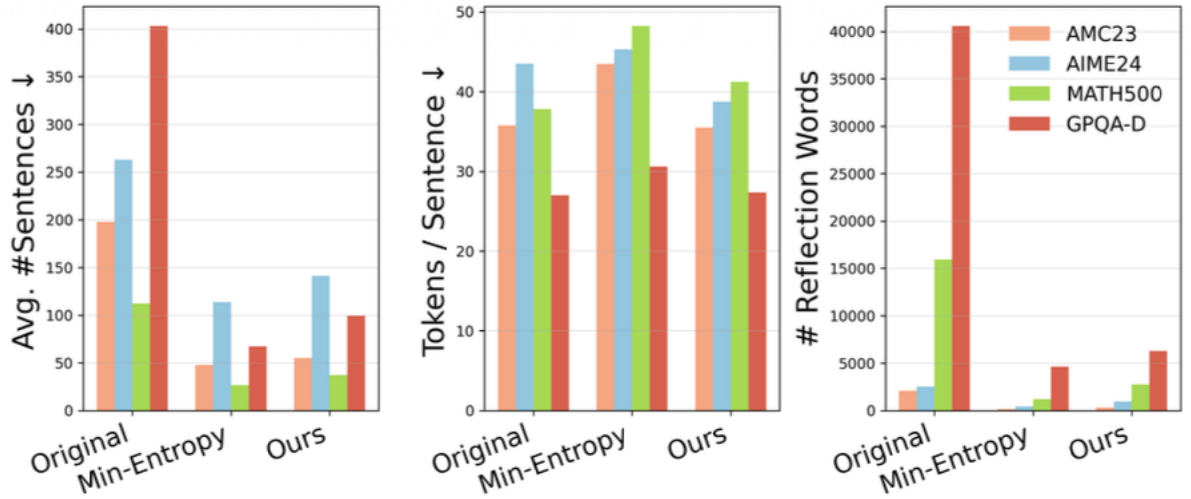


Figure 6: ETR reduces CoT length primarily by maintaining low per-step verbosity, rather than by aggressively suppressing reasoning steps or reflective processes, in contrast to entropy minimization.

Conclusion

- They show that chain-of-thought efficiency is governed by the global trend of predictive uncertainty rather than its absolute magnitude;

- Based on that insight, they propose Entropy Trend Reward, which encourages progressive uncertainty reduction while enforcing correctness as a hard constraint.

- Extensive experiments demonstrate that ETR consistently improves the accuracy efficiency trade-off.

Reason Only When Needed: Efficient Generative Reward Modeling via Model-Internal Uncertainty

Chao Xue[1,⋆], Yao Wang[1,⋆], Mengqiao Liu[2], Di Liang[2,3,†],
Xingsheng Han[2], Peiyang Liu[5], Xianjie Wu[2], Chenyao Lu[2], Lei Jiang[2],
Yu Lu[2], Haibo Shi[2,3], Shuang Liang[4], Minlong Peng[2], Flora D. Salim[1,†]

[1] University of New South Wales, Australia, [2] Tencent Hunyuan, China,
[3] Tencent Yuanbao, China, [4] UESTC, China , [5] Peking University, China

xuechao8071@gmail.com; flora.salim@unsw.edu.au

ACL 2026

Background

- Generative Reward Model is a a reward model that does not only generate a scalar as a score, but may also provide reasons, comments, and event Chain-of-Thought and then make the final judge.

- Representative work includes:

    1) LLM as a Judge;

    2) Generative verifier (reward -> next-token prediction) (previous?, no content?)

    3) Self-taught evaluators (model generate data to train itself).

    4) DeepSeek-GRM (Score based on CoT) → this paper: when is it worth to generate a CoT?

This does not sound like a reward model design, but it answers the critical question – when it is good enough to stop.

Motivation

- CoT is applied indiscriminately to all inputs regardless of their inherent complexity;

  -- Categorize into short or long reasoning paths based on the convergence of parallel decoding outputs.

- Corse Reward signals (voting-based selection, or scalar score given by GRM), lack the ability to identify and favor subtly higher-quality reasoning paths.

  -- Enable the RM to provide fine-grained quality scores by combining robust regression (Huber loss) and discriminative ranking (hinge loss).

E-GRM Framework

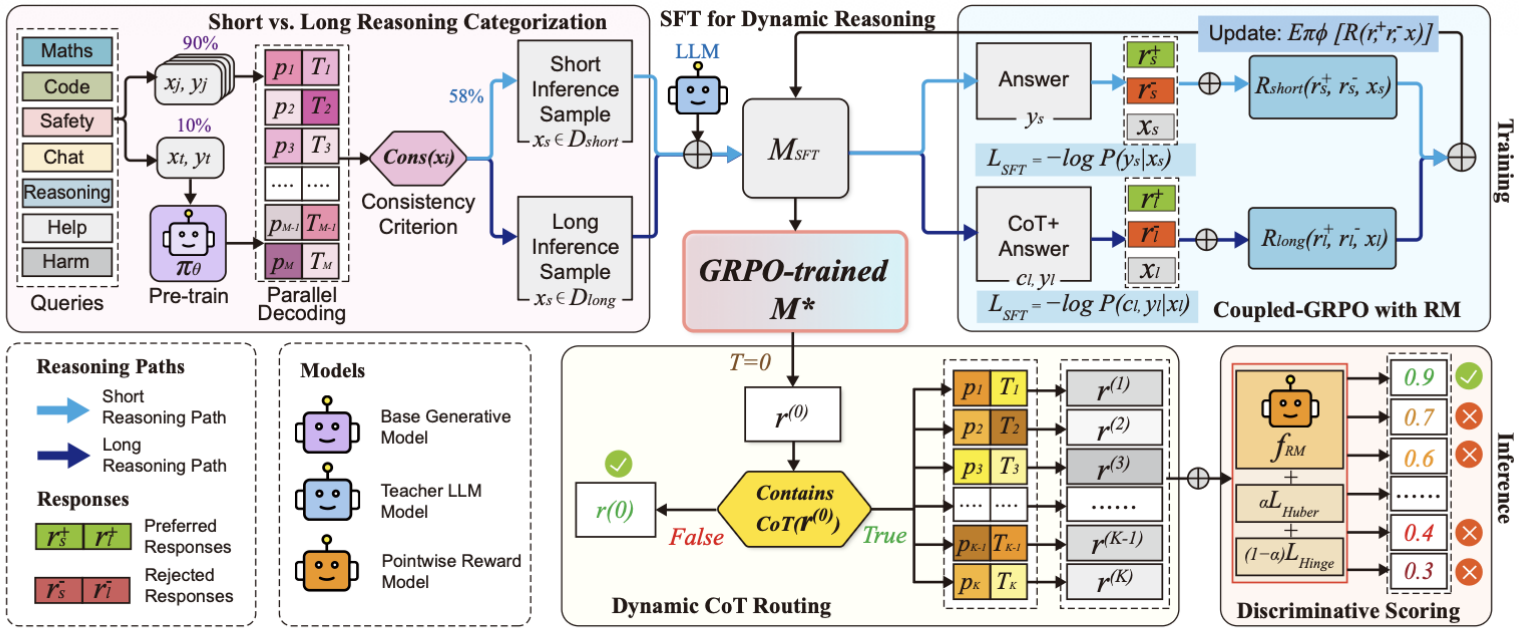


Figure 2: E-GRM is an enhanced generative framework improving efficiency and quality through dynamic CoT triggering (activating stepwise reasoning only when needed) and discriminative scoring (selecting optimal CoT via a lightweight reward model). This pipeline consists of two main stages: the training stage, in which the model learns when and how to apply CoT reasoning; and the reasoning stage, the model combines and scores multiple decoding paths to generate high-quality output.

Module 1: Dynamic CoT Triggering via Model-Internal Uncertainty

General idea: Assume prompts solvable via direct inference tend to produce consistent answers across runs.

Step 1: Given an input x, run multiple times with varied sampling hyperparameters (such as d $\{\hat{y}_i\}_{i=1}^{M}$t temperatures or top-n sampling) and yield a set of initial responses .

Step 2: Set up a threshold (m/M) to distinguish between short VS. long reasoning

Training Pipeline 1

Supervised Fine-Tuning followed by RL with Preference Optimization.

Dataset: $\mathcal{D}_{\text{SFT}} = \mathcal{D}_{\text{short}} \cup \mathcal{D}_{\text{long}}$

Combined SFT loss: $\mathcal{L}_{\text{SFT}} = \sum_{\mathcal{D}_{\text{short}}} \mathcal{L}_{\text{short}} + \sum_{\mathcal{D}_{\text{long}}} \mathcal{L}_{\text{long}}$,

where $\mathcal{L}_{\text{short}} = -\log P(y_s|x_s)$ and $\mathcal{L}_{\text{long}} = -\log P(c_l, y_l|x_l)$

Here the model learns when to generate CoT.

Module 2: Discriminative Scoring with Hybrid Loss

1) Goal: to train scoring model $\hat{q} = \mathcal{S}_\phi(x, r) \in [0, 1]$, given an input x and a candidate CoT response r.

2) Dataset: $\mathcal{D}_{\text{score}} = \{(x_i, r_i, q_i)\}$, and $q_i \in [0, 1]$ is from human annotation or a teacher model.

3) Training objective: $\mathcal{L}_{\text{scorer}} = \alpha \cdot \ell_{\text{Huber}} + (1 - \alpha) \cdot \ell_{\text{Hinge}}$

L2 loss, sensitive to outliers

, where $\ell_{\text{Huber}} = \begin{cases} \frac{1}{2}(q_i - \hat{q}_i)^2 & |q_i - \hat{q}_i| < \delta \\ \delta |q_i - \hat{q}_i| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$,

MAE, not smooth when loss is closer to 0

Module 2: Discriminative Scoring with Hybrid Loss

3) Training objective: $\mathcal{L}_{\text{scorer}} = \alpha \cdot \ell_{\text{Huber}} + (1-\alpha) \cdot \ell_{\text{Hinge}}$

, where $\ell_{\text{Hinge}} = \max(0, m - (\hat{q}_i - \hat{q}_j))$ is to enforce ranking consistency given a pair of samples (r_i, r_j) with quality scores $q_i > q_j + m$.

Penalize when $m - (\hat{q}_i - \hat{q}_j) > 0$ => 1) Ranking is wrong $\hat{q}_i - \hat{q}_j \leq 0$, or

2) Score gap is not met $0 < \hat{q}_i - \hat{q}_j < m$

Training Pipeline 2

Supervised Fine-Tuning followed by RL with Preference Optimization.

Data: $\mathcal{D}_{\text{pref}} = \{(x_i, r_i^+, r_i^-)\}$

Reward model: Adapt GRPO to 1) explicitly leverage the paired data;
2) incorporate the discriminative scorer

$$R_{\text{pair}}(x, r^+, r^-) = \mathcal{S}_\phi(x, r^+) - \mathcal{S}_\phi(x, r^-) + \beta \cdot \mathbb{I}(\text{Ans}(r^+) = y)$$

Correctness Weight

Optimize the policy parameters: $\mathcal{J}(\theta) = \mathbb{E}_{(x,r^+,r^-)\sim\mathcal{D}_{\text{pref}}}\left[R_{\text{pair}}(x, r^+, r^-)\right] - \lambda \cdot \mathbb{D}_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}]$

Training Pipeline 2

Change:

1) Group Advantage -> Preference Learning;

2) A new term to enhance the gap between positive and negative reasonings.

Standard GRPO

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim P(Q) \\ \{o_i\}\sim\pi_{\theta_{\text{old}}}}} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \Big\{ \min\Big[r_{i,t}\hat{A}_{i,t}, \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\Big] - \beta\,\mathbb{D}_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}]\Big\}$$

$$\text{where}\quad r_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$$

E-GRM Preference Optimization

$$\mathcal{J}_{\text{E-GRM}}(\theta) = \mathbb{E}_{(x,r^+,r^-)\sim\mathcal{D}_{\text{pref}}}\Bigg[\frac{1}{2}\sum_{\kappa\in\{+,-\}}\frac{1}{|r^\kappa|}\sum_{t=1}^{|r^\kappa|} \min\Big[\tilde{r}_t^\kappa\hat{A}_t^\kappa, \text{clip}(\tilde{r}_t^\kappa, 1-\epsilon, 1+\epsilon)\hat{A}_t^\kappa\Big] + \gamma\cdot[\mathcal{S}_\phi(x,r^+) - \mathcal{S}_\phi(x,r^-)] - \beta\,\mathbb{D}_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}]\Bigg]$$

$$\text{where}\quad \tilde{r}_t^\kappa = \frac{\pi_\theta(r_t^\kappa|x,r_{<t}^\kappa)}{\pi_{\theta_{\text{old}}}(r_t^\kappa|x,r_{<t}^\kappa)},\quad \kappa\in\{+,-\}$$

Figure 3: Comparison of standard GRPO and our extended formulation used in E-GRM. While both leverage group-based relative optimization, E-GRM explicitly incorporates paired preference signals through the discriminative scorer $\mathcal{S}_\phi$, enhancing alignment with nuanced response quality distinctions.

Inference Procedure

Goal: Reduce computational overhead while ensuring qualified outputs.

1) For input x, perform M parallel decodings to obtain initial responses $\{\hat{y}_i\}_{i=1}^{M}$
2) Dynamic Routing: If $\text{Consensus}(x) = \frac{\max_y \text{Count}(y)}{M} \geq \tau$, output the consensus answer and terminate;
3) CoT generation and selection: If consensus is low, generate K diverse CoT responses $\{r_k\}_{k=1}^{K}$ using varied decoding parameters;
4) Apply the discriminator scorer to each candidate;
5) Select the response with the highest score

Experiments – Reward Model Performance 1

Models	Chat	Math	Code	Safety	Easy	Normal	Hard	Avg
Scalar RMs								
steerlm-70b	56.4	53.0	49.3	51.2	48.3	54.9	54.3	52.5
tulu-v2.5-70b-preference-mix-rm	58.2	51.4	55.5	87.1	72.8	65.6	50.7	63.0
Mistral-7B-instruct-Unified-Feedback	56.5	58.0	51.7	86.8	87.1	67.3	35.3	63.2
RM-Mistral-7B	57.4	57.0	52.7	87.2	88.6	67.1	34.9	63.5
Eurus-RM-7b	59.9	60.2	56.9	86.5	87.2	70.2	40.2	65.9
internlm2-7b-reward	61.7	71.4	49.7	85.5	85.4	70.7	45.1	67.1
GRM-llama3-8B-sftreg	62.7	62.5	57.8	90.0	83.5	72.7	48.6	68.2
internlm2-20b-reward	63.1	66.8	56.7	86.5	82.6	71.6	50.7	68.3
Llama-3-OffsetBias-RM-8B	71.3	61.9	53.2	89.6	84.6	72.2	50.2	69.0
Nemotron-340B-Reward	71.2	59.8	59.4	87.5	81.0	71.4	56.1	69.5
URM-LLaMa-3.1-8B	71.2	61.8	54.1	93.1	84.0	73.2	53.0	70.0
Skywork-Reward-Llama-3.1-8B	69.5	60.6	54.5	**95.7**	**89.0**	74.7	46.6	70.1
GenRMs								
tulu-v2.5-dpo-13b-chatbot-arena-2023	64.9	52.3	50.5	62.3	82.8	60.2	29.5	57.5
tulu-v2.5-dpo-13b-nectar-60k	56.3	52.4	52.6	73.8	86.7	64.3	25.4	58.8
stablelm-2-12b-chat	67.2	54.9	51.6	65.2	69.1	63.5	46.6	59.7
tulu-v2.5-dpo-13b-stackexchange-60k	66.4	49.9	54.2	69.0	79.5	63.0	37.2	59.9
Nous-Hermes-2-Mistral-7B-DPO	58.8	55.6	51.3	73.9	69.5	61.1	49.1	59.9
tulu-v2.5-dpo-13b-hh-rlhf-60k	68.4	51.1	52.3	76.5	53.6	63.0	69.6	62.1
tulu-2-dpo-13b	66.4	51.4	51.8	85.4	86.9	66.7	37.7	63.8
ReasonRMs								
Qwen-Instruct-7B-Ours	66.9	66.8	54.4	92.9	79.5	71.3	59.5	70.1
Qwen-Instruct-14B-Ours	75.3	75.6	60.9	93.3	82.9	77.7	68.5	76.4
Qwen-Instruct-32B-Ours	**75.6**	**80.0**	**66.5**	94.2	86.0	**80.8**	**70.7**	**79.2**

Table 1: RM-Bench evaluation across domains and complexity levels. The proposed **Qwen-Instruct-*B-Ours** models exhibit excellent capabilities in multiple categories, reaching a top average performance of 79.2% while excelling in math, chat, code, and challenging tasks. **Bold** denotes top performance. Underlined denotes runner-up.

Evaluate 1) sensitivity to subtle content differences (subtle factual or logical mistakes); 2) resistance to style biases

Experiments - Reward Model Performance 2

Models	Helpfulness		Harmlessness		Overall
	BoN	Pairwise	BoN	Pairwise	
Scalar RMs					
Tulu-v2.5-13b-preference-mix-rm	0.355	0.562	0.351	0.545	0.453
Skywork-Reward-Gemma-2-27B	0.472	0.653	0.561	0.721	0.602
Internlm2-20b-reward	0.585	0.763	0.499	0.670	0.629
ArmoRM-Llama3-8B-v0.1	0.636	0.787	0.497	0.663	0.646
Internlm2-7b-reward	0.626	0.782	0.563	0.712	0.671
Eurus-RM-7b	0.679	0.818	0.543	0.693	0.683
Skywork-Reward-Llama-3.1-8B	0.627	0.781	0.603	0.759	0.693
Starling-RM-34B	0.604	0.774	0.674	0.795	0.712
GenRMs					
Llama2-70b-chat	0.289	0.613	0.249	0.602	0.438
Llama3.1-8B-Instruct	0.365	0.675	0.267	0.653	0.490
Gemini-1.5-pro	0.536	0.763	0.299	0.661	0.565
Mixtral-8x7B-Instruct-v0.1	0.480	0.706	0.491	0.671	0.587
skywork-critic-llama3.1-8B	0.600	0.725	0.578	0.578	0.620
skywork-critic-llama3.1-70B	0.640	0.753	0.614	0.614	0.655
Llama3.1-70B-Instruct	0.648	0.811	0.558	0.739	0.689
Mistral-Large-2407	0.678	0.817	0.583	0.725	0.701
Claude-3-5-sonnet	**0.705**	**0.838**	0.518	0.764	0.706
Qwen2-72B-Instruct	0.645	0.810	0.649	0.789	0.723
GPT-4o-2024-05-13	0.639	0.815	0.682	0.814	0.738
ReasonRMs					
Deepseek-GRM-27B-RFT	0.592	0.801	0.548	0.765	0.670
Deepseek-GRM-27B	0.623	0.805	0.570	0.761	0.690
Base-Qwen-Instruct-7B (Ours)	0.553	0.756	0.625	0.775	0.677
Base-Qwen-Instruct-14B (Ours)	0.605	0.791	0.635	0.793	0.706
Base-Qwen-Instruct-32B (Ours)	0.647	0.807	**0.696**	**0.823**	**0.743**

Table 2: RMB evaluation results. **Bold** denotes top performance. Underlined denotes runner-up performance.

Evaluate alignment in practical scenarios (emails, coding, advising..)

Experiments - Reward Model Performance 3

Models	Chat	Chat_Hard	Safety	Reasoning	Overall
Scalar RMs					
SteerLM-RM 70B	91.3	80.3	**92.8**	90.6	88.8
Cohere-0514	96.4	71.3	<u>92.3</u>	**97.7**	89.4
GenRMs					
Llama3.1-8B-Instruct	85.5	48.5	75.6	72.1	70.4
Prometheus-8*7B-v2	93.0	47.1	80.5	77.4	74.5
Llama3.1-70B-Instruct	**97.2**	70.2	82.8	86.0	84.0
Llama3.1-405B-Instruct	**97.2**	74.6	77.6	87.1	84.1
Claude-3-5-sonnet-20240620	96.4	74.0	81.6	84.7	84.2
GPT-4o-0806	96.1	76.1	86.6	88.1	86.7
Gemini-1.5-pro-0514	92.3	80.6	87.9	92.0	88.2
Self-taught-evaluator-llama3.1-70B	96.9	**85.1**	89.6	88.4	<u>90.0</u>
ReasonRMs					
SynRM	38.0	82.5	74.1	87.1	70.4
CLoud	<u>97.0</u>	58.0	84.0	92.0	82.8
DeepSeek-GRM-16B	90.8	74.3	84.7	81.8	82.9
DeepSeek-GRM-27B	94.1	78.3	88.0	83.8	86.0
Qwen-Instruct-7B-Ours	94.2	74.8	85.3	87.0	85.3
Qwen-Instruct-14B-Ours	93.8	80.6	87.2	92.1	88.4
Qwen-Instruct-32B-Ours	95.4	<u>83.3</u>	92.0	<u>95.4</u>	**91.5**

Table 3: Performance of various models on the RewardBench benchmark. **Qwen-Instruct-*B-Ours** consistently outperforms baseline methods across RewardBench evaluations. <u>Underlined</u> denotes runner-up performance.

Experiments – Comparison of CoT Triggering Strategies

Method	Accuracy	Avg. Latency	Task-specific Heuristic?
Forced-CoT	75.1	3.8	No
Rule-based	70.5	2.1	Yes
AdaCoT	76.8	2.9	Yes
E-GRM (Ours)	**78.4**	**2.2**	**No**

Table 4: Comparison of CoT triggering strategies. Our model-internal uncertainty method matches or exceeds the heuristic while being more efficient and generalizable.

Experiments – Ablation Study

Variant	MATH	HelpSteer2	RMB Harmlessness
E-GRM (Std. GRPO)	76.9	81.5	0.765
E-GRM (Extended)	**78.4**	**82.3**	**0.775**

Table 5: Ablation on preference optimization. Our GRPO formulation provides a consistent boost, demonstrating its utility for paired preference data.

Variant	Acc. (%)	FLOPs (T)	Latency (s)
Full E-GRM	**78.4**	**15.7**	**2.2**
w/o Dynamic CoT	75.2	23.4	3.4
w/o Discrim. Scoring	72.8	15.9	2.2
Base CoT-GRM	69.1	23.7	3.6

Table 6: Ablation study on the MATH dataset. Removing dynamic CoT increases cost significantly, while removing scoring causes the largest accuracy drop.

Purging the Gray Zone: Latent-Geometric Denoising for Precise Knowledge Boundary Awareness

Hao An[*], **Yibin Lou**[*], **Jiayi Guo**, **Yang Xu**[†]
Computational Linguistics and Consciousness Sciences Lab
Southern University of Science and Technology

ACL 2026 Findings

Motivation

- LLMs often exhibit hallucinations due to inability to accurately perceive their knowledge boundaries.

- Existing abstention fine-tuning methods typically partition datasets directly based on response accuracy.

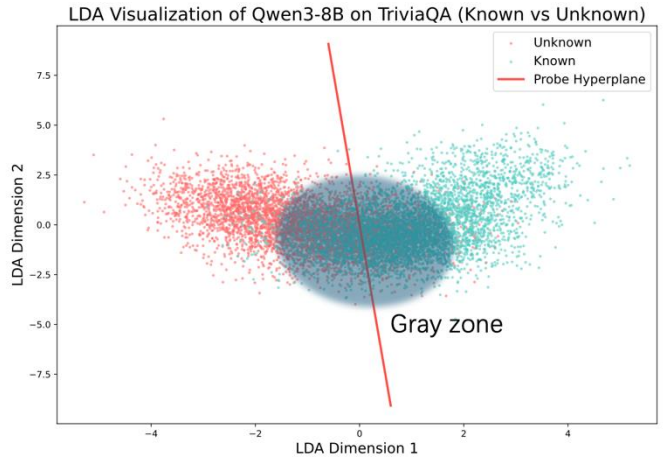


Figure 1: Visualization of the hidden states of questions that are known and unknown to the model. "Gray zone" refers to the overlapping area.

Limitation 1: Grey Zone -- lucky guess

Some known questions are geometrically closed to the unknown questions in latent space, along the hyperplane.

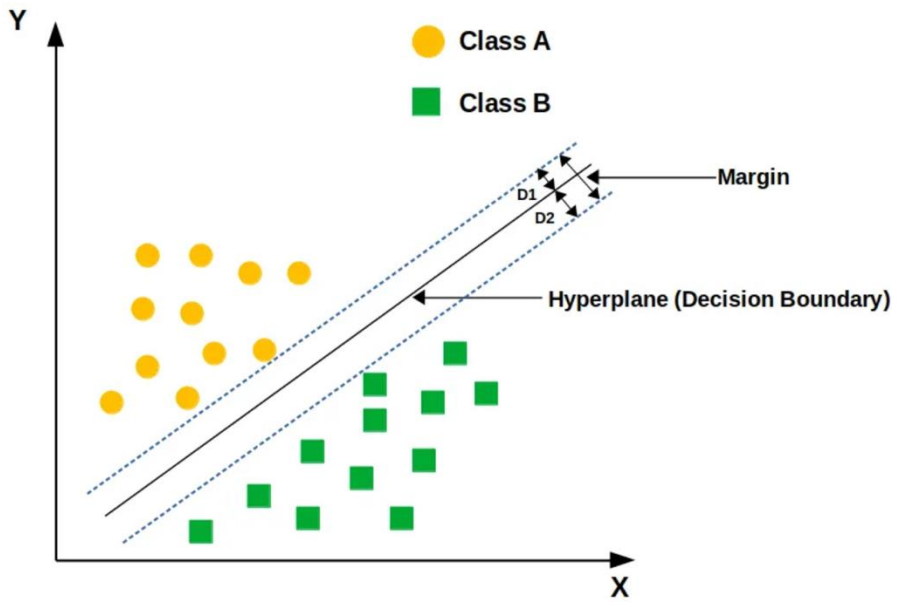
Motivation

- To reduce reliance on ground truth labels, alternative approaches utilize uncertainty metrics like semantic entropy to partition training data into known and unknown questions.

Limitation 2: Internal confidence and external correctness may sometimes misalign.

Geometric Denoising

The general idea is quite similar to Support Vector Machine.



SVM

The proposed Geometric Denoising (GeoDe) framework uses geometric distance from the hyperplane as a confidence metric.

Geometric Denoising

- Goal: Develop a hidden-state-based approach to enhance the model's awareness of its own knowledge boundaries.

- Overall, it 1) partition the training data into known / unknown subsets;

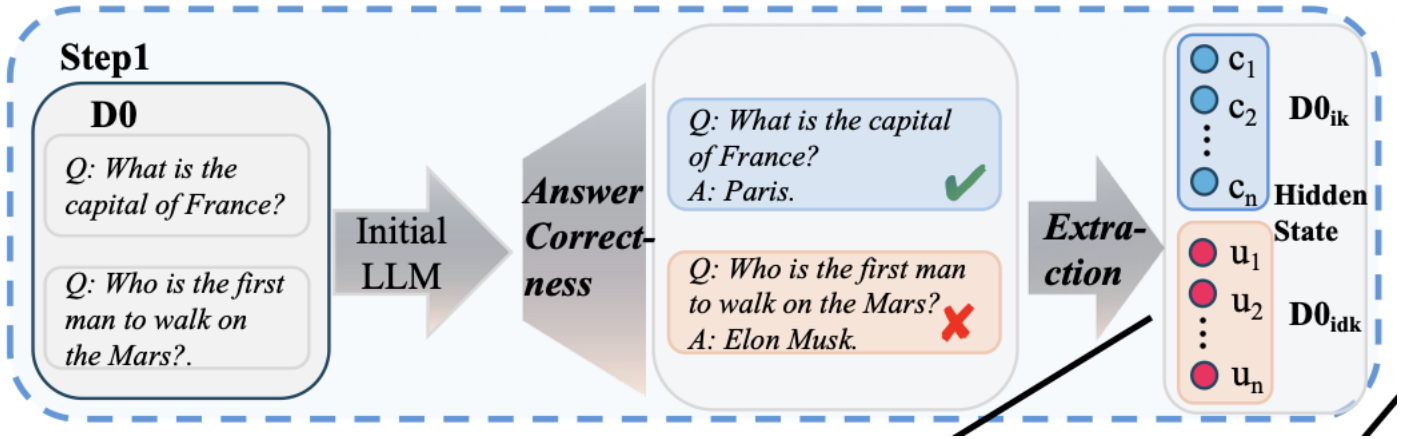
2) train a hidden-state probe to quantify the model's confidence in its responses;

3) Finally perform targeted abstention fine-tuning on the curated samples to teach the model to abstain from answering questions outside its knowledge scope.

Geometric Denoising

Step 1 -- Identify the Knowledge Boundary:

The base LLM is prompted to answer every question in a source training dataset ($D0$) and split to known ($D0_{ik}$) and unknown knowledge ($D0_{idk}$) according to correctness;



Geometric Denoising

Step 1 -- Identify the Knowledge Boundary:

The base LLM is prompted to answer every question in a source training dataset ($D0$) and split to known ($D0_{ik}$) and unknown knowledge ($D0_{idk}$) according to correctness;

Step 2 – Curate Denoised Dataset:

1) Train the probe model (Linear Regression):

$f_{\text{probe}}(\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b)$, where the feature is the hidden state representation of the question $\mathbf{x} = f_{\text{LLM}}(q)$.

Geometric Denoising

Step 2 – Curate Denoised Dataset:

2) Next, define the confidence of LLM answering question by measuring the distance from x to the learned hyperplane (w; b):

$$d(\mathbf{x}) = \frac{\left|\mathbf{w}^\top \mathbf{x} + b\right|}{\|\mathbf{w}\|_2}$$

3) Select top x% samples (big d(x)).

Geometric Denoising

Step 3 – Abstention Fine-Tuning.

Split $D1^{\text{selected}}$ into $D1^{\text{selected}}_{\text{ik}}$ $(d(x) > 0)$ and $D1^{\text{selected}}_{\text{idk}}$ $(d(x) < 0)$.

Replace ground truth of $D1^{\text{selected}}_{\text{idk}}$ with "I don't know."

Fine-tune M with cross-entropy loss on $D1^{\text{selected}}$

Return $M_{\text{fine-tuned}}$

Experiments – Evaluation Metrics for Abstention

GT \ R	Correctly answered	Wrongly answered	Abstained
Known	N_1	N_2	N_3
Unknown	–	N_4	N_5

Table 1: Abstention confusion matrix. R denotes the answer of the refined (fine-tuned) model. GT denotes the initial model. "Known" denotes that the initial model answered correctly, while "unknown" indicates that the initial model answered wrongly.

Helpfulness ($\mathbf{F1_{ans}}$): For known questions, we calculate $F1_{ans}$ (Kim et al., 2024) as the harmonic mean of answerable recall ($\frac{N_1}{N_1+N_2+N_3}$) and answerable precision ($\frac{N_1}{N_1+N_2+N_4}$).

Truthfulness ($\mathbf{F1_{abs}}$): For unknown questions, we calculate $F1_{abs}$ (Kim et al., 2024) as the harmonic mean of unanswerable recall ($\frac{N_5}{N_4+N_5}$) and unanswerable precision ($\frac{N_5}{N_3+N_5}$).

Reliability ($\mathbf{F1_{rel}}$): Existing studies indicate that enhancing helpfulness leads to a decline in factuality (Xu et al., 2024; An and Xu, 2025). Therefore, we calculate the harmonic mean of metrics $F1_{ans}$ and $F1_{abs}$ as a reliability metric for comprehensive evaluation (An and Xu, 2025).

Experiments – Overall Performance

Different latent Representation:

TBG – Q

SLT – Q+A

Demonstrates robustness regardless
Of specific extraction point.

Dataset	TriviaQA			NQ			SciQ			SimpleQA		
Method	$F1_{ans}$	$F1_{abs}$	$F1_{rel}$	$F1_{ans}$	$F1_{abs}$	$F1_{rel}$	$F1_{ans}$	$F1_{abs}$	$F1_{rel}$	$F1_{ans}$	$F1_{abs}$	$F1_{rel}$
Llama3-8B-Instruct												
IDK	78.7	35.3	48.8	**61.4**	56.3	58.7	**83.3**	5.6	10.4	14.6	64.0	23.8
Uncertainty	67.7	46.4	55.0	45.9	18.0	25.9	64.7	17.9	28.0	10.0	52.2	16.8
R-Tuning	77.3	71.6	74.4	47.0	78.1	58.7	69.9	58.2	63.5	14.6	**96.1**	25.4
Probe-Tuning TBG	78.8	72.8	75.7	51.6	79.1	62.5	81.4	53.4	64.5	12.8	**96.1**	22.6
GeoDe TBG	**80.9**	**73.7**	**77.1**	54.0	78.4	**64.0**	81.9	58.5	68.3	18.4	94.8	30.7
Probe-Tuning SLT	75.8	71.3	73.4	44.8	78.3	56.9	75.4	58.9	66.1	18.4	95.8	30.9
GeoDe SLT	79.8	**73.7**	76.7	52.6	**79.4**	63.3	82.4	**59.8**	**69.3**	**18.6**	96.0	**31.2**
Qwen3-8B												
IDK	74.0	55.1	63.2	55.1	65.4	59.8	81.8	44.5	57.6	11.7	58.6	19.5
Uncertainty	75.9	38.8	51.3	**57.3**	52.1	54.6	70.3	38.6	49.8	12.2	66.6	20.6
R-Tuning	75.8	74.3	75.0	50.3	70.5	58.7	**86.5**	45.1	59.3	12.6	63.6	21.0
Probe-Tuning TBG	75.0	71.7	73.3	51.2	70.5	59.4	**86.5**	44.2	58.5	12.6	63.4	21.0
GeoDe TBG	**77.7**	**77.4**	**77.6**	54.3	71.3	**61.6**	81.8	47.2	59.8	13.0	**67.3**	21.7
Probe-Tuning SLT	75.4	72.8	74.1	50.1	69.2	58.1	85.7	45.8	60.0	12.3	62.0	20.6
GeoDe SLT	75.6	75.6	75.6	51.6	**71.7**	60.0	84.6	**48.5**	**61.6**	**13.8**	65.9	**22.8**

Table 2: Performance on in-domain and out-of-domain question answering benchmarks. All results are multiplied by 100. The best result is bolded. The second best is underlined.

Experiments – In RAG Setting

	Golden					Golden & RRN				
	$F1_{ans}$	$F1_{abs}$	$F1_{rel}$	Acc.	Hallu.	$F1_{ans}$	$F1_{abs}$	$F1_{rel}$	Acc.	Hallu.
ICL	-	-	-	71.2	28.8	-	-	-	63.8	36.2
IDK	76.1	45.5	57.0	58.1	17.9	71.6	45.6	55.7	52.7	23.6
R-Tuning	80.8	52.5	63.7	61.0	11.8	74.4	53.3	62.1	50.4	15.3
Probe-tuning TBG	83.8	47.9	61.0	67.3	13.0	81.1	48.6	60.8	63.3	16.8
GeoDe TBG	75.2	52.5	61.8	51.5	9.2	74.3	56.9	64.5	49.1	11.4
Probe-Tuning SLT	79.1	52.7	63.2	54.8	11.9	75.9	54.7	63.5	50.0	15.7
GeoDe SLT	79.9	61.1	69.3	55.8	9.2	75.8	57.9	65.7	53.4	13.0

Table 3: RAG results. Acc. is accuracy. Hallu. is hallucination rate.

Analysis -- hypothesis validation

Dataset	TriviaQA			NQ		
Metric	$F1_{ans}$	$F1_{abs}$	$F1_{rel}$	$F1_{ans}$	$F1_{abs}$	$F1_{rel}$
Ours TBG	**77.7**	**77.4**	**77.6**	**54.3**	71.3	**61.6**
-middle	74.9	74.2	74.6	50.0	70.2	58.4
-nearest	75.2	71.2	73.2	49.7	69.6	58.0
Ours SLT	75.6	75.6	75.6	51.6	71.7	60.0
-middle	75.4	72.8	74.1	50.1	69.2	58.1
-nearest	73.8	67.5	70.5	48.9	67.2	56.6
R-Tuning-01	75.5	77.0	76.2	51.6	**71.8**	60.0

Table 5: Ablation study on distance-based data partitioning for abstention fine-tuning. Samples are partitioned into three tiers based on their geometric distance $|d(x)|$ to the probing hyperplane.

To validate the hypothesis that the geometric distance to the Probing hyperplane serves as a proxy for sample quality, they Partition the training data into three tiers Ours-Farthest (75%-100%), Ours-Middle (37.5%-62.5%), Ours-Nearest (0-25%).