CONTENTS

A Additional Preliminary

Group Sequential Policy Optimization. Recently, Zheng et al. (2025) introduce group sequence policy optimization (GSPO), a new reinforcement learning algorithm for training large language models. Following the basic principle of importance sampling, GSPO defines importance ratios based on sequence likelihood and performs sequence-level clipping, rewarding, and optimization. The GSPO objective $\mathcal{J}_{\text{GSPO}}(\theta)$ is then defined as:

$$\mathop{\mathbb{E}}_{\substack{(\boldsymbol{x},\boldsymbol{a})\sim\mathcal{D} \\ \{\boldsymbol{y}_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|\boldsymbol{x})}} \Big[\frac{1}{\sum_{i=1}^{G}}\sum_{i=1}^{G}\min\left(s_i(\theta)\hat{A}_{i,k}, \hat{A}_{i,k}\cdot\text{clip}\left(s_i(\theta), 1-\varepsilon, 1+\varepsilon\right)\right)\Big] \tag{7}$$

where the defined the importance ratio $s_i(\theta)$ is based on sequential likelihood:

$$s_i(\theta) = \left(\frac{\pi_\theta(\boldsymbol{y}_i|\boldsymbol{x})}{\pi_{\theta_{\text{old}}}(\boldsymbol{y}_i|\boldsymbol{x})}\right)^{\frac{1}{|\boldsymbol{y}_i|}} = \exp\left(\frac{1}{|\boldsymbol{y}_i|}\sum_{k=1}^{|\boldsymbol{y}_i|}\gamma_{i,k}(\theta)\right) \tag{8}$$

The token-level objective variant of GSPO, namely $\mathcal{J}_{\text{GSPO-token}}(\theta)$ allows token-wise advantage customization and is defined as:

$$\mathop{\mathbb{E}}_{\substack{(\boldsymbol{x},\boldsymbol{a})\sim\mathcal{D} \\ \{\boldsymbol{y}_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|\boldsymbol{x})}} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{k=1}^{|y_i|}\min\left(s_{i,k}(\theta)\hat{A}_{i,k},\ \text{clip}(s_{i,k}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,k}\right)\right], \tag{9}$$

where

$$s_{i,k}(\theta) = \text{sg}[s_i(\theta)]\cdot\frac{\pi_\theta(\boldsymbol{y}_{i,k}|\boldsymbol{x}, \boldsymbol{y}_{i,<k})}{\text{sg}[\pi_\theta(\boldsymbol{y}_{i,k}|\boldsymbol{x}, \boldsymbol{y}_{i,<k})]}, \tag{10}$$

and $\text{sg}[\cdot]$ denotes only taking the numerical value but stopping the gradient, corresponding to the `detach` operation in PyTorch. The gradient of GSPO-token can be derived as:

GSPO demonstrates notably superior training stability, efficiency, and performance compared to GRPO and exhibits particular efficacy for the large-scale RL training of MoE models. To be specific,

B Additional Experiment Details.

Additional Details for Qwen2.5-0.5B-Ins: For the 0.5B model, training is conducted on two A6000 GPUs with a batch size of 32, a maximum rollout length of 2500 tokens, a learning rate of $5e^{-7}$, and a mini-batch size of 16—resulting in two iteration updates per training step. For the greedy decoding performance, we report the best accuracy across multiple checkpoints due to significant fluctuations during training. For all other settings, we report the performance at the final checkpoint. In addition to high-THR tokens, we also include those within the top 20% highest-entropy tokens that do not overlap with high-THR (approximate 4.1 % tokens), and keep their advantage unchanged being $\hat{A}_{i,k}$. For formatting, we follow Zeng et al. (2025), adopting simple prompts since the model struggles with complex instructions. We use $p = 0.2$ and $p = -0.2$ for exploitation and exploration respectively.

Additional Details for Qwen-Math: The Qwen-Math model Yang et al. (2024) uses its full context length of 3072 tokens for rollouts. For format, we folow Zeng et al. (2025) to use Qwen Chat template and require final answer to be enclosed in a latex command `\boxed{}`. Unless otherwise specified, we set $p = 0.1$ for exploitation and $p = -0.1$ for exploration.

Additional Training Details for Llama: For the Llama3.2-3B-Instruct Dubey et al. (2024) model, training is carried out on 8 A100 GPUs with a batch size of 256, a maximum rollout length of 3000 tokens, a learning rate of 1×10^{-6}, and a mini-batch size of 16. For greedy decoding, we report the best accuracy across multiple checkpoints due to the substantial fluctuations observed during training, while for all other settings we report results from the final checkpoint. In addition to high-THR tokens, we also include those within the top 20% highest-entropy tokens that do not overlap with

Base Model	Method	AIME25	AIME24	AMC23	MATH500	Minerva	Olympiad	Avg.
Qwen2.5-Math-1.5B	Base	0.0	3.3	20.0	39.6	7.7	24.9	15.9
	GRPO	3.3	13.3	57.5	**71.8**	29.0	34.1	34.8
	Pos Only	3.3	10.0	57.5	70.6	30.1	31.0	33.8
	THR ($p = 0.1$)	3.3	13.3	**62.5**	<u>71.4</u>	**33.1**	**34.5**	**36.3**

Table 5: Exploitation Results. Pass@1 accuracy (%) using greedy decoding across different methods and datasets. **Bold** indicates the best performance, while <u>underline</u> marks the second-best.

high-THR (approximate 3.5 % tokens), and fix their keep their advantage unchanged being $\hat{A}_{i,k}$. For formatting, we follow Zeng et al. (2025), adopting simple prompts since the model struggles with complex instructions.

C Additional Experiments

C.1 Ablation Study on Positive and Negative-Only Training.

We further investigate the impact of training with only positive or negative tokens by modifying $\hat{A}_{i,k}$. In the "Pos Only" setting, we set all values where $\hat{A}_{i,k} < 0$ to 0, thereby increasing the confidence of correct responses only. Conversely, in the "Neg Only" setting, we set all values where $\hat{A}_{i,k} > 0$ to 0, which reduces the confidence of incorrect responses without reinforcing correct ones. As shown in Table 5, "Pos Only" results in a 1.3% drop in average performance compared to GRPO, indicating that negative gradients also contribute to boosting confidence in correct responses.

Method	Qwen2.5-0.5B-Instruct Pass@K									Qwen2.5-Math-1.5B Pass@K								
	1	2	4	8	16	32	64	128	256	1	2	4	8	16	32	64	128	256
AIME 2025																		
GRPO	0.2	**0.4**	0.6	1.2	2.5	4.8	9.2	17.1	30.0	5.9	9.9	15.0	20.5	26.5	33.6	41.5	49.8	56.7
Neg Only	**0.2**	**0.4**	**0.7**	**1.4**	**2.8**	**5.3**	**9.5**	16.2	26.7	4.7	8.1	12.7	17.8	23.4	30.2	38.2	46.2	56.7
THR ($p < 0$)	**0.2**	0.3	0.6	1.1	2.3	4.6	9.0	**17.5**	**33.3**	**6.0**	**10.1**	**15.3**	**20.9**	**26.8**	**33.9**	**41.7**	**50.0**	**60.0**
AIME 2024																		
GRPO	**0.4**	**0.8**	**1.5**	**2.9**	**5.4**	**10.0**	**17.2**	**27.3**	**36.7**	11.4	17.7	24.3	30.5	36.7	43.4	50.0	56.0	63.3
Neg Only	0.2	0.5	0.9	1.8	3.3	5.9	9.7	14.9	23.3	9.9	16.0	23.1	30.2	36.7	42.8	48.1	52.9	56.7
THR ($p < 0$)	**0.4**	**0.8**	**1.5**	**2.9**	**5.4**	9.4	14.9	21.5	30.0	**11.9**	**18.2**	**24.9**	**31.2**	**37.9**	**45.3**	**52.9**	**61.2**	**70.0**
AMC23																		
GRPO	11.4	18.7	28.3	39.7	52.3	64.5	74.9	81.8	85.0	46.6	59.1	70.0	78.9	85.5	90.2	93.7	96.0	97.5
Neg Only	7.7	13.7	22.6	34.4	48.4	63.2	76.6	87.5	**95.0**	44.0	56.9	68.0	76.5	83.0	88.5	92.3	94.3	95.0
THR ($p < 0$)	**12.0**	**20.1**	**30.6**	**42.7**	**56.5**	**70.8**	**82.7**	**89.6**	92.5	**47.9**	**61.0**	**72.2**	**81.1**	**87.3**	**91.6**	**95.1**	**98.0**	**100.0**
Average																		
GRPO	4.0	6.6	10.1	14.6	20.1	26.4	33.8	42.1	50.6	21.3	28.9	36.4	43.3	49.6	55.7	61.7	67.3	72.5
Neg Only	2.7	4.9	8.1	12.5	18.2	24.8	31.9	39.5	48.3	9.5	27.0	34.6	41.5	47.7	53.8	59.5	64.5	68.4
THR ($p < 0$)	**4.9**	**7.4**	**11.6**	**15.6**	**21.4**	**28.3**	**35.5**	**43.5**	**51.9**	**21.9**	**29.8**	**37.5**	**44.4**	**50.7**	**57.3**	**63.2**	**69.7**	**76.7**

Table 6: Comparing exploration ability with Pass@K. Results for Qwen2.5-Math-1.5B and Qwen2.5-Math-7B are reported on the AIME 2024, AIME 2025, and AMC23 datasets, along with their average. **Bold** indicates the best performance.

As also shown in Table 6, "Neg Only" underperforms in most cases. For example, on AMC23 with Qwen2.5-Math-1.5B, it achieves a Pass@256 of 56.7%, compared to 63.3% for both GRPO and vanilla THR. While "Neg Only" yields moderate improvements over the Base model on average—indicating that suppressing incorrect responses provides some exploratory value—positive tokens still play a critical role in enhancing exploration. By selectively incorporating informative tokens, THR with $p < 0$ achieves substantially better exploration performance than "Neg Only" alone.

C.2 Additional Results on GSPO

We further show that THR can be seamlessly integrated with other group relative reinforcement learning objectives. In particular, we apply THR to token level variant of group sequence policy optimization (GSPO-token) Zheng et al. (2025), which optimizes at the sequence level through clipping, rewarding, and optimization while allow token level advantage adjustment (more details in Appendix Appendix A). As reported in Table 7, incorporating THR with $p < 0$ yields substantial improvements, boosting Pass@K performance across all K with an average improvement by around 0.9% to THR and 1.4% to GSPO.

Method	Qwen2.5-Math-1.5B Pass@K								
	1	2	4	8	16	32	64	128	256
AIME 2025									
GSPO	5.2	9.0	13.9	19.3	24.9	31.0	36.9	41.4	46.7
GSPO+THR	4.4	7.8	12.5	18.0	23.9	31.1	39.0	46.4	50.0
GSPO+THR ($p = -0.1$)	5.1	8.9	14.3	20.4	26.6	33.3	39.9	46.9	53.3
AIME 2024									
GSPO	10.4	16.8	24.1	31.3	38.5	45.6	52.4	59.4	66.7
GSPO+THR	10.0	16.2	23.6	30.8	37.7	44.8	52.8	60.8	66.7
GSPO+THR ($p = -0.1$)	11.0	17.2	24.2	31.0	37.8	44.9	51.8	59.1	66.7
AMC 2023									
GSPO	44.9	58.0	69.0	77.7	84.3	89.1	92.0	93.6	95.0
GSPO+THR	44.9	58.0	68.7	77.0	83.5	88.8	93.3	97.2	100.0
GSPO+THR ($p = -0.1$)	45.4	58.2	69.1	77.9	84.6	90.1	95.0	98.7	100.0
Average									
GSPO	20.2	27.9	35.7	42.8	49.2	55.2	60.4	64.8	69.5
GSPO+THR	19.8	27.3	34.9	41.9	48.4	54.9	61.7	68.1	72.2
GSPO+THR ($p = -0.1$)	**20.5**	**28.1**	**35.9**	**43.1**	**49.7**	**56.1**	**62.2**	**68.2**	**73.3**

Table 7: Performance with GSPO

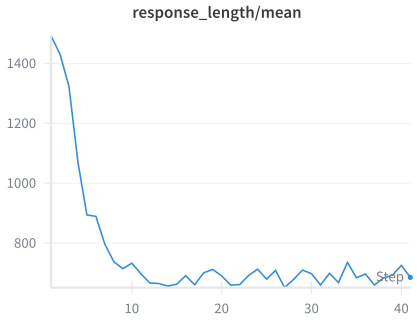

Figure 6: Response length dynamics of Llama3.2-3B-Instruct across different stages of GRPO training.

Figure 7: Word cloud of the top 50 tokens ranked by THR, generated from Qwen2.5-Math-7B on AMC23. Font size is proportional to each token's average THR. Tokens with high THR represent the key reasoning steps most critical in the model's problem-solving process.

C.3 Additional Results on Llama.

Reduced response length. As shown in Figure 6, the response length of Llama3.2-3B declines rapidly after a few epochs, with the average length dropping from about 1.5K tokens to roughly 650. This reduction may stem from the model's limited cognitive behaviors Gandhi et al. (2025).
Exploitation Results on Llama We report the greedy decoding performance of Llama in Table 8. As shown in table, while GRPO achieves the best performance, setting $p > 0$ can improve the greedy decoding performance compared with vanilla THR by 1.1%.
Exploration Results on Llama As shown in Table 9, THR still substantially boosts exploration, achieving over a 7% Pass@K improvement compared to GRPO. Setting $p < 0$ amplifies these exploration gains even further. While baselines such as COV-KL and Pass@K-mixed also provide exploration improvements, they consistently underperform relative to THR.

C.4 Additional THR Token Analysis

We further analyze tokens with high THR values using a word cloud visualization, as shown in Figure 7. The representative tokens can be organized into five functional categories that correspond to step-by-step reasoning:

Base Model	Method	AIME25	AIME24	AMC23	MATH500	Minerva	Olympiad	Avg.
Llama3.2-3B-Instruct	Base	0.0	3.3	22.5	40.2	16.5	11.9	15.7
	GRPO	0.0	26.7	30.0	54.4	22.1	18.1	**25.2**
	THR	0.0	13.3	32.5	51.8	22.1	19.9	23.3
	THR ($p = -0.2$)	3.3	6.7	27.5	51.4	20.6	16.3	21.0
	THR ($p = 0.05$)	3.3	13.3	40.0	50.6	22.4	16.7	<u>24.4</u>

Table 8: Exploitation Results. Pass@1 accuracy (%) using greedy decoding across different methods and datasets. **Bold** indicates the best performance, while <u>underline</u> marks the second-best.

Method	Llama3.2-3B-Instruct Pass@K								
	1	2	4	8	16	32	64	128	256
AIME 2025									
Base	0.2	0.3	0.6	1.2	2.4	4.6	8.45	14.2	20.0
GRPO	0.3	0.7	1.25	2.4	4.3	7.0	10.2	13.2	16.7
Cov KL	0.4	0.7	1.4	2.5	4.5	7.4	11.2	16.3	23.3
Pass@K-mixed	0.7	1.3	2.3	3.9	6.3	9.1	12.6	16.7	20.0
THR	1.0	1.8	3.4	5.7	8.6	12.0	16.7	24.0	30.0
THR ($p = -0.1$)	1.1	2.1	3.8	6.7	10.7	15.3	19.7	24.2	30.0
THR ($p = -0.2$)	0.5	0.9	1.8	3.4	6.4	11.1	17.8	26.3	36.7
AIME 2024									
Base	1.4	2.6	4.8	8.3	13.4	20.3	28.4	35.9	40.0
GRPO	12.7	17.5	22.4	27.4	31.0	33.3	34.9	36.7	40.0
Cov KL	11.9	15.9	20.4	25.6	30.6	33.8	35.8	38.3	43.3
Pass@K-mixed	12.2	17.2	22.4	27.4	30.8	32.8	35.1	38.2	43.3
THR	9.8	15.0	20.5	25.7	29.8	32.6	35.0	38.2	43.3
THR ($p = -0.1$)	9.2	13.9	19.0	24.2	29.3	33.5	36.5	40.0	46.7
THR ($p = -0.2$)	9.4	13.6	18.2	23.1	27.9	32.5	37.1	41.6	46.7
AMC 2023									
Base	9.6	17.0	27.7	41.0	55.7	69.2	80.1	86.4	90.0
GRPO	26.7	36.9	47.3	56.4	63.6	69.5	74.8	79.6	85.0
Cov KL	28.9	39.3	49.6	57.9	64.7	70.8	76.2	81.1	85.0
Pass@K-mixed	28.6	39.3	49.9	58.9	65.8	71.3	76.3	81.4	87.5
THR	26.8	37.9	48.5	57.9	67.0	75.2	82.3	87.5	90.0
THR ($p = -0.1$)	26.1	36.4	47.0	56.4	65.5	74.2	81.5	87.0	90.0
THR ($p = -0.2$)	26.5	36.7	47.6	57.8	66.9	74.4	80.2	84.3	87.5
Average									
Base	3.7	6.6	11.0	16.8	23.8	31.4	39.0	45.5	50.0
GRPO	13.2	18.4	23.7	28.7	33.0	36.6	40.0	43.2	47.2
Cov KL	13.7	18.6	23.8	28.7	33.3	37.3	41.1	45.2	50.5
Pass@K-mixed	13.8	19.3	24.9	30.1	34.3	37.7	41.3	45.4	50.3
THR	12.5	18.2	24.1	29.8	35.1	39.9	44.7	49.9	54.4
THR ($p = -0.1$)	12.1	17.5	23.3	**29.1**	**35.2**	**41.0**	**45.9**	50.4	55.6
THR ($p = -0.2$)	12.1	17.1	22.5	28.1	33.7	39.3	45.0	**50.7**	**57.0**

Table 9: Pass@K performance of different methods using Llama3.2-3B-Instruct .

• **Stating the Given Information**: tokens that capture the initial conditions or input facts (*present, data, paper*).
• **Transformation and Operations**: tokens that describe conversions, equivalence, or transfers of knowledge (*conversion, transfer, equivalent*).
• **Constraints and Relationships**: tokens indicating dependencies, limitations, or structural relations (*relative, intersects, amount, dimensions*).
• **Decision and Selection**: tokens reflecting choices among alternatives or branching reasoning paths (*determine, instead, alternating, altern, others*).
• **Verification and Conclusion**: tokens signaling validation or consolidation of results (*confirms, systematic, answer*).

C.5 Running time of each module.

We also track the average time cost of each module during training, as reported in Table 10. Notably, the data generation (Data Gen) module that using dynamic sampling accounts for the majority of the total training time. In contrast, the overhead introduced by THR is minimal, e.g. 37 seconds for Qwen2.5-Math-1.5B, contributing only a small fraction to the overall cost.

Model+dataset	Data Gen	Model Upd	THR	Ref	Old Prob	**Total** (Sec)
Qwen2.5-Math-1.5B	347	210	37	120	120	834
Qwen2.5-Math-7B	422	371	39	187	187	1206
Llama3.2-3B-Instruction	625	139	26	89	89	968

Table 10: Average running time (per step, in seconds) of each module for different models and tasks.

D Detailed Proofs

D.1 Pass@K as the question level reweighting

Chen et al. (2025); Mahdavi et al. (2025); Walder & Karkhanis (2025) develop RLVR objectives that directly target Pass@K optimization. Starting with GRPO's ancestor, REINFORCE, Mahdavi et al. (2025); Walder & Karkhanis (2025) derive reward rescalings by directly optimizing the Pass@K objective. Mahdavi et al. (2025) apply the same rescaling to advantages giving a GRPO version of their approach. These rescalings upweight the gradient contribution of correct responses that constitute "rare successes"—i.e., responses associated with "hard" questions. Crucially, the reweighting is uniform across all tokens and responses for a given question, which we term *question-level reweighting*. More recently, Chen et al. (2025) introduce an appealing alternative to optimizing Pass@K by incorporating the design directly within GRPO's group structure. Here, we simplify the formulas in Chen et al. (2025) and arrive at an explicit formulation of advantage shaping that reveals its question-level nature. Starting from the defined advantages in Chen et al. (2025):

$$\bar{R}^{\text{group}} = 1 - \frac{\binom{N^-}{K}}{\binom{G}{K}}, \sigma^{\text{group}} = \sqrt{\bar{R}^{\text{group}} \times (1 - \bar{R}^{\text{group}})}$$

$$A^{@K}_{\text{pos}} = \frac{1 - \bar{R}^{\text{group}}}{\sigma^{\text{group}}}, A^{@K}_{\text{neg}} = \left(1 - \bar{R}^{\text{group}} - \frac{\binom{N^- - 1}{K-1}}{\binom{G-1}{K-1}}\right) \times (\sigma^{\text{group}})^{-1}.$$

Since $N^- = (1-q)G$ then we can obtain:

$$\begin{aligned} A^{@K}_{\text{pos}} &= \frac{\binom{N^-}{K}}{\binom{G}{K}\sigma^{\text{group}}} \\ &= \frac{\prod_{i=0}^{K-1}((1-q)G - i)}{\prod_{i=0}^{k-1}(G-i)\sigma^{\text{group}}}, \\ &= \sqrt{\frac{\binom{N^-}{K}/\binom{G}{K}}{1 - \binom{N^-}{K}/\binom{G}{K}}} \\ &= \sqrt{\frac{\binom{N^-}{K}/\binom{G}{K}}{1 - \binom{N^-}{K}/\binom{G}{K}}} \cdot \sqrt{\frac{q}{1-q}}) \cdot \sqrt{\frac{1-q}{q}} \\ &= \sqrt{\frac{\binom{N^-}{K}/\binom{G}{K}}{1 - \binom{N^-}{K}/\binom{G}{K}}} \cdot \sqrt{\frac{q}{1-q}}) \cdot \hat{A}_{\text{pos}} \end{aligned} \tag{11}$$

then harder question will have a larger $1-q$ thus larger advantage, then we derive the negative advantage.

$$
\begin{aligned}
A_{\text{neg}}^{@K} &= \left(\frac{\binom{N^-}{K}}{\binom{G}{K}} - \frac{\binom{N^--1}{K-1}}{\binom{G-1}{K-1}}\right)\frac{1}{\sigma^{\text{group}}} \\
&= \left(\frac{\prod_{i=0}^{K-1}(N^- - i)}{\prod_{i=0}^{K-1}(N-i)} - \frac{\prod_{i=1}^{K-1}(N^- - i)}{\prod_{i=1}^{K-1}(N-i)}\right)\frac{1}{\sigma^{\text{group}}} \\
&= \left(1 - \frac{G}{N^-}\right)\frac{\prod_{i=0}^{k-1}(N^- - i)}{\prod_{i=0}^{k-1}(G-i)}\frac{1}{\sigma^{\text{group}}} \\
&= -\frac{q}{1-q}A_{\text{pos}}^{@K} \\
&= \left(A_{\text{pos}}^{@K}\cdot\sqrt{\frac{q}{1-q}}\right)\cdot\left(-\sqrt{\frac{q}{1-q}}\right) \\
&= \sqrt{\frac{\binom{N^-}{K}/\binom{G}{K}}{1-\binom{N^-}{K}/\binom{G}{K}}}\cdot\sqrt{\frac{q}{1-q}}\cdot\hat{A}_{\text{neg}}
\end{aligned}
\tag{12}
$$

By combining Equation (11) and Equation (12), we arrive at Equation (6), completing the derivation.

D.2 Relationship between THR and Entropy Regularizer

Under some mild assumptions, optimizing THR plays a similar role as regularizing[2] the evolution of the token entropy in a more efficient way. Because, as stated in the main context, THR considers cross-token influence while current analysis on token entropy consider the influence of learning a observing token on itself Cui et al. (2025). We start from Lemma 1 proposed in Cui et al. (2025), which is how the `Cov-KL` regularizer is derived.

Lemma 1 in Cui et al. (2025): Let the actor policy π_θ be a tabular softmax policy, the difference of information entropy given states between two consecutive steps satisfy:

$$
\Delta\mathcal{H}^t \triangleq \mathcal{H}(\pi_{\theta(t+1)}) - \mathcal{H}(\pi_{\theta(t)}) = -\mathsf{Cov}_{\boldsymbol{y}\sim\pi_{\theta(t)}(\cdot|x)}\left(\log\pi_{\theta(t)}(\boldsymbol{y}\mid\boldsymbol{x}), \mathbf{l}_y^{t+1} - \mathbf{l}_y^t\right), \tag{13}
$$

where $\mathbf{l}$ is the logits vector provided by the model after feeding the input $\boldsymbol{x}$. For notational simplicity, we use the superscript t to denote the training step, rather than an exponent. The equation above holds as long as a first-order Taylor expansion is valid at the logits level, independent of the specific model under consideration. In other words, this lemma is agnostic to the mechanism by which $\mathbf{l}$ evolves, which depends on the particular model architecture or parameterization.

Recall the definition of the covariance:

$$
\mathsf{Cov}_{y\sim\pi}(X,Y) = \mathbb{E}_{y\sim\pi}[X\cdot Y] - \mathbb{E}_{y\sim\pi}[X]\mathbb{E}_{y'\sim\pi}[Y].
$$

[2]The strength and direction are controlled by the value and sign of hyper-parameter p

Equation (13) can then be written as:

$$
\begin{aligned}
\Delta\mathcal{H}^t(\chi) &= -\mathsf{Cov}_{y\sim\pi_{\theta(t)}(\cdot|\chi)}\left(\log\pi_{\theta(t)}(y\mid\chi), \mathbf{l}_y^{t+1}-\mathbf{l}_y^t\right)\\
&= \mathbb{E}_{y\sim\pi_{\theta(t)}}[\log\pi_{\theta(t)}(y\mid\chi)]\mathbb{E}_{y'\sim\pi_{\theta(t)}}[\mathbf{l}_{y'}^{t+1}-\mathbf{l}_{y'}^t] - \mathbb{E}_{y\sim\pi_{\theta(t)}}\left[(\mathbf{l}_y^{t+1}-\mathbf{l}_y^t)\log\pi_{\theta(t)}(y\mid\chi)\right]\\
&= -\mathcal{H}(\pi_{\theta(t)})\mathbb{E}_{y\sim\pi_{\theta(t)}}[\mathbf{l}_y^{t+1}-\mathbf{l}_y^t] - \mathbb{E}_{y\sim\pi_{\theta(t)}}\left[(\mathbf{l}_y^{t+1}-\mathbf{l}_y^t)\log\pi_{\theta(t)}(y\mid\chi)\right]\\
&= -\mathcal{H}(\pi_{\theta(t)})\sum_{v=1}^{V}\pi_{\theta(t)}(y=v\mid\chi)(\mathbf{l}_v^{t+1}-\mathbf{l}_v^t) -\\
&\quad \sum_{v=1}^{V}\pi_{\theta(t)}(y=v\mid\chi)(\mathbf{l}_v^{t+1}-\mathbf{l}_v^t)\log\pi_{\theta(t)}(y=v\mid\chi)\\
&= -\sum_{v=1}^{V}\pi_{\theta(t)}(y=v\mid\chi)(\mathbf{l}_v^{t+1}+\mathbf{l}_v^t)\left(\mathcal{H}(\pi_{\theta(t)})+\log\pi_{\theta(t)}(y=v\mid\chi)\right)\\
&= -\left\langle\mathcal{H}(\pi_{\theta(t)})\pi_{\theta(t)}(\cdot\mid\chi)+\pi_{\theta(t)}(\cdot\mid\chi)\odot\log\pi_{\theta(t)}(\cdot\mid\chi), \mathbf{l}^{t+1}-\mathbf{l}^t\right\rangle\\
&= -\mathcal{H}(\pi_{\theta(t)})\left\langle\pi_{\theta(t)}(\cdot\mid\chi)+\underbrace{\frac{1}{\mathcal{H}(\pi_{\theta(t)})}\pi_{\theta(t)}(\cdot\mid x)\odot\log\pi_{\theta(t)}(\cdot\mid x)}_{V\times 1,\text{defined as } Q(\chi)}, \mathbf{l}^{t+1}-\mathbf{l}^t\right\rangle\\
&= c\left\langle -Q(\chi)-\pi_{\theta(t)}(\cdot\mid\chi), \mathbf{l}^{t+1}(\chi)-\mathbf{l}^t(\chi).\right\rangle
\end{aligned}
\tag{14}
$$

where the operator $\odot$ is the element-wise multiplication of two vectors, $\chi \triangleq \boldsymbol{x}, \boldsymbol{y}_{<k}$ is the context for the prediction of the k-th token, and c is a constant for notation conciseness. In the last equation, we reintroduce the input χ to the notation to remind readers that the entire equation is conditioned on a given context sequence χ. That is an important extension, because most existing works on entropy regularization (e.g., Cui et al. (2025)) **only focus on the influence introduced by updating the observing token on itself**. In other words, the χ for Q and $\mathbf{l}$ are identical. The `Cov-KL` method compared in Table 4 just applies the quantity above to select tokens with high covariances, and then uses the KL penalty to restrict the update of them.

We here connect THR to entropy in a more systematic way by showing that THR can control the rate of entropy growth $\mathcal{H}^t(\chi)$ through the choice of p. Beyond the simplified tabular softmax setting, our analysis extends to more realistic models with shared parameters across tokens. In this case, THR naturally captures the **cross-token** influences that arise throughout the learning process. In other words, when tracking the confidence change of $\pi_{\theta(t)}(y\mid\chi)$, THR accounts for the learning dynamics of all other tokens across all responses, i.e., $\boldsymbol{y}_{i,<k}$ for varying i and k.

To make the notations concise, we follow the settings in Ren & Sutherland (2025) and use χ_o and χ_u to denote the "observing" token and "updating" context, respectively. Then, Equation (14) becomes:

$$
\Delta\mathcal{H}^t(\chi_o) = c\left\langle -Q(\chi_o)-\pi_{\theta(t)}(\cdot\mid\chi_o), \mathbf{l}^{t+1}(\chi_o)-\mathbf{l}^t(\chi_o)\right\rangle.
$$

Following Deng et al. (2025), and under the unconstrained features assumption Deng et al. (2025); Mixon et al. (2022), we then represent $\mathbf{l}^t(\chi_o) = \mathbf{W}^t\mathbf{h}_o$, where $\mathbf{W}\in\mathbb{R}^{V\times d}$ denotes the shared read-out layer and $\mathbf{h}_o\in\mathbb{R}^{d\times 1}$ is the feature vector produced by the LLM backbone, conditioned on the context sequence $\chi_{u/o} = \boldsymbol{x}, \boldsymbol{y}_{u/o,<k}$. Note that while $\mathbf{l}^t(\chi_o)$ shares the same $\mathbf{W}^t$, the feature vector $\mathbf{h}$ differs across contexts due to variations in input sequences. The difference vector $\mathbf{l}^{t+1}(\chi_o)-\mathbf{l}^t(\chi_o)\in\mathbb{R}^{V\times 1}$ can then be expressed as:

$$
\mathbf{l}^{t+1}(\chi_o)-\mathbf{l}^t(\chi_o) = (\mathbf{W}^{t+1}-\mathbf{W}^t)\mathbf{h}_o = -\eta\nabla_{\mathbf{W}}\mathcal{L}(\sigma(\mathbf{W}\mathbf{h}_u), \boldsymbol{e}_u)\mathbf{h}_o,
$$

where η is the learning rate, $\sigma(\cdot)$ is the softmax function, and $\boldsymbol{e}_u$ is the one-hot distribution determined by the label of y_u. When the cross-entropy loss is considered, the equation above can be simplified to

$$
\mathbf{l}^{t+1}(\chi_o)-\mathbf{l}^t(\chi_o) = \underbrace{(\boldsymbol{e}_u-\pi_{\theta(t)}(\cdot\mid\chi_u))}_{V\times 1}\cdot\underbrace{\mathbf{h}_u^\top\mathbf{h}_o}_{1\times 1}.
$$

Substituting this back to Equation (14), we can get

$$
\Delta\mathcal{H}^t(\chi_o) = c\left\langle -Q(\chi_o)-\pi_{\theta(t)}(\cdot\mid\chi_o), \boldsymbol{e}_u-\pi_{\theta(t)}(\cdot\mid\chi_u)\right\rangle\cdot\mathbf{h}_u^\top\mathbf{h}_o \tag{15}
$$

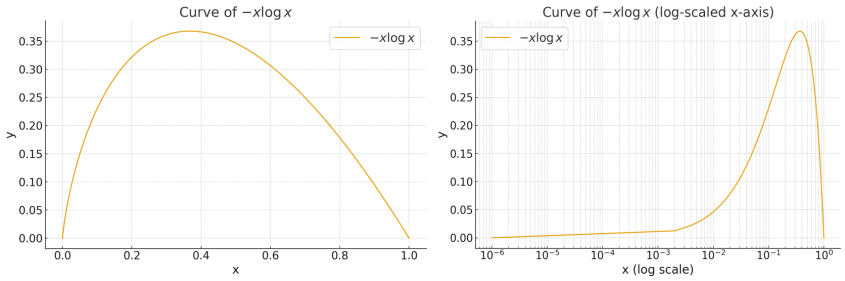

Figure 8: The shape of $-x \log x$ for $x \in (0, 1)$, shown in both the original and logarithmic scales.

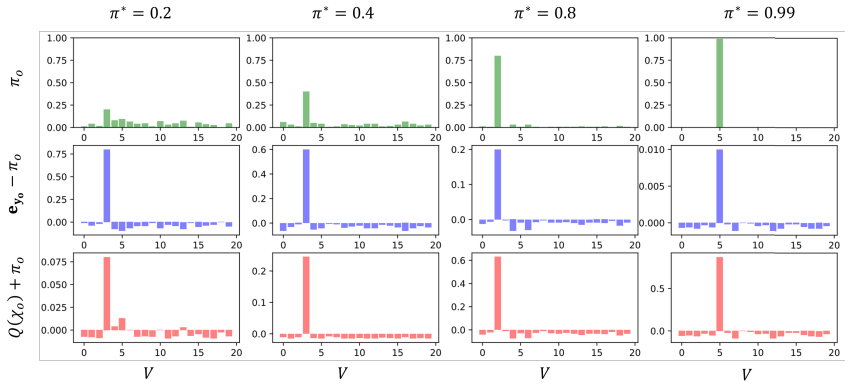

Figure 9: Four examples of the distribution of π, $\mathbf{e}_o - \pi$ and $Q + \pi$.

Now, recall our definition of THR in Definition 4.1, where for each k in the summation, the term has the format $\langle \mathbf{h}_{\mathbf{x},\boldsymbol{y}^+_{i,<k}}, \mathbf{h}_{\mathbf{x},\boldsymbol{y}_{<k'}} \rangle$, which is just $\mathbf{h}_u^\top \mathbf{h}_o$ above. Combining the definition of α and using the notations in this section, we can rewrite the signed-THR as follows:

$$\mathsf{sign}(\boldsymbol{y}_u) \cdot \mathrm{THR}(\boldsymbol{y}_o, \boldsymbol{y}_u, k) = \sum_u \langle \boldsymbol{e}_o - \pi_{\theta(t)}(\cdot \mid \chi_o), \boldsymbol{e}_u - \pi_{\theta(t)}(\cdot \mid \chi_u) \rangle \cdot \mathbf{h}_u^\top \mathbf{h}_o, \tag{16}$$

where $\mathsf{sign}(\boldsymbol{y}_u)$ depends on whether the completion is correct or not. Now, comparing the inner product in Equation (15) and Equation (16), it is clear that the directional similarity between $-Q(\chi_o)$ and $\boldsymbol{e}_o$ determines the effect introduced by THR and the entropy regularizer.

We now show that, under mild assumptions (which typically hold during LLM fine-tuning), $-Q(\chi_o)$ and $\boldsymbol{e}_o$ point to a very similar direction (measured by their cosine similarity).

This observation follows from the shape of the function $-x \log x$, illustrated in Figure 8. In a distribution where most probability mass is concentrated on few dimensions, the dominant entry of $\pi^t_{\theta(t)}(\cdot \mid \chi_o) \odot \log \pi^t_{\theta(t)}(\cdot \mid \chi_o)$ is significantly larger than the rest. To validate this, we randomly generate distributions and compute the cosine similarity between $-Q(\chi_o)$ and $\mathbf{e}_o$ in Figure 9 and Figure 10. The results show a clear trend: as both the vocabulary size and the peakiness of the distribution increase, the alignment between the two vectors becomes stronger.

We now examine the relationship between THR and entropy. Recall that THR is defined as

$$\hat{A}^{\mathrm{THR(p)}}_{i,k} = \mathbb{1}[|\mathrm{THR}_{i,k}| > \tau] \cdot (1 + \mathrm{sign}(\mathrm{THR}_{i,k}) \cdot p) \cdot \hat{A}_{i,k}.$$

When $p < 0$, tokens with larger THR values receive stronger penalties. Since, in most cases, $\Delta\mathcal{H}^t(\chi)$ and THR point in similar directions, this implies that tokens with higher potential entropy change are penalized, closely aligning with the intuition behind `Cov-KL`. However, as shown in our experiments, THR achieves greater improvements in exploration performance because it explicitly accounts for **cross-token** influence, rather than relying solely on entropy-based signals on a token's self-influence, as in COV-KL Cui et al. (2025).

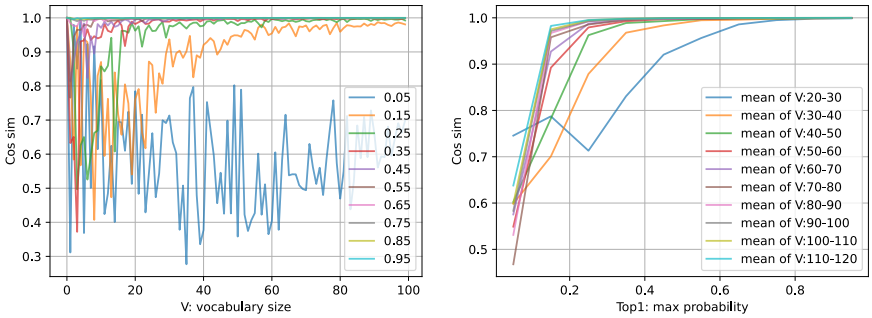

Figure 10: We sweep the value of vocabulary size V and argmax probability of the distribution π^*. The distribution is generated by fixing π^* and randomly assign the extra probability mass to other dimensions. The results show that the cosine similarity between $\mathbf{e}_o - \pi$ and $Q + \pi$ is indeed very large when V and π^* are large enough.

E USAGE OF LARGE LANGUAGE MODEL

In preparing this paper, we made limited use of ChatGPT to support writing and editing. Specifically, LLMs were employed for language polishing, grammar refinement, and rephrasing sentences to improve clarity and readability. Importantly, all technical content, including theoretical analysis, algorithm design, and experimental results, was conceived, implemented, and validated by the authors. LLM outputs were always critically reviewed, verified, and revised before inclusion. No LLM-generated text, figures, or tables were incorporated without careful human oversight.