UNIFIED DATA SELECTION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Effectively training Large Language Models (LLMs) for complex, long-CoT reasoning is often bottlenecked by the need for massive high-quality reasoning data. Existing methods are either computationally expensive or fail to reliably distinguish high- from low-quality reasoning samples. To address this, we propose **High-Entropy Sum (HES)**, a training-free metric that quantifies reasoning quality by summing only the entropy of the top (*e.g.*, 0.5%) highest-entropy tokens in each reasoning sample. We validate HES across three mainstream training paradigms: Supervised Fine- tuning (SFT), Rejection Fine-tuning (RFT), and Reinforcement Learning (RL), with extensive results demonstrating its consistent effectiveness and significantly reduced computational overhead. In SFT, training on the top 20% HES-ranked data matches full-dataset performance, while using the lowest-HES data degrades it. In RFT, our HES-based training approach significantly outperforms baseline methods. In RL, HES-selected successful trajectories enable the model to learn strong reasoning patterns, significantly surpassing other compared methods. Our findings establish HES as a robust, training-free metric that enables a unified, effective, and efficient method for developing advanced reasoning in LLMs.

1 INTRODUCTION

The ability of Large Language Models (LLMs) to solve complex problems through Chain-of-Thoughts (CoT) reasoning has become a central focus in frontier research (Jaech et al., 2024; DeepSeek-AI, 2025; Yang et al., 2025). To enhance models' reasoning capabilities, dominant training paradigms, including Supervised Fine-tuning (SFT) (Ouyang et al., 2022), Rejection Fine-tuning (RFT) (Yuan et al., 2023a), and Reinforcement Learning (RL) (Shao et al., 2024), heavily rely on high-quality training data. However, the indiscriminate expansion of training data often introduces more noise and incurs additional costs, spurring a critical need for efficient and effective data selection methods (Zhou et al., 2023). The key lies in defining a robust metric to rapidly and accurately distinguish high-quality data from low-quality data (Li et al., 2024b).

Current research has explored various strategies for filtering training data, such as filtering by length (Rae et al., 2021), perplexity (Marion et al., 2023), or average token entropy (Sabbineni et al., 2023). However, these metrics share common limitations when applied to long-CoT reasoning scenarios, where extended and methodical reasoning processes are particularly valuable for model learning (Chen et al., 2025). Specifically, they rely on a **coarse-grained, global evaluation** of the reasoning path, assigning equal weight to all tokens. This uniform averaging fails to capture the nuanced, multi-stage structure of complex reasoning, which often involves planning, exploration, and reflection (Chen et al., 2025). Although some methods address this by training auxiliary task-specific models for data selection (SHUM et al., 2025) or by employing powerful LLMs to choose among multiple responses (Toshniwal et al., 2025), such approaches incur **substantial computational overhead**. Moreover, because these methods are not directly aligned with or adaptive to the target model being optimized, they risk yielding suboptimal training data.

To address the above gaps, our objective is to develop an **efficient** method for selecting high-quality reasoning samples by identifying and leveraging **fine-grained, critical information** embedded in long-CoT responses. Recently, Wang et al. (2025) has suggested that a small number of high-entropy *critical tokens* in the reasoning process serve as a key driver for performance improvement. Inspired by this, we conduct a preliminary study on token-level entropy distribution throughout the reasoning process. Note that high-entropy tokens are identified as those falling within the top 0.5% of the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

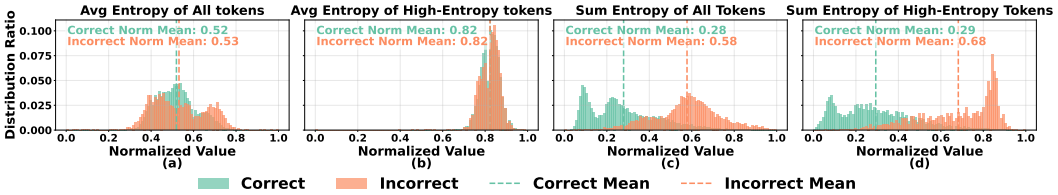


Figure 1: The comparative analysis of discriminative ability is conducted on 512 responses per problem generated by Qwen3-14B for AIME 2025. The results demonstrate that metric of *(d) Sum Entropy of High-Entropy Tokens* is the most effective to distinguish between high- and low-quality samples.

entropy distribution across all tokens. We present the results in Figure 1, from which we can observe: 1) Neither the overall average token entropy (Fig. 1 (a)) nor the average entropy of high-entropy tokens (Fig. 1 (b)) can differentiate high- from low-quality samples. 2) Comparably, the summation of entropy over high-entropy tokens (Fig. 1 (d)) provides a more robust distinction compared to the total entropy of all tokens (Fig. 1 (c)).

These findings demonstrate a strong correlation between sample quality and the cumulative entropy of *critical tokens*. Capitalizing on this relationship, we propose **High-Entropy Sum (HES)**, a train-free metric to quantify the long-CoT reasoning quality of the samples. Specifically, HES is calculated by summing the entropy values of the top $k\%$ (*e.g.*, 0.5%) of tokens with the highest entropy. Building on this metric, we propose an efficient, unified data selection method to curate high-quality reasoning data for LLMs, applicable across SFT, RFT, and RL training paradigms.

To evaluate the effectiveness of our proposed HES-based method, we conduct extensive experiments across the three training paradigms. The results reveal that: 1) **In SFT**, training on the top 20% of samples ranked by HES achieves performance comparable to the full dataset across diverse domains, while expanding to the top 80% consistently surpasses the baseline. Notably, we demonstrate that data selection using a lightweight 0.6B model yields results comparable to those of an 8B model, validating the cost-effectiveness of HES for offline data curation; 2) **In RFT**, HES significantly outperforms traditional random selection, serving as an effective, training-free reward signal; 3) **In RL**, our strategy of oversampling rollouts and selecting only the top-50% ranked by HES outperforms the standard setting where all rollouts participate in policy updates. Collectively, these findings highlight that HES is not only high-performing but also remarkably efficient: it enables scalable proxy selection with small models in offline SFT and incurs negligible storage overhead in online RFT and RL settings. The empirical results confirm that our HES-based method is both an effective and efficient approach for selecting high-quality data, thereby advancing the reasoning capabilities of LLMs.

In summary, our contributions are as follows:

- We introduce High-Entropy Sum (HES), a train-free metric that quantifies reasoning quality by analyzing only the high-entropy tokens in the response paths.
- We propose an effective and efficient HES-based data selection method that is compatible with SFT, RFT, and RL training paradigms.
- Extensive experiments across diverse domains demonstrate that HES consistently outperforms baselines in SFT, RFT, and RL, achieving superior model performance with significantly reduced computational overhead.

2 Preliminaries

2.1 Training Paradigms

Supervised Fine-Tuning (SFT). Let $\mathcal{D} = \{(x, y)\}$ denote a corpus of correct demonstrations, where x is a query and y is the reference response. SFT minimizes the cross-entropy loss, defined by the objective function: $\mathcal{L}_{SFT}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[-\log \pi_\theta(y|x)]$, where θ represents the model's parameters and $\pi_\theta(y|x)$ is the probability assigned by the model to y given x.

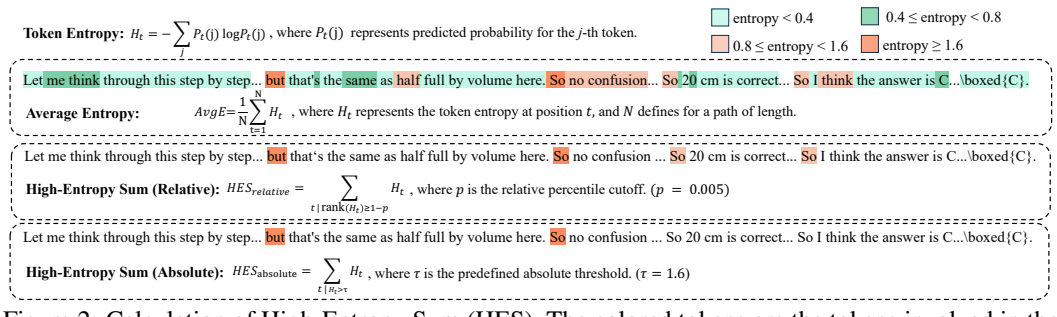


Figure 2: Calculation of High-Entropy Sum (HES). The colored tokens are the tokens involved in the calculation.

Rejection Sampling Fine-Tuning (RFT). RFT (Yuan et al., 2023a) augments SFT by generating training samples through the model's own exploration. The process involves three steps: (1) Generation: For query x, generate m different candidate responses $\{y_1, y_2, \ldots, y_m\}$; (2) Selection: Use selection function $R(y)$ to select a response subset Y: $Y = \underset{y_i \in \{y_1,\ldots,y_m\}}{\text{argmax}} \; R(y_i)$; (3) Fine-tuning: Create new dataset $\mathcal{D}^*$ with Y and fine-tune using SFT.

Reinforcement Learning (RL). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is a variant of Proximal Policy Optimization (PPO) (Schulman et al., 2017). For each query, GRPO samples a group of G responses $\{o_1, \ldots, o_G\}$ with corresponding rewards $\{r_1, \ldots, r_G\}$. The advantage $\hat{A}_i$ for response o_i is first computed as: $\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G)}$. This group-normalized advantage is then used in a clipped policy gradient objective to update parameters θ. The full objective function to be maximized is: $\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}[\min\left(r_t(\theta)\hat{A}_i, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\right) - \beta D_{\text{KL}}(\pi_{\theta_{\text{old}}} \| \pi_\theta)]$. Here, $r_t(\theta)$ is the probability ratio, $\hat{A}_i$ is the group-relative advantage, ϵ is a clipping bound, and the final term is a KL-divergence penalty with weight β to stabilize training.

2.2 Token Entropy

Token entropy. For a token position t with probability distribution P_t over the vocabulary, the token entropy H_t is given by: $H_t = -\sum_j P_t(j) \log P_t(j)$, where $P_t(j)$ represents the predicted probability for the j-th token. Token entropy is the fundamental metric to measure a model's uncertainty during generation. Low entropy typically occurs during predictable parts of a reasoning path, such as completing a common phrase, performing a simple calculation, or following a standard template. Conversely, high entropy signifies high uncertainty where the model is considering multiple viable and often competing options. In the context of long-CoT reasoning, these high-entropy moments are particularly important as they often correspond to critical forks where the model must make a non-trivial decision that will shape the subsequent trajectory (Wang et al., 2025).

Average Entropy. The common approach to quantifying the overall uncertainty of a reasoning path is average entropy (Sabbineni et al., 2023), defined for a path of length N as: AvgE $= \frac{1}{N}\sum_{t=1}^{N} H_t = \frac{1}{N}$ES, where ES is the entropy sum of all tokens. However, average entropy is limited because it masks critical local signals by averaging over long sequences. A high-quality reasoning path that successfully navigates multiple challenging forks may receive a similar score to one that follows a straightforward, low-complexity approach. This inability to distinguish between different reasoning complexity makes the metric unreliable for identifying the most valuable samples.

3 Methodology

To overcome the limitations of global metrics, we first introduce High-Entropy Sum (HES) as a quantitative measure of reasoning quality and then leverage it to guide data selection across three major training paradigms.

3.1 HIGH-ENTROPY SUM

Our core metric is HES, which quantifies the cumulative intensity of pivotal, high-entropy moments in a reasoning path. We define this metric in two primary formulations as shown in Figure 2: a robust, adaptive version using a relative, percentile-based threshold ($\text{HES}_{\textit{relative}}$), and a simpler alternative using a fixed, absolute threshold ($\text{HES}_{\textit{absolute}}$).

$\text{HES}_{\textit{relative}}$. This metric is designed to capture the cumulative complexity of a sample by focusing on its most uncertain tokens in an adaptive manner.

$$\text{HES}_{\textit{relative}} = \sum_{t \mid \text{rank}(H_t) \geq 1-p} H_t. \tag{1}$$

Here, $\text{rank}(H_t)$ selects the tokens t whose entropy H_t rank within the top p-th percentile of a sample for summation (e.g., $p = 0.005$ for the top 0.5%). A higher $\text{HES}_{\textit{relative}}$ score indicates the successful navigation of more numerous and intense forks, which represents higher quality. The relative threshold makes this metric robust to variations in length among different models and tasks.[1]

$\text{HES}_{\textit{absolute}}$. Alternatively, this metric is used for quality estimation by a fixed cutoff.

$$\text{HES}_{\textit{absolute}} = \sum_{t \mid H_t > \tau} H_t. \tag{2}$$

Here, the sum is taken over the tokens t whose entropy H_t exceeds a predefined absolute value τ (e.g., $\tau = 1.6$). While less adaptive than the relative threshold, this method provides a straightforward alternative for quality estimation, particularly in contexts where a consistent entropy scale can be expected across all samples.

AvgHE. To test whether the cumulative sum of uncertainty is a more effective signal than its average intensity, we introduce average entropy of high-entropy tokens.

$$\text{AvgHE} = \frac{1}{|T_{high}|} \sum_{t \in T_{high}} H_t = \frac{\text{HES}}{|T_{high}|}. \tag{3}$$

For this metric, we identify the set of high-entropy tokens T_{high} using the relative threshold described above. AvgHE is defined as $\text{HES}_{\textit{relative}}$ normalized by $|T_{high}|$, the number of tokens in the set. It is designed to isolate the average complexity of key-fork tokens, different from HES.

3.2 UNIFIED HES-GUIDED DATA SELECTION

We apply the proposed HES metric to curate data and guide training across three major paradigms: SFT, RFT, and RL.

Preliminaries. Let $\mathcal{D} = \{(x, y)\}$ denote a dataset where x represents the prompt and y represents the reasoning response. Let $S(y)$ denote the HES score of a response y, calculated as defined in Eq. (1). We define a selection ratio $\rho \in (0, 1]$ to control the proportion of data retained.

Selection for SFT. In the SFT stage, our goal is to identify a high-quality subset $\mathcal{D}_{\text{SFT}}$ from the full dataset $\mathcal{D}$. We rank all samples based on their HES scores and select the top-ρ percentile. Formally, the curated dataset is defined as:

$$\mathcal{D}_{\text{SFT}}(\rho) = \{(x, y) \in \mathcal{D} \mid S(y) \geq \tau_\rho\} \tag{4}$$

where τ_ρ is the score threshold corresponding to the $(1 - \rho)$-th percentile of the HES distribution. We primarily investigate two settings: (1) **Efficiency Test** ($\rho = 0.2$), training on the top 20% highest-HES

[1] As experiments will demonstrate, $\text{HES}_{\textit{relative}}$ provides better results than $\text{HES}_{\textit{absolute}}$. This is because its adaptive nature makes it more robust across the diverse entropy distributions found in different models and reasoning paths. So, for the remainder, HES will refer to $\text{HES}_{\textit{relative}}$ unless otherwise specified.

Table 1: Definition of asymmetric sampling strategies for RL. Y^+ and Y^- denote the pools of correct and incorrect trajectories, respectively. ρ_S indicates selection based on HES ranking. And half of the positive and negative samples are sampled in the full batch.

Strategy	Positive Samples (P)	Negative Samples (N)	Rationale
Pos-High, Neg-Rand (Ours)	Top-$\rho_S(Y^+)$	Random(Y^-)	Promotes complexity mastery & error diversity
Pos-Rand, Neg-Rand	Random(Y^+)	Random(Y^-)	Unbiased stochastic baseline
Pos-High, Neg-Low	Top-$\rho_S(Y^+)$	Bottom-$\rho_S(Y^-)$	Maximizes positive-negative contrast
Pos-Rand, Neg-Low	Random(Y^+)	Bottom-$\rho_S(Y^-)$	Evaluates impact of weak negatives
Pos-Length/Difficulty, Neg-Rand	Top-Length / Difficulty	Random(Y^-)	Benchmarks against heuristic criteria

samples to evaluate sample efficiency; and (2) **De-noising Test** ($\rho = 0.8$), pruning the bottom 20% lowest-HES samples to test if removing low-quality data improves performance over the full dataset.

Selection for RFT. For RFT, we first generate K candidate responses $\mathcal{Y}_i = \{y_{i,1}, \ldots, y_{i,K}\}$ for each query x_i, and filter them to retain only the correct solutions, denoted as $\mathcal{Y}_i^+$. We then apply HES selection in two scopes:

- **Per-Query Selection:** For each query x_i, we select the top-k responses with the highest HES scores from the candidate set: $\mathcal{Y}_i^* = \text{Top-}k(\{y \in \mathcal{Y}_i^+ \mid S(y)\})$. This tests HES's ability to identify high-quality solutions from a localized pool with a fixed budget per problem[2].
- **Global Pool Selection:** We aggregate all correct responses from all queries into a global pool $\mathcal{P}_{\text{global}} = \bigcup_i \mathcal{Y}_i^+$. We then perform selection on this global set, retaining the top-N samples with the highest HES scores (where N matches the total data volume of the per-query setting). This challenging setting tests whether HES can distinguish high-quality reasoning across a diverse, global distribution.

Selection for RL. In the RL stage, the model generates a group of outputs for each prompt. Let Y^+ and Y^- denote the sets of correct and incorrect trajectories, respectively. Unlike standard approaches that sample uniformly or utilize the full batch, we propose an *asymmetric sampling strategy* that differentiates the selection logic for positive and negative signals to maximize training efficiency.

Our primary strategy, **Pos-High, Neg-Rand**, prioritizes the highest-HES positive samples to encourage the mastery of complex reasoning patterns, while retaining random negative samples to ensure robust exposure to diverse error modes. To systematically isolate the contributions of these components, we define a comprehensive sampling space encompassing five distinct strategies, as detailed in Table 1. This framework allows us to validate the efficacy of HES in identifying valuable positive reinforcement signals while demonstrating the necessity of unbiased negative feedback.

4 Experiments

4.1 Experimental Setup

To ensure a fair and consistent evaluation, we define the common baselines and experimental configurations used across the SFT, RFT, and RL stages.

Baselines. We compare our proposed method against the following selection strategies: **Random** uniformly samples a subset of data from the available pool, serving as the standard baseline. **Length** prioritizes samples with the longest token counts. **Difficulty** selects samples with intermediate difficulty scores, filtering out both trivial and extremely hard cases. **Highest-HES** selects the top samples ranked by our HES metric, as defined in Section 3.2. **Lowest-HES** selects the bottom samples ranked by HES, serving as a negative control to validate the sample quality captured by our metric.

Datasets and Models. We conduct experiments across three domains: Math, STEM, and Code. Detailed dataset statistics and training hyperparameters are provided in Appendix C. For **SFT**, our

[2]If the number of $\mathcal{Y}_i^+$ is less than k, then all of $\mathcal{Y}_i^+$ are taken. In our experiments, K is 32 and $k = 2, 4, 8$.

Table 2: Main results of SFT performance on **Math** domain. Results are reported as average pass@1 accuracy over 16 sampling paths (Average@16 %). **Bold** indicates the best performance per benchmark.

Method	Ratio	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Fullset	100	50.83	34.17	35.21	28.13	24.58	42.94	**6.94**	31.83
Random	20	39.79	27.92	24.79	20.21	20.42	30.00	3.88	23.86
Length	20	46.67	34.79	31.04	24.79	23.54	36.06	6.44	29.05
Difficulty	20	38.13	27.50	25.83	22.50	18.96	26.50	3.69	23.30
Lowest-HES	20	18.54	18.96	11.88	11.04	7.92	11.00	2.94	11.75
Highest-HES	20	50.00	37.92	35.00	26.46	23.33	41.00	5.69	31.34
Highest-HES	80	**52.50**	**41.67**	**37.50**	**29.17**	**27.71**	**47.00**	6.69	**34.61**

Table 3: SFT performance on **STEM** and **Code** domains. GPQA results are reported as Average@16, LiveCodeBench results are reported as Average@10 following Jain et al. (2025). **Bold** indicates the best.

Method	Ratio	GPQA	LiveCodeBench
Fullset	100	47.66	58.76
Random	20	45.86	49.06
Length	20	50.82	53.84
Lowest-HES	20	31.44	35.32
Highest-HES	20	**51.96**	54.43
Highest-HES	80	49.43	**61.65**

Table 4: Analysis of small-to-large model transferability. We compare the performance of the Qwen3-8B model trained on data selected by proxy models of varying sizes (0.6B) versus self-selection (8B). **Bold** indicates the best.

Method	Ratio	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Highest-HES (8B)	20	**50.00**	**37.92**	**35.00**	**26.46**	**23.33**	41.00	5.69	**31.34**
Highest-HES (0.6B)	20	49.17	35.42	32.50	24.17	22.29	**41.50**	**6.38**	30.20

primary experiments are conducted using Qwen3-8B-Base (Yang et al., 2025) on the *Mixture-of-Thoughts* dataset (Hugging Face, 2025).[3] For **RFT**, experiments are performed using DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025) on the *DeepScaleR* dataset (Luo et al., 2025a). For **RL**, we also utilize the *DeepScaleR* dataset (Luo et al., 2025a). Due to the high computational cost of RL, we adopt the smaller DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025).

Evaluation Settings. We evaluate model performance on widely-recognized benchmarks, including AIME24 (Art of Problem Solving, 2024a;b), AIME25 (Art of Problem Solving, 2025a;b), HMMT23 (HMMT, 2023), HMMT24 (HMMT, 2024), HMMT25 (HMMT, 2025), OlymMATH (Sun et al., 2025), GPQA-Diamond (Rein et al., 2024), and LiveCodeBench (Jain et al., 2025). We use a temperature of 0.6 and a maximum generation length of 32,768.

4.2 SFT Experiments

Main Results. To evaluate the efficacy of HES as a unified metric, we investigate its performance across Math, Code, and STEM domains. We compare HES against random and heuristic baselines in two settings: the *Efficiency Test* and *De-noising Test*. As shown in Table 2 and 3, in the *Efficiency Test*, HES consistently outperforms all baselines, even surpassing the full dataset in STEM, which indicates effective filtration of redundant signals. In the *De-noising Test*, pruning the bottom 20% of HES-ranked data yields the best overall performance in Math and Code, significantly beating the full

[3] SFT results for Qwen3-4B-Base and DeepSeek-R1-Distill-Qwen-7B, and RFT experiments using DeepSeek-R1-Distill-Llama-8B are provided in Appendix G. And because the Code and STEM subsets of Mixture-of-Thoughts do not provide difficulty metrics, there is no difficulty baseline.

Table 5: Ablation study on different entropy-based metrics for SFT data selection. **Bold** indicates the best.

Method	Ratio	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Highest-AvgE	20	40.21	31.25	28.54	23.13	19.17	31.19	4.88	25.48
Highest-AvgHE	20	42.29	29.79	31.46	22.08	18.75	33.06	4.31	25.96
Highest-ES	20	47.92	34.17	30.42	25.63	22.71	38.50	5.25	29.23
Highest-HES$_{absolute}$	20	42.29	32.29	32.92	**29.38**	19.38	36.06	**6.94**	28.47
Highest-HES	20	**50.00**	**37.92**	**35.00**	26.46	**23.33**	**41.00**	5.69	**31.34**

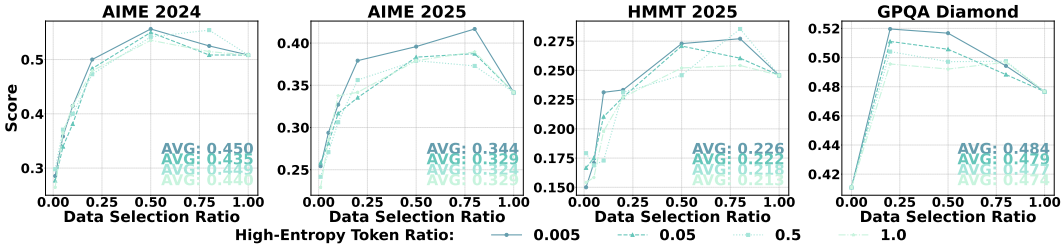


Figure 3: Parameter sensitivity across diverse domains, including data selection ratio and high-entropy token ratio.

dataset baseline. Meanwhile, the lowest-HES 20% subset performs drastically worse than random selection, confirming that the metric accurately isolates harmful training noise.

Analysis on Efficiency and Transferability. To assess the efficiency of HES for large-scale data curation, we investigate whether small, computationally efficient models can serve as effective proxies for selecting data for larger foundation models. We utilize a lightweight 0.6B model to screen the training data and then train the larger 8B model on the selected subset, comparing the results against the 8B model's self-selection. As shown in Table 4, the performance achieved using the 0.6B proxy model is comparable to that of the 8B model's self-selection (30.20% vs. 31.34%), reducing the inference cost for data selection by over an order of magnitude without compromising data quality. This strong cross-model consistency suggests that HES captures the intrinsic reasoning complexity inherent to the data rather than model-specific artifacts, confirming its value as a cost-effective metric for massive-scale pipelines.

Comparable Analysis with Other Entropy Metrics. We compare HES against alternative entropy-based metrics in Table 5. HES consistently outperforms all alternatives: averaging metrics (AvgE, AvgHE) underperform by diluting sparse reasoning signals with abundant low-entropy tokens, and while ES is competitive, HES achieves superior results by strictly targeting the top percentile of uncertainty. Moreover, HES$_{absolute}$ also underperforms the relative variant as its fixed threshold cannot adapt to the varying scales across different samples, confirming the importance of the adaptive, percentile-based formulation. These results validate that the cumulative magnitude of critical forking points, captured adaptively, is the most effective proxy for reasoning quality.

Sensitivity Analysis. To validate HES's robustness, we analyze its sensitivity to two key hyperparameters in Figure 3: data selection ratio and high-entropy token ratio across diverse domains. We find peak performance is almost consistently achieved when using approximately 20% or 80% of the data as ranked by HES, confirming that pruning the lowest-quality samples is a reliable strategy for improvement. While HES is robust to the specific high-entropy token ratio used, smaller, more targeted ratio of 0.005 delivers the best performance.

4.3 RFT Experiments

Main Results. We evaluate the robustness of HES across different selection scopes in RFT by investigating both localized *Per-Query* and broader *Global Pool Selection*, as detailed in Table 6. HES demonstrates consistent superiority over random baselines in both settings, proving its robustness regardless of the selection scope. Notably, the *Per-Query Selection* often yields higher performance

Table 6: Main results of RFT experiments across different candidate pool sizes (k). **Bold** indicates the best.

Method	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Per-Query Selection ($k = 2$)								
Random	46.46	34.17	31.88	**28.54**	21.46	35.63	4.31	28.92
Length	46.04	33.33	33.75	28.13	19.58	35.94	4.81	28.80
Lowest-HES	43.75	34.17	30.63	25.00	18.96	33.81	4.44	27.25
Highest-HES	**48.33**	**34.58**	**34.38**	28.13	**21.67**	**38.06**	**5.56**	**30.10**
Global Pool Selection ($k = 2$)								
Random	42.29	31.04	30.00	**26.88**	18.54	30.81	4.75	26.33
Length	45.21	**35.42**	27.92	25.21	18.33	32.13	5.13	27.05
Difficulty	45.00	32.71	28.13	23.54	17.71	30.13	4.31	25.93
Lowest-HES	19.38	14.58	10.21	11.04	7.92	10.81	2.00	10.85
Highest-HES	**46.46**	34.58	**32.50**	25.42	**20.83**	**34.00**	**5.31**	**28.44**
Per-Query Selection ($k = 4$)								
Random	48.33	31.88	33.33	27.71	18.75	36.00	4.50	28.64
Length	46.04	33.33	**33.75**	28.13	19.58	35.94	4.81	28.80
Lowest-HES	43.75	33.54	30.42	26.46	**21.67**	32.69	4.75	27.61
Highest-HES	**48.96**	**36.25**	**33.75**	**29.17**	21.04	**37.13**	**5.13**	**30.20**
Global Pool Selection ($k = 4$)								
Random	45.42	**38.13**	**30.83**	**26.04**	20.21	33.06	4.63	28.33
Length	45.21	35.42	27.92	25.21	18.33	32.13	5.13	27.05
Difficulty	45.00	32.71	28.12	23.54	17.71	30.13	4.31	25.93
Lowest-HES	21.46	18.96	12.92	11.04	9.38	11.56	1.69	12.43
Highest-HES	**48.96**	33.13	30.00	24.38	**21.88**	**36.19**	**5.56**	**28.59**
Per-Query Selection ($k = 8$)								
Random	49.17	34.58	29.38	28.75	**22.08**	**36.69**	4.69	29.33
Length	46.67	35.42	32.29	**30.21**	20.63	35.94	5.63	29.54
Lowest-HES	42.50	32.50	**32.71**	28.54	20.00	34.69	4.88	27.97
Highest-HES	**50.42**	**37.92**	30.42	28.96	**22.08**	35.31	**5.81**	**30.13**
Global Pool Selection ($k = 8$)								
Random	46.67	35.42	31.67	28.33	18.54	33.06	4.19	28.27
Length	48.75	36.04	30.42	26.46	**22.08**	36.44	**5.56**	29.39
Difficulty	44.38	32.29	31.04	24.79	20.42	31.50	4.63	27.01
Lowest-HES	25.63	21.25	18.96	14.66	11.67	14.50	2.44	15.59
Highest-HES	**48.96**	**36.46**	**31.88**	**29.17**	19.38	**38.31**	5.19	**29.91**

than the corresponding *Global Pool Selection*. We attribute this advantage to the structural guarantee of query coverage in the *Per-Query Selection*, which suggests that while HES filters for intrinsic quality, maintaining data diversity remains equally essential for maximizing model generalization.

Analysis on Scalability. We scale the candidate pool size (k) from 2 to 8 to examine HES in sample-rich environments. HES consistently outperforms random selection, achieving Per-Query gains of +1.18, +1.56, and +0.80 points for $k = 2, 4, 8$ respectively. The stable efficacy across expanding search spaces confirms HES as a reliable signal for identifying high-quality trajectories regardless of candidate volume.

4.4 RL Experiments

Main Results. We evaluate an asymmetric sampling strategy that pairs Highest-HES positive samples with randomly selected negative ones, against baselines defined in Table 1. As shown in Table 7, *Pos-High, Neg-Rand* achieves the best average accuracy of 19.26%, significantly outperforming random sampling (17.51%) and even surpassing the Full-Batch baseline (18.34%) with only 50% of the data per step. This confirms that prioritizing high-quality positives is a more potent training signal than using larger, undifferentiated batches.

Analysis on Necessity of Negative Diversity. Complementing our positive selection strategy, we examine whether curating the negative sample pool yields benefits similar to prioritizing high-quality

Table 7: RL results comparing different sampling strategies. **Bold** indicates the best performance per benchmark.

Method	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Full-Batch	33.33	25.63	17.30	14.00	**15.21**	19.69	3.19	18.34
Pos-Rand, Neg-Rand	32.30	25.21	13.54	14.11	13.96	19.18	4.25	17.51
Pos-Rand, Neg-Low	33.75	24.79	15.42	13.73	11.67	19.00	3.38	17.39
Pos-Low, Neg-Rand	30.42	24.17	15.63	13.10	14.17	19.50	3.25	17.18
Pos-High, Neg-Low	31.25	25.42	15.21	13.96	12.71	18.88	3.56	17.28
Pos-Length, Neg-Rand	31.04	26.25	16.46	13.95	14.79	**20.00**	4.06	18.08
Pos-Difficulty, Neg-Rand	35.00	24.79	16.88	13.77	14.17	18.94	3.75	18.19
Pos-High, Neg-Rand	**35.42**	**27.29**	**17.92**	**18.13**	11.88	19.88	**4.31**	**19.26**

positives. We compare strategies that constrain negative samples to specific quality tiers (e.g., Neg-Low) against the random negative sampling approach. Strategies that constrain the negative pool consistently degrade performance, falling even below the random baseline, indicating that restricting negative samples to the lowest-quality tier proves counterproductive. In contrast, random negative sampling proves essential for exposing the model to a diverse, unbiased distribution of failure modes, which is required to build robust generalization.

5 Related Work

Training-Time Efficiency. Improving training-time efficiency is a critical challenge across the dominant paradigms for LLM reasoning: SFT, RFT, and RL. To address this, some works focus on curating higher-quality training sets through methods like verification-based selection or preference optimization (Hu et al., 2022; Rein et al., 2024; Yang et al., 2025; Yuan et al., 2023b). Other works aim to provide more granular learning signals by directly supervising intermediate steps with process-level rewards (Lightman et al., 2023; Cui et al., 2025; Luo et al., 2025b). While powerful, the effectiveness of these methods hinges on a reliable signal to identify which reasoning paths or steps are valuable, and they often rely on costly external reward models or human annotations. Our work addresses this need by introducing HES, a simple, intrinsic metric that provides this critical signal uniformly across all three paradigms.

Data Selection. Data selection aims to identify optimal training subsets from large-scale datasets (Albalak et al., 2024; Li et al., 2024a). Traditional approaches rely on heuristic rules (Robertson et al., 2009; Xie et al., 2023) or global metrics like perplexity (Wettig et al., 2024; Marion et al., 2023). While gradient-based methods offer finer-grained selection (Killamsetty et al., 2021; Han, 2023), their computational costs scale prohibitively with sequence length and model size. Alternative approaches using task-specific models (SHUM et al., 2025) or LLM-based selection (Toshniwal et al., 2025) introduce substantial overhead. Current data selection thus faces two challenges: (1) the inability to identify crucial tokens through global metrics, and (2) the lack of efficient metrics deployable across different training paradigms. Our HES addresses both limitations.

6 Conclusion

In this paper, we introduced **High-Entropy Sum (HES)**, a novel, training-free metric designed to efficiently identify high-quality reasoning data. We proposed a unified data selection framework based on HES and validated its effectiveness and efficiencyacross three dominant training paradigms: SFT, RFT, and RL. In conclusion, our work established HES as a powerful, cost-effective solution for scaling up LLM reasoning capabilities, highlighting the pivotal role of data selection.

Limitations and future work. While HES proves effective across SFT, RFT, and RL, several limitations remain. Although we validate HES on Math, Code, and STEM domains, its applicability to other reasoning-intensive tasks such as agentic planning and open-ended generation have not been explored. And our current framework treats HES as a static score computed once before training; integrating HES as a dynamic, curriculum-based signal that adapts during training remains an open direction.

AI USE STATEMENT

During the writing process, we used LLMs to assist with language polishing and proofreading. All LLM-assisted outputs were carefully reviewed and verified by the human authors. The authors take full responsibility for the entire content of this paper, including the accuracy of all claims and the final wording of the text.

ETHICS STATEMENT

This work focuses on improving the training efficiency of LLMs through data selection and does not involve the collection or use of private, sensitive, or personally identifiable data. All datasets used in our experiments are publicly available and have been widely adopted by the research community. Our method is designed to enhance reasoning quality by filtering low-quality training data, which may also help reduce the risk of models learning from flawed or misleading reasoning paths. We do not foresee direct negative societal impacts from this work, though we acknowledge that improvements in LLM reasoning capabilities should be accompanied by responsible deployment practices.

REPRODUCIBILITY STATEMENT

The methodology describes the HES metric and its selection procedures for SFT, RFT, and RL. Section 4.1 presents the experimental setup. Appendix C provides training configurations, Appendix D describes the datasets, and Appendix E reports computational resources. Supplementary experimental results are provided in Appendix G.

REFERENCES

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

Art of Problem Solving. 2024 AIME I, 2024a. URL `https://artofproblemsolving.com/wiki/index.php/2024_AIME_I`.

Art of Problem Solving. 2024 AIME II, 2024b. URL `https://artofproblemsolving.com/wiki/index.php/2024_AIME_II`. Accessed: 2025.

Art of Problem Solving. 2025 AIME I, 2025a. URL `https://artofproblemsolving.com/wiki/index.php/2025_AIME_I`. Accessed: 2025.

Art of Problem Solving. 2025 AIME II, 2025b. URL `https://artofproblemsolving.com/wiki/index.php/2025_AIME_II`. Accessed: 2025.

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.

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, pp. 17682–17690, 2024.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. *arXiv preprint arXiv:2508.15260*, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023.

Xiaochuang Han. In-context alignment: Chat with vanilla language models before fine-tuning. *arXiv preprint arXiv:2308.04275*, 2023.

HMMT. HMMT 2023, 2023. URL https://www.hmmt.org/. Accessed: 2025.

HMMT. HMMT 2024, 2024. URL https://www.hmmt.org/. Accessed: 2025.

HMMT. HMMT 2025, 2025. URL https://www.hmmt.org/. Accessed: 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Hamish Ivison, Noah A Smith, Hannaneh Hajishirzi, and Pradeep Dasigi. Data-efficient finetuning using cross-task nearest neighbors. *arXiv preprint arXiv:2212.00196*, 2022.

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.

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. URL https://openreview.net/forum?id=chfJJYC3iL.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pp. 5464–5474. PMLR, 2021.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024a.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7595–7628, 2024b.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025a.

Ruilin Luo, Zhuofan Zheng, Yifan Wang, Yiyao Yu, Xinzhe Ni, Zicheng Lin, Jin Zeng, and Yujiu Yang. Ursa: Understanding and verifying chain-of-thought reasoning in multimodal mathematics. *arXiv preprint arXiv:2501.04686*, 2025b.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. Skeleton-of-thought: Large language models can do parallel decoding. *Proceedings ENLSP-III*, 2023.

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. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

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.

Matthew Renze and Erhan Guven. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*, 2024.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Anusha Sabbineni, Nikhil Anand, and Maria Minakova. Comprehensive benchmarking of entropy and margin based scoring metrics for data selection. *arXiv preprint arXiv:2311.16302*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Bilgehan Sel, Ahmad Al-Tawaha, Vanshaj Khattar, Ruoxi Jia, and Ming Jin. Algorithm of thoughts: Enhancing exploration of ideas in large language models. *arXiv preprint arXiv:2308.10379*, 2023.

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shariari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, et al. Bond: Aligning llms with best-of-n distillation. *arXiv preprint arXiv:2407.14622*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

KaShun SHUM, Yuzhen Huang, Hongjian Zou, dingqi, YiXuan Liao, Xiaoxin Chen, Qian Liu, and Junxian He. Predictive data selection: The data that predicts is the data that teaches. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=tTVYR82Iz6.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *arXiv preprint arXiv:2503.21380*, 2025.

Shubham Toshniwal, Ivan Sorokin, Aleksander Ficek, Ivan Moshkov, and Igor Gitman. Genselect: A generative approach to best-of-n. In *2nd AI for Math Workshop @ ICML 2025*, 2025. URL https://openreview.net/forum?id=8LhnmNmUDb.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*, 2024.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227, 2023.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Yiyao Yu, Yuxiang Zhang, Dongdong Zhang, Xiao Liang, Hengyuan Zhang, Xingxing Zhang, Ziyi Yang, Mahmoud Khademi, Hany Awadalla, Junjie Wang, et al. Chain-of-reasoning: Towards unified mathematical reasoning in large language models via a multi-paradigm perspective. *arXiv preprint arXiv:2501.11110*, 2025.

Z Yuan, H Yuan, C Li, G Dong, K Lu, C Tan, C Zhou, and J Zhou. Scaling relationship on learning mathematical reasoning with large language models (arxiv: 2308.01825). arxiv, 2023a.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. Scaling relationship on learning mathematical reasoning with large language models. *CoRR*, abs/2308.01825, 2023b. doi: 10.48550/ARXIV.2308.01825. URL https://doi.org/10.48550/arXiv.2308.01825.

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.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

A Broader Impact

This paper presents a unified, computation-efficient framework for data selection in LLM reasoning. Our work has several broader impacts on the community:

Environmental Sustainability. The massive computational cost of training LLMs poses significant environmental challenges. By demonstrating that training on a small, rigorously curated subset (e.g., 20%) of data can match or even exceed the performance of full-dataset training, our method directly contributes to reducing the energy consumption and carbon footprint associated with model post-training. Furthermore, unlike methods relying on training expensive Reward Models, our proposed HES metric is training-free and computationally lightweight, further minimizing the resource overhead of the data curation pipeline itself.

Democratization of Research. A key finding of our work is the effective small-to-large transferability, where small models (e.g., 0.6B) can serve as accurate proxies to select data for large foundation models. This capability significantly lowers the barrier to entry for academic laboratories and smaller organizations with limited computing resources, empowering a broader community to contribute to high-quality dataset construction and model development without requiring massive GPU clusters.

Model Reliability. By effectively identifying and removing low-quality data—which often contains hallucinations, inconsistent logic, or noise—our method contributes to building more reliable and robust reasoning systems. Reducing a model's exposure to flawed reasoning paths during training helps mitigate the risk of low-quality reasoning, where models generate plausible-sounding but factually or logically incorrect outputs.

B Detailed Related Work

Efficient Reasoning. Recent progress on efficient reasoning has advanced along two directions: (1) test-time efficiency (Jaech et al., 2024; Snell et al., 2024) and (2) training-time efficiency. In the former, mainstream methods broaden the explored trajectory set by deepening CoT and branching into multiple paths (Wei et al., 2022; Sessa et al., 2024; Yao et al., 2023; Besta et al., 2024; Sel et al., 2023; Ning et al., 2023; Fu et al., 2025). Building on this, some work incorporates tool use, heuristic control, or self-reflection to verify processes or outcomes and stabilize search (Chen et al., 2022; Gou et al., 2023; Yu et al., 2025; Renze & Guven, 2024). However, these methods often require substantial token budgets. This motivates efforts to compress outputs by extracting key information from reasoning trajectories (Zhang et al., 2025; Xu et al., 2025; Aytes et al., 2025). On the training-time side, the focus is on efficiently converting exploration-derived signals into policy updates. Dominant pipelines improve reasoning by verification-based selection or preference optimization (Hu et al., 2022; Rein et al., 2024; Yang et al., 2025; Yuan et al., 2023b). In addition, some work explicitly targets process quality by supervising intermediate steps with process-level signals (Lightman et al., 2023; Cui et al., 2025; Luo et al., 2025b). Therefore, efficient reasoning either allocates more computation to critical branches at test time or concentrates training budgets on samples or spans with higher pedagogical value at training time. Both require a stable signal that reliably identifies critical fork tokens. Our work is mainly focused on training-time efficiency, and we leave the exploration of test-time efficiency to future work.

Data Selection. Data selection takes the full training data as input and chooses a subset to train (Albalak et al., 2024). Large-scale, high-quality datasets play a decisive role in training effectiveness (Li et al., 2024a; Penedo et al., 2024). Traditional selection methods rely on rules and heuristics, which make it difficult to assess the quality and complexity of reasoning processes (Robertson et al., 2009; Xie et al., 2023). Therefore, many works adopt global proxies such as perplexity, average token entropy, sample length, or heuristic difficulty (Wettig et al., 2024; Marion et al., 2023; Ivison et al., 2022). Yet these metrics inevitably mix routine low-entropy spans with the sparse high-entropy tokens that signal critical decision points, which dilutes the signal and makes it hard to distinguish correct solutions with low process complexity from those with high process complexity. More fine-grained strategies rely on selection criteria derived from gradients or influence estimates (Killamsetty et al.,

2021; Han, 2023; Xia et al., 2024). However, their compute and memory costs scale rapidly with sequence length and model size, which limits their practicality for large-scale training. Consequently, current data selection faces two core gaps: (1) the lack of a model-intrinsic signal that directly captures process-level critical decision points; (2) the lack of a metric that can be deployed consistently and efficiently across diverse training methods without introducing new supervision. Our proposed HES addresses both gaps.

C Training Details

SFT Training Details. Our implementation is based on the `open-r1` (Hugging Face, 2025) framework, and we follow the recommended SFT hyperparameters. We use the AdamW optimizer with a learning rate of 4×10^{-5}. The learning rate follows a cosine decay schedule with a warm-up ratio of 0.1. All experiments are run with a global batch size of 64 and are trained for a total of 3 epochs.

RFT Training Details. First, we sample 32 candidate responses using the model for each query in the dataset. Then, we filter these candidates to retain only the correct trajectories. This creates a pool which serves as the foundation for selection. Training settings, such as the optimizer and learning rate schedule, remain consistent with SFT, with the only difference being the curated training dataset.

RL Training Details. Our implementation is built upon the `verl` codebase (Sheng et al., 2025) and follows the standard GRPO settings. Consistent with the DeepScaleR-1.5B-Preview (Luo et al., 2025a) setup, the maximum generation length is set to 8192 tokens. We use a temperature of 0.6 and generate 32 rollouts per query. We train for 3 epochs, a total of 628 steps. It is worth noting that we train the baseline to its officially reported accuracy.

D Datasets

Mixture-of-Thoughts (Hugging Face, 2025) is a comprehensive compilation of high-quality chain-of-thought reasoning data, specifically assembled to enhance model capabilities across diverse logic-intensive domains, including Math, Code and STEM domain. This multi-domain dataset is aggregated from several rigorously selected sources. It features a strong foundation in mathematical reasoning, incorporating the challenging OpenR1-Math and Open-Math-Reasoning collections. To ensure cross-domain generalization and robustness, the collection is significantly broadened with specialized subsets from other fields, including complex programming tasks from the Codeforces-CoT dataset and scientific reasoning problems from the STEM subset of the Llama-Nemotron dataset.

DeepScaler (Luo et al., 2025a) is a curated collection of approximately 40,000 unique mathematics problem-answer pairs, specifically assembled to train advanced problem-solving models. This comprehensive dataset is compiled from a variety of high-quality and challenging sources. It features a deep historical archive of problems from prestigious competitions, including the American Invitational Mathematics Examination (AIME) from 1984 to 2023 and the American Mathematics Competition (AMC) from years prior to 2023. To ensure breadth and diversity, the collection is further supplemented with content from the Omni-MATH and Still datasets.

OpenR1-Math-220k (Hugging Face, 2025) is a massive dataset designed to advance mathematical reasoning in LLMs. It comprises 220,000 math problems sourced from the NuminaMath 1.5 collection. A key feature of this dataset is that each problem is accompanied by two to four distinct reasoning traces, or step-by-step solutions, generated by the DeepSeek R1 model. To ensure reliability, these traces underwent a rigorous verification process: most were validated using Math Verify, while 12% were assessed by a Llama-3.3-70B-Instruct judge. Critically, every problem in the dataset is guaranteed to contain at least one reasoning trace that leads to a correct answer. The dataset is organized into two distinct splits: (1) The default split, containing 94,000 problems, is the recommended version for training. It has been shown to yield the best performance improvements after SFT. (2) The extended split expands the collection to 131,000 problems by incorporating additional data sources. While this provides a greater volume of reasoning traces, models fine-tuned on this split have shown slightly lower performance, which is likely because the questions from the added sources are less difficult than those in the core dataset.

Table 8: Performance comparison of SFT using Qwen3-8B-Base on OpenR1-Math-220k.

Method	Ratio	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Fullset	100	**48.96**	32.50	31.67	25.83	22.50	38.38	5.81	29.38
Random	20	30.00	30.21	24.17	19.79	17.50	25.00	5.56	21.75
Highest-AvgE	20	39.38	26.67	22.29	19.38	18.75	26.38	4.38	22.46
Highest-AvgHE	20	39.58	29.58	23.54	18.75	20.00	29.13	3.63	23.46
Lowest-HES	20	15.42	12.29	6.04	8.13	6.04	5.63	2.13	7.95
Highest-HES	20	48.54	32.92	30.63	23.33	22.29	37.94	**6.56**	28.89
Highest-HES	80	**48.96**	**35.63**	**36.04**	**26.67**	**22.71**	**41.88**	6.00	**31.13**

Open-Math-Reasoning (Moshkov et al., 2025) is also a large-scale dataset created to train LLMs in advanced mathematical reasoning. At its core, it contains 306,000 unique mathematical problems sourced from the Art of Problem Solving (AoPS) forums. These are complemented by a massive volume of solutions, including 3.2 million long-CoT examples and 1.7 million Tool-Integrated Reasoning (TIR) solutions, showcasing diverse and detailed problem-solving paths. The dataset also includes 566,000 "GenSelect" samples, which are designed to teach models how to evaluate and select the most promising solution from multiple candidates. To further expand its utility, an additional 193,000 problems from the AoPS forums are included without solutions. The creation process involved state-of-the-art models, with Qwen2.5-32B-Instruct used for preprocessing problems and both DeepSeek-R1 and QwQ-32B for generating the high-quality solutions. Notably, this dataset was a foundational component of the winning submission to the AIMO-2 Kaggle competition, highlighting its power and effectiveness in building top-tier AI reasoning systems.

To maintain a balanced distribution and manageable computational cost across all domains, we constructed the *Full-Dataset* by randomly sampling approximately 100,000 examples from the source collections of each domain (Math, Code, and STEM). This standardized subset serves as the baseline for all comparative experiments.

E Computational Resources

All experiments were conducted on a single node equipped with 8× NVIDIA A100 80GB GPUs, 224 CPU cores, and 2048 GB of system memory. The total computational cost of all experiments reported in this paper amounts to approximately 500 GPU hours.

F Licenses

Datasets: OpenR1-Math-220k (Hugging Face, 2025) is released under the Apache 2.0 License. Open-Math-Reasoning (Hugging Face, 2025) is released under the CC-BY-4.0 License. DeepScaleR (Luo et al., 2025a) is released under the MIT License. Mixture-of-Thoughts (Hugging Face, 2025) is released under the Apache 2.0 License.

Models: Qwen3-8B-Base (Yang et al., 2025), Qwen3-4B-Base (Yang et al., 2025), and Qwen3-0.6B (Yang et al., 2025) are released under the Apache 2.0 License. DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025) and DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025) are released under the MIT License. DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI, 2025) is released under the MIT License and additionally governed by the Llama 3.1 Community License Agreement.

All datasets and models are used in this work for their originally intended purposes, i.e., training and evaluating LLMs on reasoning tasks, which is fully consistent with the terms of their respective licenses.

G Supplementary Results

To further validate the robustness and generalizability of the HES metric, we conducted additional experiments across different datasets, model sizes, and architectures. This section presents detailed results that complement the main findings.

Table 9: Performance comparison of SFT using Qwen3-4B-Base on Open-Math-Reasoning.

Method	Ratio	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Fullset	100	39.17	31.04	29.58	24.17	20.83	37.00	4.63	26.63
Random	20	30.83	26.04	20.42	16.67	17.08	20.06	3.25	19.19
Highest-AvgE	20	34.58	28.13	21.46	14.79	17.92	23.63	3.06	20.51
Highest-AvgHE	20	33.75	25.42	21.67	17.29	15.83	23.56	2.19	19.96
Lowest-HES	20	16.04	16.46	9.17	7.08	6.46	7.00	2.13	9.19
Highest-HES	20	38.13	31.88	22.71	20.42	18.54	28.00	**4.88**	23.51
Highest-HES	80	**46.25**	**33.96**	**31.88**	**25.83**	**21.46**	**39.44**	**5.13**	**29.14**

Table 10: Performance comparison of SFT using DeepSeek-R1-Distilled-Qwen-7B on OpenR1-Math-220k. **Bold** indicates the best performance per benchmark.

Method	Ratio	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Full-Dataset	100	46.25	33.13	31.67	26.25	22.29	38.50	5.50	29.08
Random	20	42.08	37.08	29.79	26.67	23.13	38.13	5.13	28.86
Lowest-HES	20	29.38	27.29	17.50	18.33	14.58	19.25	3.06	18.48
Highest-HES	20	**51.67**	**40.83**	33.96	**30.21**	24.58	**45.75**	**6.88**	**33.41**
Highest-HES	80	50.42	35.21	35.21	26.25	24.17	42.25	5.44	31.28

G.1 Consistency Across Datasets and Model Scales

We replicated the SFT experiments on the OpenR1-Math-220k dataset using the Qwen3-8B-Base model (Table 8) and on the Open-Math-Reasoning dataset using the smaller Qwen3-4B-Base model (Table 9).

The results consistently demonstrate the efficacy of HES-based data selection:

- **Superiority of Highest-HES:** In both settings, training on the top 20% of data ranked by HES (Highest-HES) achieves performance comparable to or better than random selection, often approaching the full-dataset baseline.
- **Effective Noise Filtering:** Training on the top 80% of data (i.e., pruning the bottom 20% with the lowest HES) consistently yields the best overall performance. For instance, on OpenR1-Math-220k, the Highest-HES (80) strategy achieves an average accuracy of 31.13%, surpassing the full-dataset performance of 29.38%. Similarly, on Open-Math-Reasoning with the 4B model, the top 80% subset achieves an average of 29.14%, outperforming the full dataset's 26.63%.
- **Detrimental Effect of Low-HES Data:** Consistent with our main findings, training on the bottom 20% of data ranked by HES (Lowest-HES) leads to a catastrophic drop in performance, significantly underperforming even the random baseline. This confirms that HES effectively identifies low-quality data that can be harmful to model training.

G.2 Robustness Across Model Architectures

To ensure that the effectiveness of HES is not limited to the Qwen model family, we conducted RFT experiments using the DeepSeek-R1-Distilled-Llama-8B model (Table 11). We compared random selection, HES-based selection, and inverse HES selection across both per-query and global pool settings with $k = 4$.

The results on the Llama-based model align with our previous findings:

- **HES Outperforms Random Selection:** In both per-query and global pool settings, selecting the highest HES responses consistently yields better average performance than random selection. For example, in the per-query setting, Highest-HES achieves an average of 13.23%, compared to 12.42% for random selection.
- **Criticality of Global Pool Selection:** The discriminative power of HES is particularly evident in the global pool setting. While random selection achieves 11.85%, Highest-HES

Table 11: Performance comparison of RFT strategies using DeepSeek-R1-Distilled-Llama-8B.

Method	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Per-Query Selection ($k = 4$)								
Random	17.08	**22.50**	10.63	**12.71**	9.17	12.38	2.44	12.42
Lowest-HES	14.58	18.13	12.08	10.21	9.38	12.06	2.06	11.21
Highest-HES	**18.54**	19.38	**14.79**	10.83	**11.88**	**14.31**	**2.88**	**13.23**
Global Pool Selection ($k = 4$)								
Random	14.17	19.58	11.04	**11.46**	10.56	12.81	**3.35**	11.85
Lowest-HES	1.46	10.00	5.21	5.21	2.92	3.56	1.31	4.24
Highest-HES	**17.92**	**20.63**	**14.38**	11.25	**12.29**	**15.19**	3.13	**13.54**

Length Group	Selection	AIME24	AIME25	HMMT23	HMMT24	HMMT25	Oly(E)	Oly(H)	AVG
Long	Lowest-HES	35.00	30.21	**31.88**	**22.92**	**20.83**	28.88	**4.63**	24.91
	Highest-HES	**40.42**	**31.88**	26.67	22.71	18.75	**31.38**	4.50	**25.19**
Medium	Lowest-HES	26.67	23.96	17.08	13.54	16.04	16.25	2.88	16.63
	Highest-HES	**40.63**	**29.17**	**25.63**	**23.75**	**19.17**	**28.44**	**3.38**	**24.31**
Short	Lowest-HES	11.88	10.00	1.67	6.46	1.67	5.63	1.19	5.50
	Highest-HES	**28.13**	**20.42**	**17.08**	**12.08**	**12.50**	**16.75**	**2.00**	**15.57**

Table 12: Performance comparison of HES under different length groups. The Math subset is stratified into Long, Medium, and Short groups based on token length. Within each group, we compare the top 20% (Highest-HES) versus the bottom 20% (Lowest-HES) selections trained on Qwen3-8B. **Bold** indicates the better performance within each length group.

improves this to 13.54%. Conversely, selecting the lowest HES responses results in a severe performance degradation to 4.24%, highlighting the metric's ability to distinguish high-quality reasoning paths from poor ones across different model architectures.

G.3 Decoupling Quality from Length

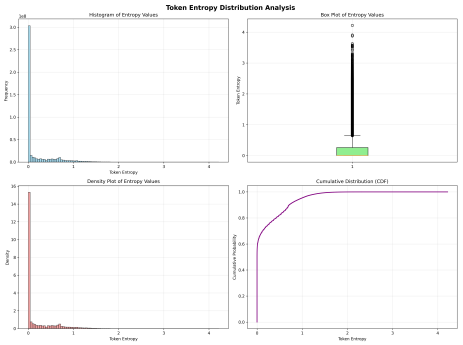
To ensure that HES captures true reasoning density rather than merely acting as a proxy for length, we analyze its performance independent of token count. We stratify the Math subset into Long, Medium, and Short groups and compare the performance of the top 20% (Highest-HES) versus the bottom 20% (Lowest-HES) within each length interval, as shown in Table 12. Highest-HES consistently outperforms Lowest-HES across all length groups, confirming that the metric functions independently of sample length. The discrimination power is most pronounced in the Short group, where Highest-HES (15.57%) nearly triples the accuracy of Lowest-HES (5.50%), indicating that identifying critical steps remains crucial even for shorter reasoning chains. In the Long group, Highest-HES maintains a distinct advantage despite the inherent complexity of longer samples, proving that HES effectively captures thought density rather than simply favoring longer sequences.

These supplementary results reinforce the conclusion that HES is a robust, model-agnostic metric for data selection in LLM reasoning, capable of improving performance across various datasets, model scales, and architectures.

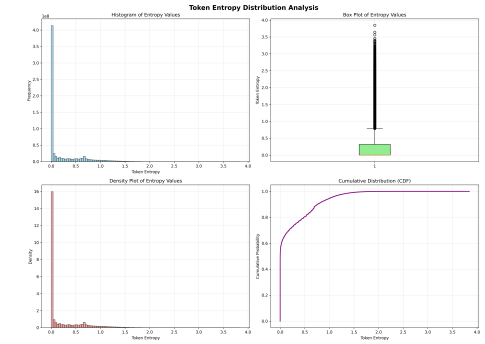
H Analysis

H.1 Discriminative Ability of HES

To validate the effectiveness of HES as a quality metric, we analyze its ability to discriminate between high- and low-quality samples against several alternative entropy-based metrics. This analysis is conducted across multiple benchmarks, including AIME24, HMMT23, HMMT24, and HMMT25. As illustrated by the score distributions in Figure 7 to Figure 10, the HES demonstrates a significantly clearer separation between correct and incorrect reasoning paths compared to other metrics. Specifically, the distributions of HES scores for correct and incorrect samples show minimal overlap. The mean HES for incorrect paths is substantially higher than for correct paths, indicating that flawed reasoning corresponds to a higher cumulative sum of uncertainty at pivotal moments.

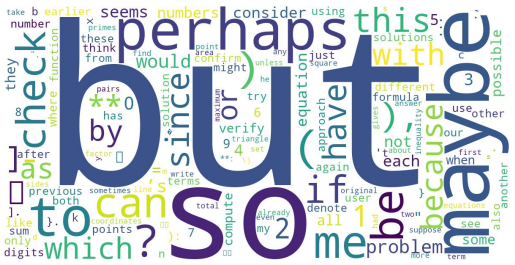


(a) Token entropy distribution of OpenR1-Math-220k.

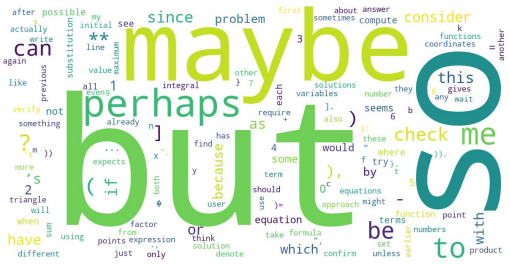


(b) Token entropy distribution of Open-Math-Reasoning.

Figure 4: Wordcloud of high-entropy tokens.



(a) Wordcloud of OpenR1-Math-220k.



(b) Wordcloud of Open-Math-Reasoning.

Figure 5: Wordcloud of high-entropy tokens.

In contrast, global metrics like the AveE and even the more targeted AvgHET show less separation between their means for correct and incorrect samples. This analysis confirms that the cumulative signal captured by HES is a more robust and reliable indicator of reasoning quality than metrics based on either global averages or the average intensity of pivotal moments alone. The clear separation validates its use as a foundational metric for our data selection framework.

Additionally, Figure 6 provides specific entropy value references.

H.2 High-entropy Token Distribution

To better understand the characteristics of our training data, we conduct a detailed analysis of the token-level entropy distributions for both the OpenR1-Math-220k and Open-Math-Reasoning datasets. The analysis, illustrated in Figure 4a and 4, reveals that the entropy distribution is heavily right-skewed. The vast majority of tokens exhibit very low entropy, corresponding to routine or predictable steps, while a small minority of tokens accounts for the high-entropy tail of the distribution. Our calculations show that the entropy value corresponding to the top 0.5% of high-entropy tokens is approximately 1.55 for OpenR1-Math-220k and 1.60 for Open-Math-Reasoning. This data-driven finding provides an empirical basis for our threshold choice of 1.6 in the $HES_{absolute}$ experiments. Furthermore, a qualitative analysis using word clouds shown in Figure 5 confirms that these high-entropy tokens are not random noise; they typically correspond to semantically significant forking points that guide the reasoning process. Taken together, these findings validate our core premise: a small, identifiable subset of tokens represents the most complex decision points, motivating our focus on the HES metric.

H.3 Token Distribution Example

The entropy distribution of a sample is shown as follows. We can see that high entropy words are small in number of forking points, while low entropy words are large in number.

I Computational Efficiency and Cost Analysis

A primary concern in data selection is whether the cost of computing the metric outweighs the benefits of reduced training data. In this section, we analyze the computational overhead of HES in both online generation (RFT/RL) and offline screening (SFT) scenarios, demonstrating that HES is a highly cost-effective metric.

I.1 Online Settings (RFT and RL): Zero Extra Inference Overhead

In RFT and RL, the model generates reasoning trajectories (rollouts) in real-time. In these scenarios, calculating HES incurs **zero extra inference overhead** in terms of model forward passes.

- **Piggybacking on Generation:** During the auto-regressive generation process, the model must perform a forward pass at every time step t to compute the probability distribution over the vocabulary, $P(x_t|x_{<t})$, in order to sample the next token.
- **Synchronous Calculation:** The entropy calculation is performed synchronously using these intermediate logits. We simply calculate the entropy of the sampled token's distribution and accumulate it in memory. This operation involves only negligible vector operations (complexity $O(V)$, where V is vocabulary size), which is orders of magnitude faster than the matrix multiplications required for the model's forward pass.
- **No Additional Forward Pass:** Unlike methods that require a separate reward model or a second pass over the completed sequence, HES reuses the computation already performed for generation. Thus, it does not increase the inference latency or compute budget during rollout.

I.2 Offline Settings (SFT): Efficient Screening via Proxy Models

For SFT, data selection is typically performed offline on static datasets. While calculating HES requires a forward pass to obtain logits, we demonstrate that this cost can be minimized by using smaller proxy models.

Small Model Transferability: We investigated whether HES scores derived from smaller models correlate with data quality for larger models. As shown in Table 4, we used a Qwen3-0.6B model to screen data for training a Qwen3-8B model.

- **Performance Consistency:** The 8B model trained on data selected by the 0.6B proxy achieved an average accuracy of 30.20%, comparable to the performance when using the 8B model for selection (31.34%).
- **Cost Reduction:** The inference cost of a 0.6B model is approximately $1/13$th that of an 8B model. By using a small proxy model, we can screen massive datasets at a fraction of the cost.

I.3 Overall Return on Investment (ROI)

The computational cost of HES selection should be weighed against the savings in training compute.

- **Training vs. Inference:** The computational cost of training (backward pass + optimizer) is significantly higher than inference (forward pass only).
- **Data Pruning:** Our experiments show that training on the top 20% of HES-selected data matches or exceeds full-dataset performance. This represents an **80% reduction** in total training FLOPs.

In summary, the inference cost of calculating HES (especially when using small proxy models) is negligible compared to gradient updates on 80% of low-value data, which makes HES a highly efficient strategy.

J Additional Analysis of HES and Response Length

This section analyzes the relationship between HES and response length, compares selection strategies across post-training settings, and presents additional entropy diagnostics and efficiency measurements.

J.1 Correlation, Selection Overlap, and Token Budget

On OpenMathReasoning, the Spearman correlation between HES and token length is 0.8703, consistent with the cumulative nature of HES. The top-20% subsets selected by HES and length have an overlap of 49.69%, showing that the two criteria select substantially different samples despite their strong correlation.

Table 13 compares selected token totals and downstream performance at the same 20% example-retention ratio. HES selects 139.10 million tokens and achieves an average score of 31.34, compared with 180.48 million tokens and 29.05 for length selection. Thus, HES improves the score by 2.29 points while selecting fewer tokens, supporting its utility beyond ranking examples by length alone.

Table 13: Selected token totals and downstream scores for top-20% selection on OpenMathReasoning with Qwen3-8B.

Selection	Selected tokens (millions)	Average score
Highest-HES	139.10	31.34
Length	180.48	29.05

J.2 Sample Counts and Observed Length Ranges

Table 14 presents sample counts, observed full-token ranges, and downstream scores for length-stratified selection. Each selected subset contains 7,409 examples. The minimum and maximum token counts describe the samples selected within each group.

Table 14: Length-stratified selection: subset sizes, observed full-token ranges, and downstream average scores. The high-minus-low score gaps are 10.07, 7.68, and 0.28 for Short, Medium, and Long, respectively.

Bucket	Selection	Examples	Min. tokens	Max. tokens	Avg.
Short	Lowest-HES	7,409	325	2,652	5.50
Short	Highest-HES	7,409	3,406	5,021	15.57
Medium	Lowest-HES	7,409	5,021	16,218	16.63
Medium	Highest-HES	7,409	6,765	23,300	24.31
Long	Lowest-HES	7,409	8,064	24,768	24.91
Long	Highest-HES	7,409	8,064	21,510	25.19

Highest-HES exceeds Lowest-HES in all three groups, with gains of 10.07, 7.68, and 0.28 points for Short, Medium, and Long, respectively. The largest gains occur in the Short and Medium groups, where the selected samples also differ substantially in length. In the Long group, both selections have the same minimum length, and Highest-HES has a smaller maximum length while achieving a slightly higher score. Together with the subset-overlap and token-budget comparisons, these results characterize HES as a selection criterion related to both trajectory length and the distribution of token uncertainty.

J.3 Output-Length Statistics and Length Baselines

Table 15 compares average output lengths under HES and random selection. HES produces shorter outputs in five of the six settings; SFT-Math-20% produces longer outputs. The reduction is largest for SFT-Code-20% at 679.28 tokens, followed by SFT-STEM-20% at 446.47 tokens and RL at 407.18 tokens. These results show that selecting high-HES training examples can yield shorter model outputs across several post-training settings.

Table 15: Average output lengths for HES and random selection. The difference is HES minus Random.

Setting	HES	Random	Difference
SFT-Code-20%	11,874.70	12,553.98	-679.28
SFT-STEM-20%	5,886.50	6,332.97	-446.47
SFT-Math-20%	15,196.90	14,530.91	$+665.99$
SFT-Math-80%	13,309.70	13,656.65	-346.95
RFT $k = 4$ (Per-Query)	18,259.10	18,365.30	-106.20
RL	1,675.70	2,082.88	-407.18

Table 16: Downstream performance of HES and length-based selection. Gains are absolute score differences in percentage points.

Setting / benchmark	Length	HES	Gain
Math average	29.05	31.34	$+2.29$
GPQA	50.82	51.96	$+1.14$
LiveCodeBench	53.84	54.43	$+0.59$
RFT $k = 4$ Per-Query average	28.80	30.20	$+1.40$
RL average	18.08	19.26	$+1.18$

Table 16 shows that HES outperforms length-based selection in all five comparisons. The gains span Math, GPQA, LiveCodeBench, RFT, and RL, supporting the use of high-entropy information across both domains and training paradigms.

K Quantitative Summary of the Entropy Diagnostics

We analyze correct and incorrect trajectories from Qwen3-14B to examine the effect of entropy aggregation. Table 17 gives the difference between the incorrect-trajectory mean and the correct-trajectory mean for each metric, using the score scales of the corresponding diagnostic plots.

Table 17: Mean entropy-score differences between incorrect and correct trajectories from Qwen3-14B, using the scales of the corresponding diagnostic plots.

Metric	AIME25	AIME24	HMMT23	HMMT24	HMMT25
Average entropy (all tokens)	0.01	0.10	-0.01	0.00	-0.01
Average entropy (high-entropy tokens)	0.00	0.02	0.00	0.00	-0.01
Entropy sum (all tokens)	0.30	0.36	0.15	0.18	0.18
High-Entropy Sum (HES)	0.39	0.40	0.24	0.20	0.25

Average-entropy differences are close to zero across the five benchmarks, whereas cumulative entropy measures yield consistently positive differences. HES has the largest numerical mean gap on the displayed scales in each benchmark, reflecting the accumulation of uncertainty at high-entropy decision points. The selection protocols combine this uncertainty signal with correctness verification or positive rewards, as detailed in Appendix O.

L Confidence Intervals and Performance Comparisons

Tables 18–20 present performance comparisons with 95% confidence intervals. Each entry gives the point estimate and interval half-width; gains are differences between point estimates in percentage points.

The Math evaluation uses Average@16, averaging over 16 sampling paths to summarize performance across sampled generations. The tables compare HES against low-HES, random, length-based, and full-data or full-batch alternatives under the corresponding SFT, RFT, and RL settings.

Table 18: SFT performance comparisons with 95% confidence intervals.

Comparison	HES	Baseline	Gain
HES-20% vs. Lowest-HES-20%	31.34 ± 1.36	11.75 ± 1.02	$+19.59$
HES-20% vs. Random-20%	31.34 ± 1.36	23.86 ± 1.32	$+7.48$
HES-20% vs. Length-20%	31.34 ± 1.36	29.05 ± 1.38	$+2.29$
HES-80% vs. Fullset	34.61 ± 1.43	31.83 ± 1.41	$+2.78$

Table 19: RFT performance comparisons with 95% confidence intervals. HES denotes Highest-HES in each selection setting.

Setting and baseline	HES	Baseline	Gain
Per-Query $k = 2$: Random	30.10 ± 1.39	28.92 ± 1.39	$+1.18$
Per-Query $k = 4$: Length	30.20 ± 1.39	28.80 ± 1.37	$+1.40$
Global Pool $k = 2$: Lowest-HES	28.44 ± 1.36	10.85 ± 0.97	$+17.59$
Global Pool $k = 8$: Random	29.91 ± 1.39	28.27 ± 1.34	$+1.64$

Table 20: RL performance comparisons with 95% confidence intervals. The HES strategy is Pos-High, Neg-Rand in every row.

Baseline	HES strategy	Baseline	Gain
Full-Batch	19.26 ± 1.20	18.34 ± 1.18	$+0.92$
Pos-Rand, Neg-Rand	19.26 ± 1.20	17.51 ± 1.17	$+1.75$
Pos-Length, Neg-Rand	19.26 ± 1.20	18.08 ± 1.18	$+1.18$
Pos-High, Neg-Low	19.26 ± 1.20	17.28 ± 1.15	$+1.98$

HES achieves higher point estimates in every comparison. In SFT, selecting the highest-HES 20% improves over the lowest-HES, random, and length-based subsets, while retaining 80% exceeds full-set training. RFT applies HES after correctness filtering and improves performance under both per-query and global-pool selection. In RL with the 1.5B policy, the asymmetric combination of high-HES positives and random negatives yields the highest average among the listed strategies. This pattern supports prioritizing informative successful trajectories while retaining diverse negative examples.

M Proxy Selection and Entropy-Scoring Models

Qwen3-0.6B proxy selection and Qwen3-8B self-selection have a subset overlap of 85.89%, with downstream average scores of 30.20 and 31.34, respectively. The high overlap shows substantial agreement between the selected subsets, while self-selection achieves the higher downstream score. Table 21 summarizes the entropy-scoring source used in each setting.

Table 21: Models and probability sources used to compute HES in each setting.

Setting	Entropy-scoring source
Offline SFT, self-selection	The target model forward-scores the training trajectories.
Offline SFT, proxy selection	A fixed Qwen3-0.6B model scores the data used to train the 8B target model.
RFT	Logits of the model generating the candidate trajectories.
RL	Logits of the current policy during rollout generation.

Self-selection uses the target model to score candidate trajectories, while proxy selection uses the smaller Qwen3-0.6B model. In offline SFT, scoring consists of forward passes over the candidate data. RFT and RL reuse logits produced during trajectory generation, integrating entropy computation with the existing rollout process.

N End-to-End RL Wall-Clock Measurements

Under the same RL setting on $8\times$A100 GPUs, Full-Batch training takes approximately 15 hours and HES asymmetric sampling takes approximately 13 hours. The reduction is about 2 hours, or 13.3%. The timing includes rollout generation, HES scoring, and policy updates.

Table 22: End-to-end RL wall-clock time and average score on $8\times$A100 GPUs. Both strategies generate 32 rollouts per prompt; HES scoring reuses generation logits.

Strategy	Policy updates	Time	Avg. score
Full-Batch	All trajectories	$\sim$15 h	18.34
Pos-High, Neg-Rand	Selected trajectories	$\sim$13 h	19.26

HES preserves the rollout pool and reduces the trajectories used for policy updates. During generation, token entropy is computed from the logits already available for sampling, followed by entropy reduction and score accumulation. Selecting trajectories for the update stage reduces end-to-end training time while increasing the average score from 18.34 to 19.26.

O ADDITIONAL EXPERIMENTAL PROTOCOLS

Correctness-conditioned selection. HES measures cumulative uncertainty at high-entropy decision points. For SFT, it ranks verified demonstrations. For RFT, multiple candidate trajectories are generated and answer verification retains the correct candidates before HES ranking. For the proposed asymmetric RL strategy, reward separates positive and negative rollouts: high-HES selection is applied to positives, while negatives are sampled randomly to retain diverse failure modes. Correctness verification identifies successful trajectories, and HES ranks their cumulative uncertainty to prioritize informative examples.

Fixed operating points and sensitivity analysis. We use a high-entropy token ratio of 0.5% and data retention ratios of 20% and 80% as predefined operating points. The token ratio focuses scoring on a small tail of uncertain tokens. Retaining the highest-scoring 20% tests sample efficiency; retaining 80% tests the effect of removing the lowest-scoring 20%. The ratio sweep examines sensitivity around these operating points.

Training-free selection. The experiments cover Math, Code, and STEM reasoning across SFT, RFT, and RL. HES is computed directly from model token probabilities, making selection training-free: the selection rule uses a fixed entropy statistic. The overall workflow consists of scoring candidate trajectories, selecting examples, and training the target model on the resulting data.

Benchmark decontamination. Benchmark decontamination proceeds in two stages. First, source-level filtering, exemplified by OpenMathReasoning, retrieves training samples with high embedding similarity to benchmark items and uses a strong LLM to identify rephrased benchmark questions. Second, the final training candidate pool undergoes an additional benchmark-oriented check using the `lm-eval-harness` decontamination pipeline: normalize the input field, construct a word-level n-gram lookup with 8-gram overlap, match against the evaluation benchmark prompts, and remove matched samples.

Relationship to other selection approaches. HES ranks trajectories using cumulative uncertainty at high-entropy tokens. This design complements LLM-judge selection, gradient or influence selection, distribution-based filtering and deduplication, and attention-based selection, which emphasize different properties of training data. Within entropy-based approaches, HES operates at the trajectory level, selecting examples for post-training, while token-level approaches use entropy to allocate individual updates or distillation routes.

Qualitative interpretation and release plan. Qualitatively, high-HES traces of similar length can contain more conditional branches, strategy changes, or revisions, whereas low-HES traces can contain more repeated computation or restatement. This interpretation links HES to the distribution of uncertain decisions within a trajectory. We plan to release the curation, training, and HES-selection scripts together with the complete 100,000-example subsets after acceptance.

<think>\nOkay, let me try to figure out how to evaluate this expression: 0 - 1 - 2 + 3 - 4 + 5 + 6 + 7 - 8 + ... + 2000. The problem says that the terms with minus signs are exactly the powers of two. Hmm, so first, I need to make sure I understand the pattern here. Let me list out the terms step by step to see if I can spot the pattern.\n\nStarting from 0:\n\nTerm 0: 0 (but the first term is 0, then subtract 1, subtract 2, add 3, subtract 4, etc.)\n\nWait, let's write it out more clearly. The expression is:\n\n0 - 1 - 2 + 3 - 4 + 5 + 6 + 7 - 8 + ... + 2000\n\nSo, starting from 0, then subtract 1, subtract 2, add 3, subtract 4, add 5, add 6, add 7, subtract 8, and so on. The minus signs are on powers of two. Let's check:\n\nPowers of two are 1, 2, 4, 8, 16, 32, ..., up to 1024 since 2^10 = 1024, and next would be 2048 which is beyond 2000. So all the powers of two from 1 up to 1024 are subtracted, and all other numbers are added. Let me verify this with the given expression.\n\nStarting from term 0:\n\n0 (term 0) is separate. Then:\n\n-1 (term 1), which is 2^0 =1\n\n-2 (term 2), which is 2^1 =2\n\n+3 (term 3)\n\n-4 (term 4), which is 2^2=4\n\n+5 (term5), +6 (term6), +7 (term7)\n\n-8 (term8), which is 2^3=8\n\nThen from 9 to 15: all added?\n\nWait, next power of two is 16, so terms 9 to 15 would be added, then term16 is subtracted. So the pattern is that whenever the term number is a power of two, it's subtracted; otherwise, it's added. But wait, the problem says "the terms with minus signs are exactly the powers of two." So that's correct. So each term in the series is being added except for the terms that are powers of two, which are subtracted.\n\nBut note that the first term is 0. So the expression is:\n\n0 (term0) -1 (term1) -2 (term2) +3 (term3) -4 (term4) +5 (term5) +6 (term6) +7 (term7) -8 (term8) +... +2000.\n\nSo the series starts at 0, then for each term from 1 to 2000, if the term is a power of two, it's subtracted; otherwise, it's added.\n\nWait, but does the problem say "the terms with minus signs are exactly the powers of two"? So, all the subtracted terms are powers of two, and all powers of two are subtracted. So, starting from 1 to 2000, every number that is a power of two is subtracted, and all others are added. The initial 0 is just there. So, the entire expression can be written as:\n\n0 + (sum from n=1 to 2000 of (-1)^{n is power of two} * n)\n\nBut how do we compute that? Let's think.\n\nSo, the total sum would be 0 plus the sum from n=1 to 2000 of n, except that whenever n is a power of two, we subtract 2n instead of adding n. Wait, no. Wait, if a term is subtracted instead of added, the difference is that instead of +n, we have -n. So the total sum would be (sum from n=1 to 2000 of n) - 2*(sum of all powers of two from 1 to 2000).\n\nBecause for each power of two term, we subtract it instead of adding it, so we have to subtract twice that term. Because normally it's +n, but now it's -n, so the difference is -2n. Therefore, total sum is sum_{n=1}^{2000} n - 2*sum_{k=0}^{m} 2^k, where 2^m <=2000.\n\nWait, let's verify that. Let's say S = sum_{n=1}^{2000} n, which is (2000)(2001)/2. Then, the terms which are powers of two are subtracted instead of added. So each power of two term is contributing -n instead of +n, so the difference is -2n. Therefore, the total sum would be S - 2*sum_{powers of two <=2000} n. That seems correct.\n\nSo first, compute S = sum from 1 to 2000: S = (2000)(2001)/2.\n\nThen, compute T = sum of all powers of two up to 2000.\n\nWhat's the sum of powers of two from 1 to 2000?\n\nPowers of two are 1, 2, 4, 8, ..., 1024. Since 2^10 = 1024, 2^11 = 2048 which is over 2000. So the exponents go from 0 to 10, so the sum is 2^(10+1) -1 = 2048 -1 = 2047. Wait, but wait, the sum of 2^0 + 2^1 + ... +2^m = 2^(m+1) -1. So here, the powers of two up to 1024 (which is 2^10) would be 2^0 + 2^1 + ... +2^10 = 2^11 -1 = 2048 -1 = 2047. Therefore, T = 2047.\n\nTherefore, the total sum would be S - 2*T = (2000*2001/2) - 2*2047.\n\nCompute that:\n\nFirst, compute S = 2000*2001/2. Let's calculate that:\n\n2000*2001 = 2000*(2000 +1) = 2000^2 + 2000 = 4,000,000 + 2,000 = 4,002,000\n\nThen, divide by 2: 4,002,000 /2 = 2,001,000.\n\nThen, 2*T = 2*2047 = 4094.\n\nTherefore, total sum is 2,001,000 - 4,094 = let's compute that.\n\n2,001,000 - 4,000 = 1,997,000\n\nThen subtract 94: 1,997,000 - 94 = 1,996,906.\n\nWait, wait. Wait 2,001,000 - 4,094:\n\n2,001,000 - 4,000 = 1,997,000\n\nThen subtract

94 more: 1,997,000 -94 = 1,996,906\n\nYes. So the total sum is 1,996,906.\n\nWait, but hold on. The first term in the given expression is 0. So in the problem, the expression starts at 0. So when we calculated sum from n=1 to 2000, we included all terms starting from 1. So the initial 0 is separate. Wait, but the problem is written as:\n\n0 -1 -2 +3 -4 +5 +6 +7 -8 + ... +2000\n\nSo starting from 0, then all terms from 1 to 2000 are included with the signs. Therefore, in our calculation above, S is sum from 1 to 2000, and then we subtract 2*T. Then, since the expression is 0 + [sum from 1 to 2000 with signs], so the total is indeed 0 + (S - 2*T). Therefore, our calculation is correct.\n\nBut let me verify this with a small example. Let's take a small N, say up to 8.\n\nTest case: Compute 0 -1 -2 +3 -4 +5 +6 +7 -8.\n\nAccording to our formula:\n\nsum from 1 to 8: 8*9/2 = 36\n\nsum of powers of two up to 8: 1 +2 +4 +8=15\n\nTotal sum: 0 + (36 - 2*15) = 0 + (36 -30) =6\n\nBut let's compute manually:\n\n0 -1 -2 +3 -4 +5 +6 +7 -8\n\nCompute step by step:\n\n0 -1 = -1\n\n-1 -2 = -3\n\n-3 +3 = 0\n\n0 -4 = -4\n\n-4 +5 = 1\n\n1 +6 =7\n\n7 +7=14\n\n14 -8=6\n\nWhich matches the formula's result. Therefore, the formula seems correct.\n\nAnother test case: up to 16. Let's compute using formula and manually.\n\nFormula:\n\nsum from 1 to16:16*17/2=136\n\nsum of powers of two up to 16:1+2+4+8+16=31\n\nTotal sum:136 -2*31=136-62=74\n\nManually:\n\n0 -1 -2 +3 -4 +5 +6 +7 -8 +9 +10 +11 +12 +13 +14 +15 -16\n\nLet's compute step by step:\n\n0 -1 = -1\n\n-1 -2 = -3\n\n-3 +3 =0\n\n0 -4 = -4\n\n-4 +5=1\n\n1 +6=7\n\n7 +7=14\n\n14 -8=6\n\n6 +9=15\n\n15 +10=25\n\n25 +11=36\n\n36 +12=48\n\n48 +13=61\n\n61 +14=75\n\n75 +15=90\n\n90 -16=74\n\nWhich matches the formula. Therefore, the formula works.\n\nTherefore, for the original problem, the answer should be 1,996,906.\n\nBut let me check again the sum of powers of two up to 2000. The powers of two are 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. Let's check:\n\n2^0 =1\n\n2^1=2\n\n2^2=4\n\n2^3=8\n\n2^4=16\n\n2^5=32\n\n2^6=64\n\n2^7=128\n\n2^8=256\n\n2^9=512\n\n2^10=1024\n\n2^11=2048 >2000, so stop here.\n\nSo sum is 1+2+4+8+16+32+64+128+256+512+1024. Let's compute that:\n\n1+2=3\n\n3+4=7\n\n7+8=15\n\n15+16=31\n\n31+32=63\n\n63+64=127\n\n127+128=255\n\n255+256=511\n\n511+512=1023\n\n1023+1024=2047\n\nYes, so sum of powers of two up to 2000 is 2047. Therefore, the calculation is correct.\n\nSo total sum is 2,001,000 -4,094=1,996,906.\n\nTherefore, the answer is \boxed{1996906}.\n\n**Final Answer**\n\boxed{1996906}\n</think>\n\nTo evaluate the expression \(0 - 1 - 2 + 3 - 4 + 5 + 6 + 7 - 8 + \ldots + 2000\), where the terms with minus signs are exactly the powers of two, we can follow these steps:\n\n1. **Identify the sum of all terms from 1 to 2000**:\n \[\n S = \sum_{n=1}^{2000} n = \frac{2000 \times 2001}{2} = 2001000\n \]\n\n2. **Identify the sum of all powers of two up to 2000**:\n The powers of two up to 2000 are \(1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\). The sum of these powers of two is:\n \[\n T = 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 + 1024 = 2047\n \]\n\n3. **Adjust the total sum by subtracting twice the sum of the powers of two**:\n Since each power of two is subtracted instead of added, the total adjustment is \(2 \times T\). Therefore, the total sum is:\n \[\n \text{Total Sum} = S - 2T = 2001000 - 2 \times 2047\n \]\n\n4. **Calculate the final result**:\n \[\n 2001000 - 4094 = 1996906\n \]\n\nThus, the evaluated value of the given expression is \(\boxed{1996906}\)

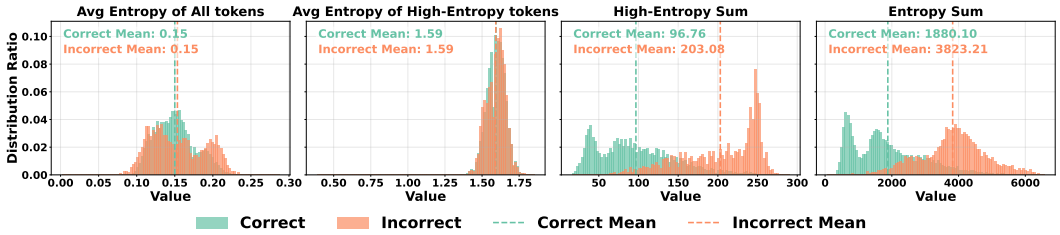


Figure 6: Comparative analysis of discriminative ability between HES and other metrics based on 512 responses per problem sampled by Qwen3-14B on AIME 2025. The x-axis is specific entropy value.

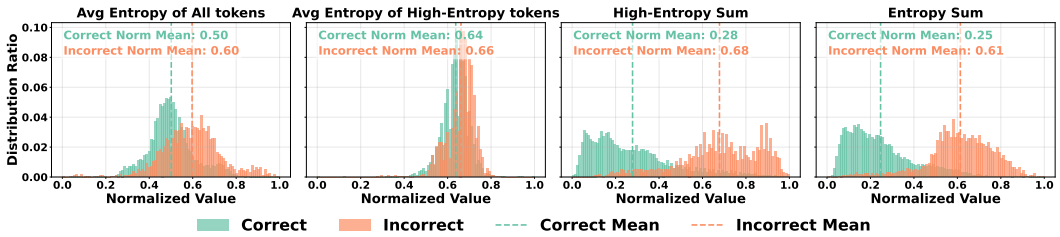


Figure 7: Comparative analysis of discriminative ability between HES and other metrics based on 512 responses per problem sampled by Qwen3-14B on AIME 2024.

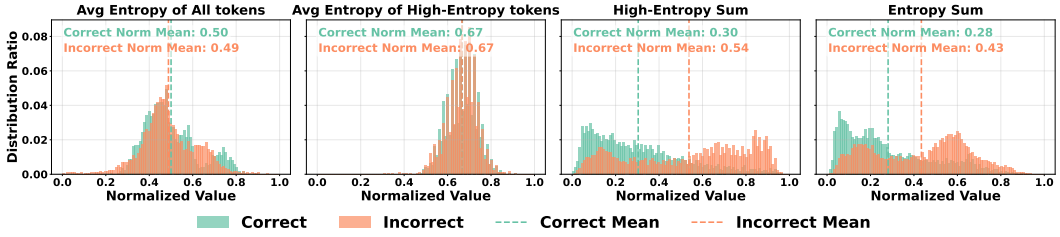


Figure 8: Comparative analysis of discriminative ability between HES and other metrics based on 512 responses per problem sampled by Qwen3-14B on HMMT 2023.

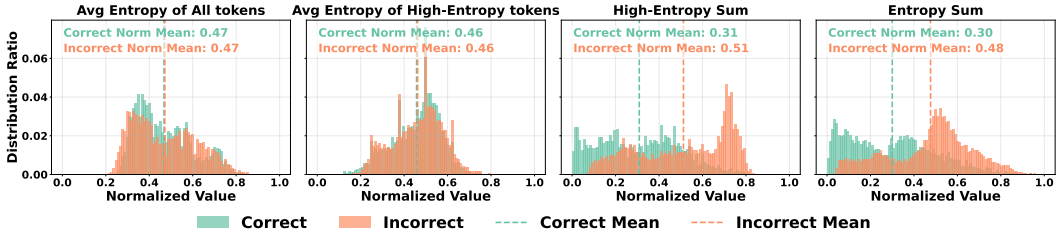


Figure 9: Comparative analysis of discriminative ability between HES and other metrics based on 512 responses per problem sampled by Qwen3-14B on HMMT 2024.

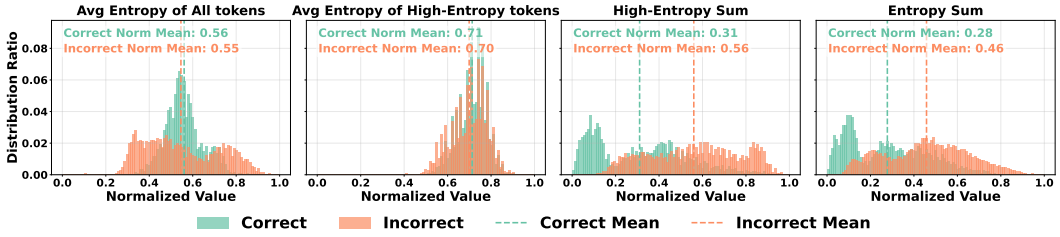


Figure 10: Comparative analysis of discriminative ability between HES and other metrics based on 512 responses per problem sampled by Qwen3-14B on HMMT 2025.