WHEN SHOULD LLMS REASON? ADAPTIVE INFERENCE ROUTING VIA ENTROPY DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Chain-of-thought (CoT) prompting can improve reasoning, but its accuracy gains and token costs vary across tasks and models. We investigate early entropy dynamics as a signal for selective reasoning. Higher CoT gains tend to coincide with decreasing entropy, whereas lower gains accompany unstable or increasing trajectories. Change-point analysis characterizes this temporal structure. We introduce **EDRM** (Entropy Dynamics-based Reasoning Manifold), a routing framework that summarizes early entropy trajectories using cumulative entropy, monotonic trend, and local volatility. Its training-free heuristic selects among Direct, Standard, and CoT decoding at either the dataset or instance level; an optional learned router provides additional flexibility. Across 15 benchmarks and 7 LLMs, selected deployable configurations reduce token consumption by approximately **16–55%** relative to always-on CoT, with model-dependent accuracy trade-offs. In the four-model ablation, **Full**, an oracle fallback using ground-truth answer selection, improves accuracy by up to **4.9 percentage points** over routing without fallback. These results establish an empirical accuracy–cost trade-off for entropy-guided reasoning.

1 INTRODUCTION

Chain-of-thought (CoT) reasoning is a widely used approach to eliciting multi-step problem solving from large language models (LLMs). However, its benefits depend on the task, model, and query: explicit reasoning can provide limited gains or even reduce accuracy on tasks requiring direct recall or fluent generation Liu et al. (2024; 2025). Applying CoT uniformly can therefore increase token consumption and latency without a corresponding accuracy benefit. This motivates a practical question: *when should a model engage in explicit reasoning, and can this decision be made at inference time?*

Answering this question requires considering both the dynamics of reasoning Kim et al. (2024); Li et al. (2024); Zhao (2026) and the interaction between model capability and query characteristics Sprague et al. (2024b); Ding et al. (2025). Existing routing and intervention methods adapt reasoning through offline profiling, learned policies, or modifications to token generation Sui et al. (2025); Su et al. (2025). We investigate a complementary signal: the evolution of token-level entropy during a short initial decode. In our experiments, tasks with higher CoT gains tend to exhibit decreasing entropy, whereas tasks with lower gains show less stable or increasing trajectories. We interpret this pattern as a change in decoding regime and examine its temporal structure using change-point analysis (§3.2).

Based on this observation, we propose **EDRM** (Entropy Dynamics-based Reasoning Manifold). EDRM maps early entropy trajectories to a compact representation of uncertainty, trend, and volatility, then uses this representation to select an inference strategy. The core heuristic requires no router training and supports both dataset-level and instance-level decisions.

Contributions. (1) We relate early entropy dynamics to CoT utility and quantify their temporal structure through change-point analysis. (2) We introduce EDRM, with training-free heuristics and an optional learned router, distinguishing deployable configurations from Full (Oracle). (3) We evaluate 15 benchmarks and 7 LLMs spanning 3B–35B parameters, including dense, MoE, and reasoning-distilled models, observing approximately 16–55% token savings with model-dependent accuracy trade-offs.

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.

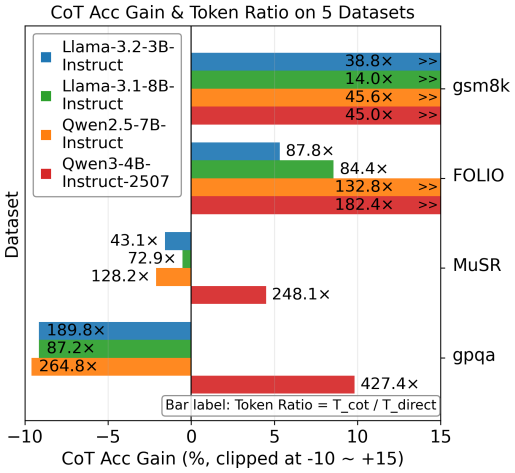

Figure 1: CoT accuracy gains over Direct across models and tasks. Extended results are in Appendix A.2.

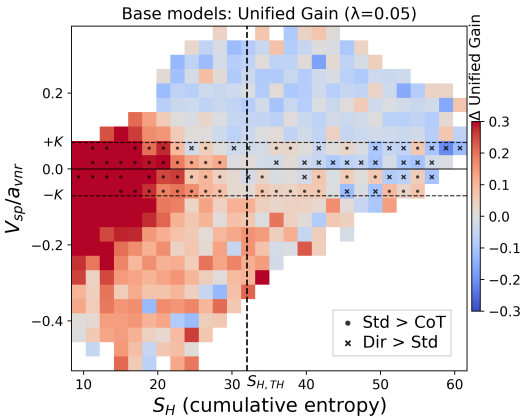

Figure 2: Unified gain of CoT over Direct across the $(V_{\text{sp}}/a_{\text{vnr}}, S_H)$ space. Each cell averages the accuracy–cost utility difference over its instances.

2 Related Work

Static *vs.* Dynamic Views of LLM Reasoning. Mechanistic studies identify circuits associated with reasoning Kim et al. (2024); Conmy et al. (2023), while analyses of CoT examine capabilities elicited by prompting Li et al. (2024); Wang & Zhou (2024). Complementary work studies uncertainty during generation: semantic entropy indicates hallucination risk Farquhar et al. (2024), and entropy trajectories have been linked to reasoning reliability Zhao (2026); Zhu et al. (2026). These observations motivate entropy-aware decoding and adaptive interventions Su et al. (2025); Jin et al. (2025); He et al. (2026). Other approaches characterize reasoning effort through deep-thinking tokens Chen et al. (2026) or distinguish reasoning from recall using activations Fartale et al. (2025). EDRM builds on this dynamic perspective by using early entropy trajectories for inference routing.

Task-Model Co-Dependency and Adaptive Efficient Reasoning. Prior studies document substantial variation in CoT utility across task types Sprague et al. (2024b); Liu et al. (2024). Adaptive approaches align computation with task difficulty and model capability through item response theory Fernandez et al. (2025), budget-aware routing Ding et al. (2025), task-aware adaptation Liu et al. (2026), hybrid reasoning Jiang et al. (2025), mixture-of-experts Fein-Ashley et al. (2025), and task decomposition Shao et al. (2025b;a); Qi et al. (2025). Their requirements vary: some train routing policies Sui et al. (2025); Huang et al. (2026), while others intervene during generation Su et al. (2025); Jin et al. (2025) or use precomputed task statistics Fernandez et al. (2025); Ding et al. (2025). Entropy-driven methods offer lightweight alternatives Sharma & Chopra (2025); Zhao (2026); Zhu et al. (2026). Token Signature Liu et al. (2025) is particularly related, using decoding features to guide CoT selection. EDRM combines three entropy descriptors with a three-way routing rule; §4 evaluates a Token-Signature-inspired comparator under the shared probing protocol.

3 Methodology

We define the decoding regimes and entropy descriptors, analyze their empirical relationship to CoT utility, and then introduce the EDRM routing rule. Additional definitions and analyses appear in Appendix A.

3.1 Preliminaries

Decoding paradigms. We compare three decoding regimes under the same task descriptions: **Direct** requests only the final answer; **Standard** uses minimal prompting and supplies the trajectory

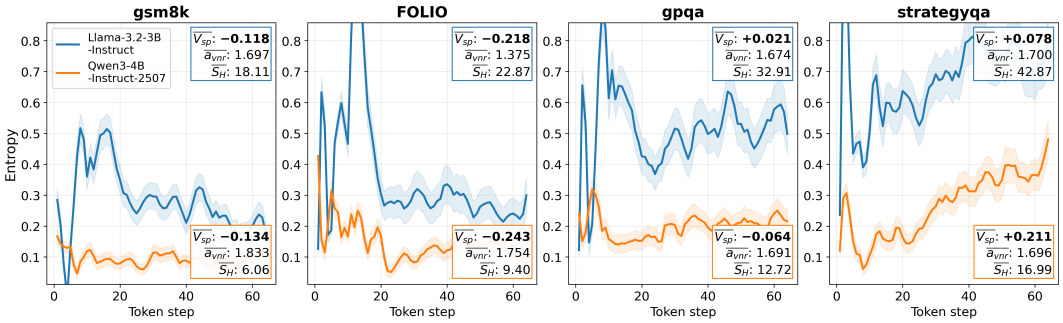

Figure 3: Mean token-level entropy during Standard probing. Tasks with higher CoT gains tend to show decreasing entropy; those with lower gains exhibit less stable or increasing trajectories.

for probing; and **CoT** requests explicit step-by-step reasoning. Model-specific controls and prompt templates are provided in Appendix B.2.

Autoregressive generation and token-level entropy. At each decoding step i, an autoregressive LLM produces a next-token distribution p_i over the vocabulary $\mathcal{V}$, whose uncertainty is quantified by the *token-level entropy* $H_i = -\sum_{v\in\mathcal{V}} p_i(v)\log p_i(v)$; the entropy trajectory $\{H_i\}_{i=1}^{N}$ captures how this uncertainty evolves during generation.

Sequential statistics for entropy trajectory characterization. We use two statistics for a scalar sequence: **Spearman correlation** summarizes its monotonic trend using ranks, reducing sensitivity to the magnitudes of outliers; the **von Neumann ratio (VNR)** divides the mean squared successive difference by the sequence variance plus a stability constant ϵ. Negative Spearman correlation indicates decreasing entropy, while a larger VNR indicates greater variation between adjacent steps relative to overall dispersion. Formal definitions appear in Appendix A.1.

3.2 Systematic Analysis

Two observations motivate EDRM (Figure 1): CoT can increase token consumption without improving accuracy, and its gains on the same benchmark can differ across models. We therefore examine whether early decoding dynamics provide a signal of reasoning utility. Over the first N=64 Standard decoding steps (Figure 3), tasks with higher CoT gains tend to exhibit decreasing entropy, whereas lower-gain tasks show oscillatory or increasing trajectories. This association motivates a *regime-shift* interpretation: decreasing entropy may indicate a transition toward more concentrated token predictions. We summarize each trajectory using three descriptors:

$$S_H = \sum_{i=1}^{N} H_i, \quad V_{\text{sp}} = \text{Spearman}(\{1,\dots,N\},\{H_1,\dots,H_N\}), \quad a_{\text{vnr}} = \text{VNR}(H_i) \tag{1}$$

For clarity, the volatility descriptor is computed as

$$a_{\text{vnr}} = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}(H_{i+1}-H_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(H_i-\bar{H})^2+\epsilon}, \qquad \bar{H} = \frac{1}{N}\sum_{i=1}^{N} H_i. \tag{2}$$

Thus, V_{sp} captures the direction of change, whereas a_{vnr} captures variation between adjacent steps relative to the trajectory's overall dispersion. The two descriptors need not vary together: an oscillating trajectory can have a weak global trend but substantial local variation.

These descriptors characterize three empirical regimes: **(1) Convergent**: negative V_{sp} and relatively small a_{vnr} indicate a stable decrease in entropy, often associated with higher CoT gains. **(2) Exploratory**: weak or positive trends and greater local volatility indicate less stable trajectories. **(3) High uncertainty**: large S_H indicates sustained uncertainty during probing. Figure 2 relates regions of this feature space to CoT utility. We use these associations to construct an empirical routing rule rather than assume that entropy alone establishes successful reasoning.

Quantifying temporal structure with change-point detection. We apply two complementary change-point detection algorithms—CUSUM and BinSeg—to the per-sample entropy sequences over the first 64 tokens, partitioning samples into three groups by their routing decisions (Direct/Standard/CoT; $n \approx 21$k samples per model; label-free, protocol in Appendix B.8). Three findings emerge: (1) *Detected splits explain more variance than random splits*: the between-group variance explained by BinSeg-detected change points is 2.55–4.38$\times$ that of random split points (all $p \ll 0.001$). (2) *Change-point location correlates with routing semantics*: CoT-routed samples exhibit the earliest change points (BinSeg median 9–29 tokens across models), while Direct-routed samples exhibit the latest (median 33–38 tokens) or are indistinguishable from Standard-routed samples—consistent with convergent trajectories committing to a reasoning strategy earlier. (3) *Pre/post change-point entropy differs significantly*: 11 of 12 model–group pairs show significant mean entropy differences before and after the change point (paired t-test, $p < 10^{-5}$; sole exception: Qwen2.5-7B-Standard, $p = 0.15$). These results characterize temporal structure in the entropy trajectories. They support the regime-shift interpretation but do not, by themselves, establish that an entropy change causes successful reasoning.

3.3 Entropy Dynamics-based Reasoning Manifold (EDRM)

We formulate adaptive reasoning as a *decoding-state routing* problem: a short initial trajectory supplies query-specific evidence for selecting among Direct, Standard, and CoT.

Framework overview. Given a model M and query Q, EDRM performs a short Standard decode and records the first N token entropies. It then forms $\mathbf{z}(Q) = (S_H, V_{\text{sp}}, a_{\text{vnr}})$ (Eq. 1), summarizing cumulative uncertainty, trend, and local volatility. We call this empirical feature space the **Entropy Dynamics-based Reasoning Manifold**; the term denotes a compact representation, not a formally established geometric manifold. If the probe ends before N tokens, the heuristic retains the completed Standard response. Otherwise, it applies the routing rule below.

Adaptive reasoning routing. Based on the manifold representation, EDRM adaptively selects among *Direct*, *Standard*, and *CoT* decoding. The routing policy is motivated by the empirical regimes above:

$$\mathcal{M}(Q) = \begin{cases} \text{Direct}, & V_{\text{sp}} > k \cdot a_{\text{vnr}} \ \vee \ (V_{\text{sp}} > 0 \wedge S_H > S_{H,\text{th}}), \\ \text{CoT}, & V_{\text{sp}} < -k \cdot a_{\text{vnr}}, \\ \text{Standard}, & \text{otherwise.} \end{cases} \tag{3}$$

A sufficiently negative trend favors CoT, while a sufficiently positive trend favors Direct. When the trend is positive, the additional S_H threshold also selects Direct for high cumulative uncertainty; intermediate cases retain Standard. The hyperparameter $S_{H,\text{th}}$ is either set empirically per model or calibrated across datasets (Appendix A.4.1). Algorithm 1 supports two granularities: dataset-level routing uses mean descriptors from a small calibration subset, whereas instance-level routing computes descriptors for each query. Both heuristic variants operate without training a router.

Cross-dataset threshold calibration. For a fixed model, the calibrated variants estimate mean cumulative entropy and mean trend from 50 probe samples per dataset. Let J_- be the number of datasets whose mean trend is negative, and sort the dataset-level mean cumulative entropies as $s_{(1)} \leq \cdots \leq s_{(J)}$. The calibrated threshold is $S_{H,\text{th}} = \lfloor s_{(\max(1,J_-))} \rfloor$. This heuristic uses entropy features without correctness labels and adapts the threshold to the model's entropy scale. Global routing then applies the decision rule to mean descriptors, while instance-level routing applies it to each query (Appendix A.4.1).

Probe reuse and inference cost. When Standard is selected, generation continues from the probe, so those tokens form part of the final Standard response. Selecting Direct or CoT instead requires a new decode under the corresponding prompt; the discarded probe contributes to the token cost. Full and Selected-FB additionally generate a Direct candidate when the routed mode is Standard or CoT. Their generation cost is therefore shared, even though their answer-selection rules differ. This separates the cost of producing candidates from the ability to select a correct answer (Appendix B.3).

Learned manifold router. For additional routing flexibility, EDRM also supports an *optional* learned router: a three-layer MLP trained on a small calibration set (10% stratified split per bench-

mark), predicting $\mathcal{M}(Q) = \arg\max_m p_\theta(m \mid \mathbf{z}(Q))$ from either the manifold representation $\mathbf{z}(Q)$ or the full entropy trajectory (training details in Appendix A.6).

Fallback compensation and deployable selector. The **Full (Oracle)** configuration adds a parallel Direct branch when the router selects Standard or CoT, and uses ground-truth correctness to select the final answer. It provides an upper bound on answer selection for the generated candidates, not a deployable policy. We separately evaluate **w/o-FB**, which returns the routed answer, and **Selected-FB**, which selects between candidates using a calibrated probability-based signal without test labels (§4; Appendix B.10).

4 Experiments

4.1 Experimental Setup

Datasets and models. We use 15 benchmarks in 4 categories: (1) mathematical reasoning: gsm8k Cobbe et al. (2021), MultiArith Roy & Roth (2016), bbh Srivastava et al. (2022); (2) commonsense and knowledge reasoning: commonsenseqa Speer et al. (2016), strategyqa Geva et al. (2021), piqa Bisk et al. (2019), siqa Sap et al. (2019), MuSR Sprague et al. (2024a); (3) scientific reasoning: arc_challenge Clark et al. (2018), arc_easy Clark et al. (2018), gpqa Rein et al. (2023); (4) formal and logical reasoning: FOLIO Han et al. (2024), ContextHub_abductive Hua et al. (2025), ContextHub_deductive Hua et al. (2025), lsat Zhong et al. (2023). We evaluate 7 LLMs: Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct Grattafiori et al. (2024); Qwen2.5-7B-Instruct and Qwen2.5-32B-Instruct Hui et al. (2024); Qwen3-4B-Instruct-2507 Yang et al. (2025); DeepSeek-R1-Distill-Qwen-7B; and Qwen3.5-35B-A3B. These cover 3B–35B parameters and include dense, reasoning-distilled, and MoE architectures. We call the Llama and Qwen2.5 instruction-tuned models the four base models for consistency with the tables.

Baselines. We compare 9 decoding strategies in two categories: **Static regimes** apply a fixed decoding mode to all samples: (1) *Direct* (no reasoning), (2) *Standard* (minimal prompting), (3) *CoT* (always-on chain-of-thought). **Adaptive routing** dynamically selects the reasoning mode: (4) *Token-Signature*, a comparator inspired by token-signature routing and adapted to our entropy probe (specification in Appendix B.13); and the **EDRM variants**, at two granularities—(5–6) *EDRM-Global-E/C* for dataset-level routing and (7–8) *EDRM-Inst-E/C* for instance-level routing, where "E" uses empirical thresholds ($S_{H,\text{th}}$=39 for Qwen2.5-32B, 32 for the other base models, 10 for Qwen3-4B, 30 for DeepSeek-R1, and 20 for Qwen3.5-35B) and "C" uses cross-dataset calibration; (9) *EDRM-MLP* uses a learned instance-level router (training and data-partitioning protocol in Appendices A.6 and B.14). Randomized components use 8 seeds. Inst-E/C and MLP in the main comparison use Full (Oracle); the appendix reports MLP results on both the full data and the held-out 90% split.

Algorithm 1 Entropy Dynamics-based Routing (EDRM-Global and EDRM-Heuristic)

Require: Input x (sample or dataset), probe length N, hyper-parameters $(k, S_{H,\text{th}})$
Ensure: Inference mode $M \in \{\text{Direct}, \text{Standard}, \text{CoT}\}$
if dataset-level routing **then**
  Probe calibration samples; compute mean descriptors $(\bar{S}_H, \bar{V}_{\text{sp}}, \bar{a}_{\text{vnr}})$
  $(S_H, V_{\text{sp}}, a_{\text{vnr}}) \leftarrow (\bar{S}_H, \bar{V}_{\text{sp}}, \bar{a}_{\text{vnr}})$
else ▷ instance-level routing
  Probe x to obtain E_x; if fewer than N tokens, return Standard
  Compute $S_H(x), V_{\text{sp}}(x), a_{\text{vnr}}(x)$
end if
if $V_{\text{sp}} > k \cdot a_{\text{vnr}}$ **or** $(V_{\text{sp}} > 0$ **and** $S_H > S_{H,\text{th}})$ **then**
  $M \leftarrow$ Direct
else if $V_{\text{sp}} < -k \cdot a_{\text{vnr}}$ **then**
  $M \leftarrow$ CoT
else
  $M \leftarrow$ Standard
end if
return M

Table 1: Overall comparison across 15 benchmarks (sample-weighted accuracy in % / average tokens). Inst-E/C and MLP use Full (Oracle); MLP uses the full-set evaluation. Bold marks highest accuracy per model, including Oracle results; underlining marks lowest token cost excluding Direct. DR1 = DeepSeek-R1-Distill.

Method	Llama3.2 3B	Llama3.1 8B	Qwen2.5 7B	Qwen2.5 32B	Qwen3 4B-T	DR1-Qwen 7B	Qwen3.5 35B
Direct	50.73 / 4.4	57.31 / 5.0	62.28 / 3.4	71.35 / 4.7	63.75 / 6.2	53.28 / 3.5	73.66 / 6.6
Standard	59.89 / 194.4	66.08 / 127.1	71.43 / 240.6	78.37 / 238.7	79.90 / 518.4	70.20 / 1389.0	83.52 / 1895.2
CoT	60.20 / 251.7	68.11 / 277.8	72.44 / 330.3	78.57 / 306.7	**80.75** / 642.5	70.83 / 1314.2	83.34 / 1703.4
Token-Sign.	60.78 / 251.0	68.26 / 264.3	72.78 / 356.7	79.54 / 266.9	80.74 / 682.3	70.76 / 1298.4	84.50 / 1512.2
Global-E	59.87 / 113.2	67.27 / 164.4	71.66 / 191.8	78.89 / 194.6	76.35 / 335.1	69.98 / 1342.9	83.13 / 1711.3
Global-C	59.99 / 145.9	67.15 / 172.1	71.76 / 195.5	78.75 / 205.3	78.42 / 404.2	70.20 / 1389.0	83.03 / 1651.0
Inst-E (Full)	63.63 / 179.4	70.04 / 153.9	76.07 / 242.7	81.87 / 248.6	78.54 / 401.1	74.78 / 1139.4	85.25 / 1276.1
Inst-C (Full)	**65.84** / 178.6	**71.67** / 151.1	**77.19** / 237.0	**82.25** / 243.0	79.88 / 426.7	**77.71** / 1182.1	**86.81** / 1380.2
MLP (Full)	64.91 / 180.8	69.14 / 119.9	75.35 / 224.3	81.31 / 215.8	79.33 / 422.0	74.26 / 1008.3	86.00 / 1423.1

Evaluation metrics and other details. We report accuracy and average token consumption (with token accounting detailed in Appendix B.3) to characterize the performance–efficiency trade-off. The probe length is $N{=}64$ and the heuristic router uses $k{=}0.07$ (base models and Qwen3-4B) or $k{=}0.15$ (DeepSeek-R1 and Qwen3.5-35B); sensitivity analyses are provided in Appendices B.9 and B.7. All generation uses greedy decoding; for calibration, we sample 50 instances per benchmark over 8 trials with different random seeds. The main-text MLP results use the full entropy trajectory (64D) input, with 3D/67D-input and label-variant results in Appendix A.6; the probe-overhead amortization is analyzed in Appendix B.12.

4.2 Main Results

Token efficiency across model families. Table 1 shows that EDRM-Global-E reduces token consumption by 37–55% relative to CoT on the four base models and by 48% on Qwen3-4B-T. On the base models, accuracy differs from CoT by at most 0.9 percentage points, at 113.2–194.6 tokens per sample versus 251.7–330.3 for CoT. For DeepSeek-R1-Distill-Qwen-7B, MLP with Full (Oracle) uses 1008.3 tokens versus CoT's 1314.2 (23.3% fewer), with a 3.4-point accuracy gain. The latter gain includes oracle answer selection; deployable performance is assessed separately below and in Appendix B.6.

Deployable variants vs. the Oracle ceiling. We distinguish three configurations of EDRM-Inst-E: **w/o-FB** returns the routed answer; **Selected-FB** applies a per-benchmark d_{mp} selector only when the routed and Direct answers disagree; and **Full (Oracle)** selects using ground-truth correctness. Selected-FB switches to Direct when $d_{\text{mp}} \geq \theta_{\text{bench}}$, with the threshold calibrated by 5-fold cross-validation on 50 calibration samples without test labels. Table 4 separates these configurations. The deployable selector recovers only part of the Oracle gap and slightly reduces accuracy on Qwen3-4B. The additional selector experiments in Appendix B.10 characterize this empirical limitation.

4.3 Comparisons across Scales and Architectures

We compare model pairs to examine how routing performance varies with scale, distillation, and architecture. These comparisons describe behavior across checkpoints; they do not isolate the causal effect of any single model attribute.

Scale generalization (Qwen2.5 7B→32B). CoT accuracy increases from 72.4% at 7B to 78.6% at 32B. In Table 1, MLP with Full (Oracle) saves 32.1% and 29.6% of CoT tokens, respectively, with accuracy gains of 2.9 and 2.7 percentage points. These gains include oracle answer selection and characterize the Full configuration at both scales. CoT also retains a 7.2-point advantage over Direct at 32B, illustrating the accuracy cost of uniformly disabling reasoning.

Table 2: Static regimes and Global routing on Llama-3.2-3B-Instruct across 15 benchmarks (accuracy in % / average tokens). D:S:C gives the 8-seed routing counts; per-dataset Global-E/C accuracy variances are below 1.9×10^{-3}.

Dataset	Direct	Standard	CoT	Global-E	Global-C	E D:S:C	C D:S:C
ARC-C	71.93/4.0	76.54/177.9	75.17/243.8	71.93/4.0	73.66/69.2	8:0:0	5:3:0
ARC-E	87.37/4.0	88.85/169.2	89.23/237.1	87.37/4.0	88.66/148.6	8:0:0	1:7:0
BBH	44.65/4.2	53.33/215.5	55.06/277.8	53.55/223.3	53.54/223.3	0:7:1	0:7:1
CSQA	65.60/5.4	67.90/88.3	65.44/186.1	65.60/5.4	65.60/5.4	8:0:0	8:0:0
CH-Abd	33.46/3.6	35.08/242.2	36.42/254.6	35.25/243.7	35.25/243.7	0:7:1	0:7:1
CH-Ded	20.67/3.9	42.83/280.1	41.54/258.1	42.02/266.4	42.03/266.4	0:3:5	0:3:5
FOLIO	43.27/3.8	44.77/298.2	48.59/334.4	48.59/334.4	48.59/334.4	0:0:8	0:0:8
GPQA	33.93/4.0	25.67/530.0	24.78/767.0	30.83/201.3	25.67/530.0	5:3:0	0:8:0
GSM8K	8.42/6.0	75.66/227.2	74.07/231.5	75.06/228.8	75.07/228.8	0:5:3	0:5:3
LSAT	41.13/4.0	41.13/303.3	39.25/420.5	41.13/4.0	41.13/191.1	8:0:0	3:5:0
M-Ari	24.50/5.6	96.83/129.6	97.67/115.7	97.46/119.2	97.46/119.2	0:2:6	0:2:6
MuSR	51.32/4.1	48.68/154.9	49.74/178.1	51.32/4.1	51.32/4.1	8:0:0	8:0:0
PIQA	75.52/4.8	71.49/130.4	72.52/191.6	75.52/4.8	75.52/4.8	8:0:0	8:0:0
SIQA	63.15/4.9	64.84/76.0	63.36/191.1	63.15/4.9	63.15/4.9	8:0:0	8:0:0
StrQA	61.05/4.3	61.14/121.9	64.06/225.9	61.05/4.3	61.05/4.3	8:0:0	8:0:0
Overall	50.73/4.4	59.89/194.4	60.20/251.7	59.87/113.2	59.99/145.9	–	–

Reasoning distillation (Qwen2.5-7B→DeepSeek-R1-Distill-Qwen-7B). Both models share the Qwen2.5-7B base, but the distilled model produces approximately 4.0× as many CoT tokens (330 vs. 1314), with 1.6 percentage points lower accuracy. This is consistent with a mismatch between reasoning length and task utility. MLP with Full (Oracle) reduces the distilled model's token cost by 23.3%, while Inst-C with Full reaches 77.7% accuracy. Because Direct accuracy also differs substantially (53.3% vs. 62.3%), these results reflect changes in both routing opportunities and candidate-answer quality.

Dense vs. MoE (Qwen2.5-32B→Qwen3.5-35B-A3B). Qwen3.5-35B-A3B has 4.8 percentage points higher CoT accuracy, but generates 5.6× as many CoT tokens (1703 vs. 307). Its Standard output is longer still, at 1895 tokens. MLP with Full (Oracle) yields 16.5% savings, or approximately 280 tokens per sample, compared with 29.6% savings on Qwen2.5-32B. Its accuracy advantage over CoT is approximately 2.7 points on both models. Differences in architecture, training, and generation behavior all contribute to this comparison.

Across these comparisons, routing provides different accuracy–cost operating points. The benefit depends on candidate quality and relative decoding costs, rather than model size or reasoning length alone.

4.4 Global-Level Routing

Table 2 reports Llama-3.2-3B-Instruct results; other models appear in Appendix B.4. **Routing stability.** The 8-seed decisions are concentrated on several datasets: Global-E selects Direct in all trials on ARC-C and CoT in all trials on FOLIO. Other datasets, including GSM8K and GPQA, show mixed decisions near the routing boundary. **Accuracy–cost trade-offs.** Global-E often selects Direct on knowledge and commonsense tasks, substantially reducing token use, sometimes at an accuracy cost. On FOLIO it selects CoT, which is more accurate than Direct. This variation motivates instance-level routing.

4.5 Instance-Level Routing

To mitigate the sampling sensitivity on multi-step benchmarks (GSM8K, BBH), we evaluate instance-level routing, which adapts to per-sample entropy trajectories; Table 3 presents per-dataset results on Llama-3.2-3B-Instruct (other models in Appendix B.5).

Table 3: Instance-level routing on Llama-3.2-3B-Instruct across 15 benchmarks (accuracy in % / average tokens). Inst-E/C and MLP use Full (Oracle). TS is the adapted Token-Signature comparator; static and Global results are in Table 2.

Dataset	TS	Inst-E (Full)	Inst-C (Full)	MLP (Full)
ARC-C	75.17/305.0	77.56/129.8	77.56/135.8	77.65/126.4
ARC-E	89.35/289.5	89.73/132.1	89.89/138.2	89.90/117.5
BBH	55.06/340.4	58.87/242.2	59.01/242.7	60.08/228.3
CSQA	65.60/58.4	68.22/82.4	68.50/83.4	70.60/90.5
CH-Abd	36.42/318.5	46.88/220.5	47.51/226.0	50.79/242.1
CH-Ded	41.54/322.0	41.63/258.6	42.31/264.0	41.92/253.2
FOLIO	47.51/349.4	57.56/324.1	57.70/326.9	54.15/243.7
GPQA	33.93/58.5	38.84/358.8	39.23/375.0	41.83/453.6
GSM8K	74.07/295.2	72.40/251.9	72.55/252.8	72.63/248.6
LSAT	42.02/279.2	46.98/287.4	47.08/290.3	48.46/314.4
M-Ari	97.67/179.6	94.50/159.4	94.50/159.4	96.17/145.2
MuSR	52.65/137.8	54.63/92.3	54.73/94.7	58.33/123.1
PIQA	75.52/172.7	77.04/97.4	77.07/98.4	77.69/99.4
SIQA	63.15/65.2	63.77/66.8	63.97/67.1	67.55/98.0
StrQA	64.06/285.4	66.20/118.4	66.51/120.8	67.12/119.4
Overall	60.78/251.0	63.63/179.4	**65.84**/178.6	64.91/180.8

Full results relative to global routing. In Table 1, Inst-E with Full (Oracle) exceeds Global-E on all four base models: 59.87%→63.63% on Llama-3.2-3B, 67.27%→70.04% on Llama-3.1-8B, 71.66%→76.07% on Qwen2.5-7B, and 78.89%→81.87% on Qwen2.5-32B. For Llama-3.2-3B, it uses 179.4 tokens versus CoT's 251.7. These differences combine instance-level routing and oracle fallback; Table 4 isolates the fallback contribution.

An efficiency spectrum across the three variants. Table 3 shows that under Full (Oracle), Inst-E/C/MLP reach 63.63/65.84/64.91% accuracy on Llama-3.2-3B at similar token costs. Inst-C additionally calibrates its threshold across datasets. This calibration concerns the routing rule; the Full answer-selection step still requires correctness labels.

Variation across tasks. Tables 2–3 show task-dependent trade-offs. On BBH, Inst-E with Full reaches 58.87% accuracy versus Global-E's 53.55%. On StrategyQA, MLP with Full uses 119.4 tokens versus CoT's 225.9. These results illustrate how instance-level selection changes computation across tasks; the accuracy gains should be interpreted together with the fallback ablation.

4.6 Statistical Analysis

We report two complementary analyses for the four models in Table 4. Instance-level bootstrap (1000 resamples) finds that Full exceeds w/o-FB by 1.7–4.9 percentage points ($p<0.01$), while w/o-FB is 0.8–3.9 points below CoT. Benchmark-level paired permutation tests ($n=15$, $B=10000$) show significant token reductions of 27–46% for w/o-FB (all $p<0.001$). Accuracy differences from CoT are not significant for Llama-3.2 and Qwen2.5-7B under the benchmark-level test, although this does not establish equivalence. Full exceeds CoT by 3.86 points on Llama-3.2 ($p=0.005$) and 4.04 points on Qwen2.5-7B ($p=0.014$). Benchmark-level differences average datasets equally and therefore differ from the sample-weighted main results. Confidence intervals and additional results appear in Appendix B.11.

4.7 Ablation and Analysis

Table 4 evaluates EDRM-Inst-E through three ablations: (1) **w/o-FB** removes oracle fallback; (2) **w/o** S_H removes the cumulative-entropy condition; and (3) **w/o** a_{vnr} further removes volatility, leaving trend-based routing. The table also reports Selected-FB to distinguish practical answer selection from the Full reference.

Table 4: EDRM-Inst-E on four models. Top: benchmark-level paired permutation tests (n=15, B=10000; accuracy differences in percentage points). Bottom: ablation accuracy in % / average tokens. Token savings compare w/o-FB with CoT. Full = Oracle; Selected-FB shares its token budget. Confidence intervals appear in Appendix B.11.

	w/o-FB vs. CoT		**Full vs. CoT**		**Token savings**	
Model	ΔAcc	p	ΔAcc	p	%	p
Llama-3.2-3B	-0.88	0.13 (ns)	$+3.86$	0.005 (**)	29.6%	$< 0.001^{***}$
Llama-3.1-8B	-2.42	0.014 (*)	$+2.63$	0.11 (ns)	46.1%	$< 0.001^{***}$
Qwen2.5-7B	-0.38	0.54 (ns)	$+4.04$	0.014 (*)	27.1%	$< 0.001^{***}$
Qwen3-4B-T	-4.01	< 0.001 (***)	-2.12	0.025 (*)	38.1%	$< 0.001^{***}$

Model	**Full**	**Selected-FB**	**w/o-FB**	**w/o S_H**	**w/o a_{vnr}**
Llama-3.2-3B	63.63 / 179.4	59.59 / 179.4	58.87 / 177.1	59.18 / 188.1	58.08 / 190.5
Llama-3.1-8B	70.04 / 153.9	65.44 / 153.9	65.11 / 149.8	65.36 / 155.4	65.08 / 248.2
Qwen2.5-7B	76.07 / 242.7	71.77 / 242.7	71.63 / 240.7	71.47 / 250.1	70.89 / 249.5
Qwen3-4B-T	78.54 / 401.1	76.71 / 401.1	76.81 / 397.9	77.71 / 441.0	76.09 / 401.1

Headroom from oracle fallback. Full exceeds w/o-FB accuracy by 1.7–4.9 percentage points across the four models, at a small additional token cost. This gap measures how often the Direct branch can recover an incorrect routed answer when correctness is known. Selected-FB recovers only part of this headroom, highlighting answer selection as a remaining deployment challenge.

Complementary roles of the entropy features. Removing the cumulative-entropy condition increases token use on all four models. Relative to w/o-FB, its accuracy effect is mixed: for example, Qwen3-4B-T rises from 76.81% to 77.71%, but uses 441.0 rather than 397.9 tokens. Further removing volatility lowers accuracy on all four models, including a 1.10-point drop on Llama-3.2 and a 1.62-point drop on Qwen3-4B. These results support complementary roles for trend, uncertainty, and volatility without implying that every feature improves accuracy in isolation. The MLP input and label ablations in Appendix B.6 examine learned alternatives; Appendices B.7 and B.9 report sensitivity to k and probe length.

5 Conclusion

EDRM uses early entropy dynamics to guide selective reasoning. Its compact representation summarizes uncertainty, trend, and volatility, and supports training-free heuristic routing as well as an optional learned router. Across 15 benchmarks and 7 LLMs, selected deployable configurations reduce token consumption by approximately 16–55% relative to CoT, with accuracy trade-offs that depend on the model and routing variant. The Full (Oracle) configuration quantifies additional headroom from answer selection. Together, these results support entropy dynamics as a useful routing signal while distinguishing empirical decoding patterns from a complete account of reasoning.

Limitations and Future Work. EDRM still requires a short probing stage, introducing overhead compared to pure Direct decoding. Our current experiments span open-source text-based models in the 3B–35B range, including dense, MoE, and reasoning-distilled architectures; extending the framework to larger-scale, API-only (see Appendix B.12 for the API deployment path), and multimodal systems remains important future work. Entropy is also sensitive to prompt format and model calibration, so transferring a threshold does not guarantee unchanged routing behavior. Our evaluation focuses on generated-token cost, which does not capture all prefill, scheduling, and hardware effects on wall-clock latency. Computing exact entropy additionally requires access to token distributions, which may restrict deployment through closed APIs. Integrating entropy-dynamic routing into multi-step agentic frameworks—where each step's reasoning decision adapts to the evolving context—is another promising direction for long-horizon multi-turn reasoning.

AI USE STATEMENT

In this work, we used generative AI tools (Claude, Anthropic; Codex, OpenAI) to assist with language polishing and consistency checks of the manuscript, and with LaTeX typesetting (e.g., reformatting tables and fixing compilation issues). We did not use generative AI tools for generating research ideas, conducting experiments, producing experimental results, or writing the core scientific claims: all experiments were designed and executed by the authors, all numbers reported in this paper were produced by the authors' own code, and all technical content was written and verified by the authors. We reviewed and edited all AI-assisted text, and we take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies when and how large language models should invoke explicit reasoning during inference, with the goal of reducing unnecessary computation. All experiments are conducted on publicly available benchmarks and open-weight models, and involve no human subjects or personal data. We do not foresee significant negative societal consequences of this work: the proposed routing mechanism operates on the model's own decoding statistics, requires no additional training, and reduces rather than increases computational cost. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

All results in this paper can be reproduced from the information provided. The full experimental setup is specified in §4: 15 public benchmarks, 7 open-weight models, greedy decoding, and 8-seed averaging for all randomized components. The EDRM method is fully specified in §3 (routing rule in Eq. 3, Algorithm 1), with the three entropy descriptors formally defined in Appendix A.1, the threshold calibration procedure (including pseudocode) in Appendix A.4.1, the learned MLP router variants in Appendix A.6, and the data partitioning protocol in Appendix B.14. All hyperparameters (k, $S_{H,\text{th}}$, N=64) and their sensitivity analyses are reported in Appendix B.7 and Appendix B.9. Prompting templates and per-model think-mode control are documented in Appendix B.2, and the evaluation metrics in Appendix B.3. Source code will be released upon publication.

REFERENCES

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:208290939.

Wei-Lin Chen, Liqian Peng, Tianhui Tan, Chao Zhao, Blake JianHang Chen, Zi Lin, Alex Go, and Yu Meng. Think deep, not just long: measuring llm reasoning effort via deep-thinking tokens. *arXiv preprint arXiv:2602.13517*, 2026. URL https://api.semanticscholar.org/CorpusID:285616013.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021. URL https://api.semanticscholar.org/CorpusID:239998651.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. Best-route: Adaptive llm routing with test-time optimal compute. *arXiv preprint arXiv:2506.22716*, 2025. URL https://api.semanticscholar.org/CorpusID:280011744.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625 – 630, 2024. URL https://api.semanticscholar.org/CorpusID:270615909.

Harshwardhan Fartale, Ashish Kattamuri, Rahul Raja, Arpita Vats, Ishita Prasad, and Akshata Kishore Moharir. Disentangling recall and reasoning in transformer models through layer-wise attention and activation analysis. *arXiv preprint arXiv:2510.03366*, 2025. URL https://api.semanticscholar.org/CorpusID:281843350.

Jacob Fein-Ashley, Dhruv Parikh, Rajgopal Kannan, and Viktor K. Prasanna. Mixture of thoughts: Learning to aggregate what experts think, not just what they say. *arXiv preprint arXiv:2509.21164*, 2025. URL https://api.semanticscholar.org/CorpusID:281525929.

Nigel Fernandez, Branislav Kveton, Ryan A. Rossi, Andrew S. Lan, and Jack Zichao Wang. Radar: Reasoning-ability and difficulty-aware routing for reasoning llms. *arXiv preprint arXiv:2509.25426*, 2025. URL https://api.semanticscholar.org/CorpusID:281681703.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. URL https://api.semanticscholar.org/CorpusID:230799347.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, et al. Folio: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, 2024.

Jiashu He, Meizhu Liu, Olaitan P Olaleye, Amit Agarwal, M Avendi, Yassi Abbasi, Matthew Rowe, Hitesh Laxmichand Patel, Paul Li, Tao Sheng, et al. Think twice before you write–an entropy-based decoding strategy to enhance llm reasoning. *arXiv preprint arXiv:2604.00018*, 2026.

Wenyue Hua, Kaijie Zhu, Lingyao Li, Lizhou Fan, Mingyu Jin, Shuhang Lin, Haochen Xue, Zelong Li, JinDong Wang, and Yongfeng Zhang. Disentangling logic: The role of context in large language model reasoning capabilities. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19219–19242, 2025.

Jiatan Huang, Zheyuan Zhang, Kaiwen Shi, Yanfang Ye, and Chuxu Zhang. Evolverouter: Co-evolving routing and prompt for multi-agent question answering. *arXiv preprint arXiv:2604.05149*, 2026. URL https://api.semanticscholar.org/CorpusID:287208061.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631*, 2025. URL https://api.semanticscholar.org/CorpusID:278768506.

Hyunbin Jin, Je Won Yeom, Seunghyun Bae, and Taesup Kim. "well, keep thinking": Enhancing llm reasoning with adaptive injection decoding. In *Association for Computational Linguistics*, pp. 9989–10018, 2025.

Geonhee Kim, Marco Valentino, and André Freitas. Reasoning circuits in language models: A mechanistic interpretation of syllogistic inference. In *Association for Computational Linguistics*, pp. 10074–10095, 2024. URL https://api.semanticscholar.org/CorpusID:271892176.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *International Conference on Learning Representations*, 2024.

Hui Liu, Bing-Ruei Zou, Ke Chen, Jie Liu, Wenya Wang, and Haoliang Li. Task-aware llm routing with multi-level task-profile-guided data synthesis for cold-start scenarios. 2026. URL https://api.semanticscholar.org/CorpusID:287352177.

Peijie Liu, Fengli Xu, and Yong Li. Token signature: Predicting chain-of-thought gains with token decoding feature in large language models. *ArXiv*, abs/2506.06008, 2025. URL https://api.semanticscholar.org/CorpusID:279243226.

Ryan Liu, Jiayi Geng, Addison J. Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L. Griffiths. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *arXiv preprint arXiv:2410.21333*, 2024. URL https://api.semanticscholar.org/CorpusID:273662487.

Shihao Qi, Jie Ma, Ziang Yin, Lingling Zhang, Jian Zhang, Jun Liu, Feng Tian, and Tongliang Liu. Plan before solving: Problem-aware strategy routing for mathematical reasoning with llms. *arXiv preprint arXiv:2509.24377*, 2025. URL https://api.semanticscholar.org/CorpusID:281675988.

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. *arXiv preprint arXiv:2311.12022*, 2023.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. *ArXiv*, abs/1608.01413, 2016. URL https://api.semanticscholar.org/CorpusID:560565.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Chenyang Shao, Xinyuan Hu, Yutang Lin, and Fengli Xu. Division-of-thoughts: Harnessing hybrid language model synergy for efficient on-device agents. *Proceedings of the ACM on Web Conference 2025*, pp. 1822–1833, 2025a. URL https://api.semanticscholar.org/CorpusID:276235473.

Chenyang Shao, Xinyang Liu, Yutang Lin, Fengli Xu, and Yong Li. Route-and-reason: Scaling large language model reasoning with reinforced model router. *arXiv preprint arXiv:2506.05901*, 2025b. URL https://api.semanticscholar.org/CorpusID:279245092.

Aman Sharma and Paras Chopra. Think just enough: Sequence-level entropy as a confidence signal for llm reasoning. *arXiv preprint arXiv:2510.08146*, 2025. URL https://api.semanticscholar.org/CorpusID:281950492.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*, 2016. URL https://api.semanticscholar.org/CorpusID:15206880.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations*, volume 2024, pp. 14670–14728, 2024a.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*, 2024b. URL https://api.semanticscholar.org/CorpusID:272708032.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmuller, Andrew M. Dai, Andrew La, Andrew Kyle Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta,

Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakacs, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartlomiej Bojanowski, Batuhan Ozyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Stephen Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cèsar Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Mosegu'i Gonz'alez, Danielle R. Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan J. Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Mart'inez-Plumed, Francesca Happ'e, François Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-L'opez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schutze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, John Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Koco'n, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Narain Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Oluwadara Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Jane W Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jorg Frohberg, Jos Rozen, José Hernández-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis philippe Morency, Luca Moschella, Luca Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Col'on, Luke Metz, Lutfi Kerem cSenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ram'irez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michal Swkedrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Monica Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, T. MukundVarma, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, P. Milkowski, Piyush S. Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphael Milliere, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi S. Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy,

Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Bradley Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie L. Lin, Stephen Prasad, Steven T Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, Thèo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yu Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv*, abs/2206.04615, 2022. URL https://api.semanticscholar.org/CorpusID:263625818.

Tiancheng Su, Meicong Zhang, and Guoxiu He. Entropy-aware speculative decoding toward improved llm reasoning. *arXiv preprint arXiv:2512.23765*, 2025. URL https://api.semanticscholar.org/CorpusID:284350835.

Yuan Sui, Yufei He, Tri Cao, Simeng Han, and Bryan Hooi. Meta-reasoner: Dynamic guidance for optimized inference-time reasoning in large language models. *arXiv preprint arXiv:2502.19918*, 2025. URL https://api.semanticscholar.org/CorpusID:276647786.

Xuezhi Wang and Denny Zhou. Chain-of-thought reasoning without prompting. *Advances in Neural Information Processing Systems*, 37:66383–66409, 2024.

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.

Xinghao Zhao. Entropy trajectory shape predicts llm reasoning reliability: A diagnostic study of uncertainty dynamics in chain-of-thought. 2026. URL https://api.semanticscholar.org/CorpusID:286673669.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *ArXiv*, abs/2304.06364, 2023. URL https://api.semanticscholar.org/CorpusID:258108259.

Cheng Zhu, Siyang Wu, Xiangkang Zeng, Zishan Xu, Zhaolu Kang, Yifu Guo, Yuquan Lu, Jun Huang, and Guojing Zhou. Edis: Diagnosing llm reasoning via entropy dynamics. *arXiv preprint arXiv:2602.01288*, 2026. URL https://api.semanticscholar.org/CorpusID:285269294.

A Appendix: Methodological Details

A.1 Entropy Dynamics Descriptors

This section provides detailed definitions and computational procedures for the three entropy dynamics descriptors used in EDRM: cumulative entropy S_H, univariate trend V_{sp}, and volatility a_{vnr}. We also discuss the rationale for selecting these specific descriptors over other potential metrics, emphasizing their theoretical grounding and empirical effectiveness in capturing the nuanced dynamics of LLM reasoning processes. Inspired by both kinematic physics of *S–V–a* (position–velocity–acceleration) and time series analysis, we design the three descriptors to capture distinct yet complementary aspects of the entropy trajectory.

Input to the descriptor extractor. For each instance x, we run an N-step *Standard* probing decode and record the token-level entropy sequence $E_x = \{H_1(x), \ldots, H_T(x)\}$, where $T \leq N$ is the number of generated probing tokens before EOS. In our instance-level heuristic routing, if $T < N$ (early termination), we treat the instance as "already confident" and route it to *Standard* directly (i.e.,

descriptors are not required for that case). Otherwise, we compute the three descriptors on E_x with $T = N$. The full procedure is summarized in Algorithm 2.

Algorithm 2 Extracting entropy dynamics descriptors $(S_H, V_{\text{sp}}, a_{\text{vnr}})$

Require: Probing entropy sequence $E = \{H_1, \ldots, H_T\}$, max length N, small constant ϵ
Ensure: Descriptors $(S_H, V_{\text{sp}}, a_{\text{vnr}})$
if $T < N$ **then**
  return EARLYSTOP ▷ route to Standard; skip descriptor-based routing
end if
$S_H \leftarrow \sum_{i=1}^{N} H_i$
$V_{\text{sp}} \leftarrow \text{Spearman}(\{1, \ldots, N\}, \{H_1, \ldots, H_N\})$
$\bar{H} \leftarrow \frac{1}{N} \sum_{i=1}^{N} H_i$
$\text{Var} \leftarrow \frac{1}{N} \sum_{i=1}^{N} (H_i - \bar{H})^2$
$\text{MSD} \leftarrow \frac{1}{N-1} \sum_{i=1}^{N-1} (H_{i+1} - H_i)^2$
$a_{\text{vnr}} \leftarrow \frac{\text{MSD}}{\text{Var}+\epsilon}$
return $(S_H, V_{\text{sp}}, a_{\text{vnr}})$

A.1.1 CUMULATIVE ENTROPY S_H

Like Position in kinematic physics, S_H is defined as the cumulative sum of token-level entropy over the N-step probing sequence, which captures the overall uncertainty accumulation during the early decoding phase. Formally, for a given input instance x, let $p_i(\cdot)$ be the next-token distribution at probing step i and its corresponding token-level entropy sequence $H_i(x)$, we compute

$$S_H(x) = \sum_{i=1}^{N} H_i(x), \qquad H_i(x) = -\sum_{v \in \mathcal{V}} p_i(v) \log p_i(v). \tag{4}$$

Here $\mathcal{V}$ is the vocabulary and $H_i(x)$ is the Shannon entropy of the next-token distribution at step i.

Interpretation. S_H measures the *total uncertainty budget* the model spends during early decoding. A high S_H indicates sustained uncertainty across multiple steps (uncertainty overload), where extended reasoning (e.g., CoT) is less likely to converge reliably and may amplify drift; a low-to-moderate S_H suggests the model is already operating in a relatively confident regime where lightweight reasoning can be beneficial.

Why S_H instead of other cumulative summaries? Alternative cumulative metrics include (i) average entropy $\bar{H} = \frac{1}{N} \sum_i H_i$, (ii) maximum entropy $\max_i H_i$, and (iii) endpoint difference $\Delta H = H_1 - H_N$. We prefer S_H because:

- $\bar{H}$ is scale-dependent and can be misleading when N varies or when the trajectory has high volatility;
- $\max_i H_i$ over-emphasizes a single step and discards global information about the trajectory;
- ΔH ignores mid-trajectory oscillations and is sensitive to noise at the first/last steps.

In contrast, S_H aggregates information across all steps and provides a stable, scale-consistent indicator of cumulative uncertainty.

A.1.2 UNIVARIATE TREND V_{SP}

Definition. The trend descriptor V_{sp} quantifies whether the entropy trajectory is monotonically decreasing (convergent) or increasing (divergent) over time. We define it as the Spearman rank correlation between the step index and entropy values:

$$V_{\text{sp}}(x) = \text{Spearman}(\{1, \ldots, N\}, \{H_1(x), \ldots, H_N(x)\}). \tag{5}$$

Operationally, let r_i be the rank of H_i among $\{H_1, \ldots, H_N\}$ (ties are assigned average ranks). Then

$$V_{\text{sp}} = \text{Corr}\left(\{1, \ldots, N\}, \{r_1, \ldots, r_N\}\right), \tag{6}$$

where $\text{Corr}(\cdot, \cdot)$ is the Pearson correlation computed on the rank-transformed sequence. By construction, $V_{\text{sp}} \in [-1, 1]$: negative values indicate a decreasing entropy trend (progressive uncertainty reduction), while positive values indicate increasing uncertainty and higher drift risk.

Interpretation. V_{sp} serves as the "velocity" signal in our kinematic analogy: it captures *directional evolution* rather than magnitude. Intuitively, tasks that benefit from deliberate reasoning typically show a sustained reduction in uncertainty during early decoding, corresponding to $V_{\text{sp}} < 0$; tasks with weak/negative CoT gains often exhibit oscillation or upward drift, reflected by $V_{\text{sp}} \geq 0$.

Why Spearman instead of common trend metrics? We choose Spearman correlation over several alternatives:

- **Linear regression slope** $\hat{\beta}$ of H_i on i: while it provides a signed trend, it is *scale-dependent* and sensitive to outliers (single entropy spikes can dominate $\hat{\beta}$). Moreover, entropy trajectories are frequently non-linear (piecewise or curved), making a slope a fragile summary.
- **Pearson correlation** $\mathrm{Corr}(i, H_i)$: it also assumes linear association and is sensitive to magnitude outliers; it can overreact to a few extreme points.
- **Average increment** $\frac{1}{N-1}\sum_{i=2}^{N}(H_i - H_{i-1})$: this relies only on adjacent differences and is highly sensitive to local noise, which is common in token-level entropy.

In contrast, V_{sp} is *non-parametric* (depends only on ordering), robust to non-linear monotone patterns, and less sensitive to occasional spikes/dips. This makes it well-suited for noisy entropy trajectories produced by early decoding.

A.1.3 Volatility a_{VNR} (Von Neumann Ratio)

Definition. We instantiate the volatility descriptor using the *Von Neumann ratio* (VNR), defined as the ratio between the mean square successive difference (MSD) and the (population) variance of the sequence:

$$a_{\text{vnr}}(x) = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}\big(H_{i+1}(x) - H_i(x)\big)^2}{\frac{1}{N}\sum_{i=1}^{N}\big(H_i(x) - \bar{H}(x)\big)^2 + \epsilon}, \tag{7}$$

where $\bar{H}(x) = \frac{1}{N}\sum_{i=1}^{N} H_i(x)$ and ϵ is a small constant (e.g., 10^{-8}). If the variance is zero, we set $a_{\text{vnr}}(x) = 0$.

Interpretation. a_{vnr} compares an ℓ_2-type dispersion (standard deviation) against an ℓ_1-type dispersion (mean absolute deviation, MAD). It acts as a normalized "acceleration" signal: when the entropy curve oscillates sharply (bursty uncertainty), the squared deviations grow faster than absolute deviations, increasing a_{vnr}; when the curve is smooth and stable, a_{vnr} is lower. In routing, a_{vnr} plays a critical role as a stability prior: trend estimates (e.g., V_{sp}) are less reliable under high volatility, so we couple them via the ratio $V_{\text{sp}}/a_{\text{vnr}}$ in the heuristic rule.

Why this volatility form instead of standard volatility metrics? We prefer a_{vnr} over several common choices:

- **Standard deviation** σ: captures global dispersion but is scale-dependent and can be inflated by a few extreme points; used alone, it cannot distinguish "smoothly high" entropy from "highly oscillatory" entropy.
- **Mean absolute increment** $\frac{1}{N-1}\sum_{i=2}^{N}|H_i - H_{i-1}|$: focuses on local changes, but it is also sensitive to deterministic trends (a steady monotone decrease can still yield large increments), thus conflating trend with volatility.
- **MAD alone** $\frac{1}{N}\sum_i |H_i - \bar{H}|$: more robust than σ but still scale-dependent and does not explicitly emphasize bursty oscillations.

The ratio σ/MAD is (i) *scale-normalized* (invariant to multiplying the entire entropy sequence by a constant), (ii) sensitive to spiky/oscillatory deviations, and (iii) empirically effective as a noise-aware normalization term when combined with V_{sp}. This design matches our practical objective: separating genuine convergence (stable downward trend) from spurious monotonicity induced by noise.

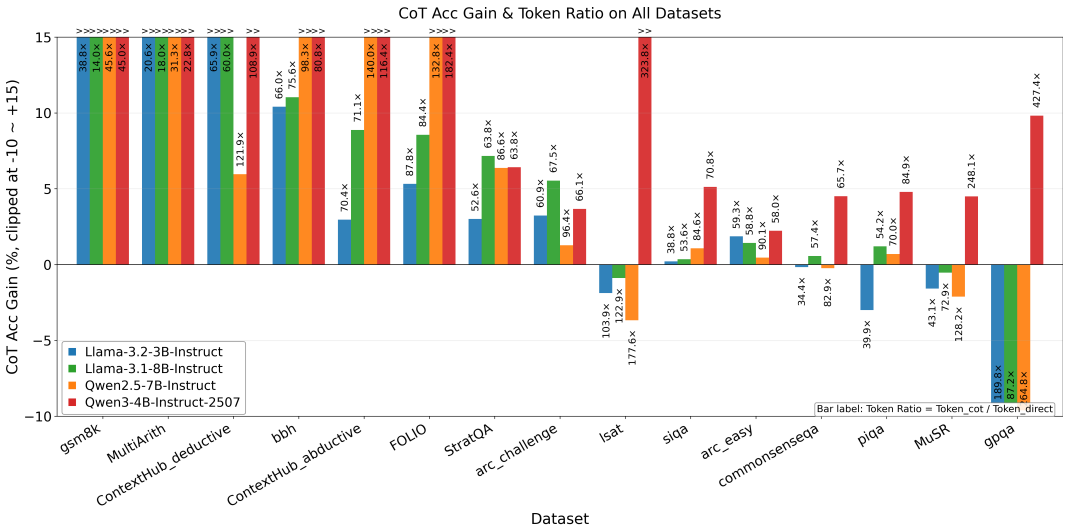

Figure 4: Comprehensive comparison of *CoT* and *Direct* decoding strategies across all evaluated benchmarks and models. Each subplot corresponds to a specific model, with bars representing the accuracy of *CoT* and *Direct* on each benchmark. The performance differences are analyzed to identify patterns of CoT gains or losses across different tasks and model sizes.

A.2 Comparison of CoT and Direct Decoding across All Benchmarks

This section provides a comprehensive comparison of *CoT* and *Direct* decoding strategies across all evaluated benchmarks and models. As shown in Figure 4, we analyze the performance gain brought by Chain-of-Thought reasoning. The results highlight distinct trends based on task type and model capability.

Mathematical Reasoning. For math-intensive benchmarks such as `gsm8k` and `MArith`, *CoT* decoding yields massive performance improvements across all models. The gains are so significant that they reach the upper limit of our visualization scale, indicating that step-by-step reasoning is essential for solving these complex arithmetic problems, far outperforming direct decoding.

Logical Deduction and Complex Tasks. In logical reasoning tasks like `CH-ded`, `bbh`, and `CH-abd`, *CoT* generally provides a positive boost, though the magnitude varies by model. Notably, the **Qwen3-4B** model (red bars) demonstrates exceptional robustness, achieving the highest CoT gains in several of these categories, often surpassing larger models like Llama-3.1-8B and Qwen2.5-7B. This suggests that Qwen3-4B is particularly effective at leveraging reasoning paths for deduction.

General Knowledge and Common Sense. For benchmarks involving common sense or general knowledge (e.g., `siqa`, `arc-e`, `csqa`), the advantages of *CoT* are much more marginal. Most models show only slight improvements or negligible differences compared to direct decoding. In some cases, such as `lsat`, `gpqa`, and `stqa`, several models (particularly the Llama series and Qwen2.5) exhibit negative gains, suggesting that for these specific tasks, generating a reasoning trace may introduce noise or errors, making direct decoding the superior strategy.

Model Comparison. A key observation is the consistent performance of **Qwen3-4B**. While other models fluctuate between positive and negative gains depending on the benchmark, Qwen3-4B maintains positive CoT gains across almost all tasks, including those where other models struggle (e.g., `gpqa`). This indicates a strong generalization capability in utilizing Chain-of-Thought reasoning across diverse domains.

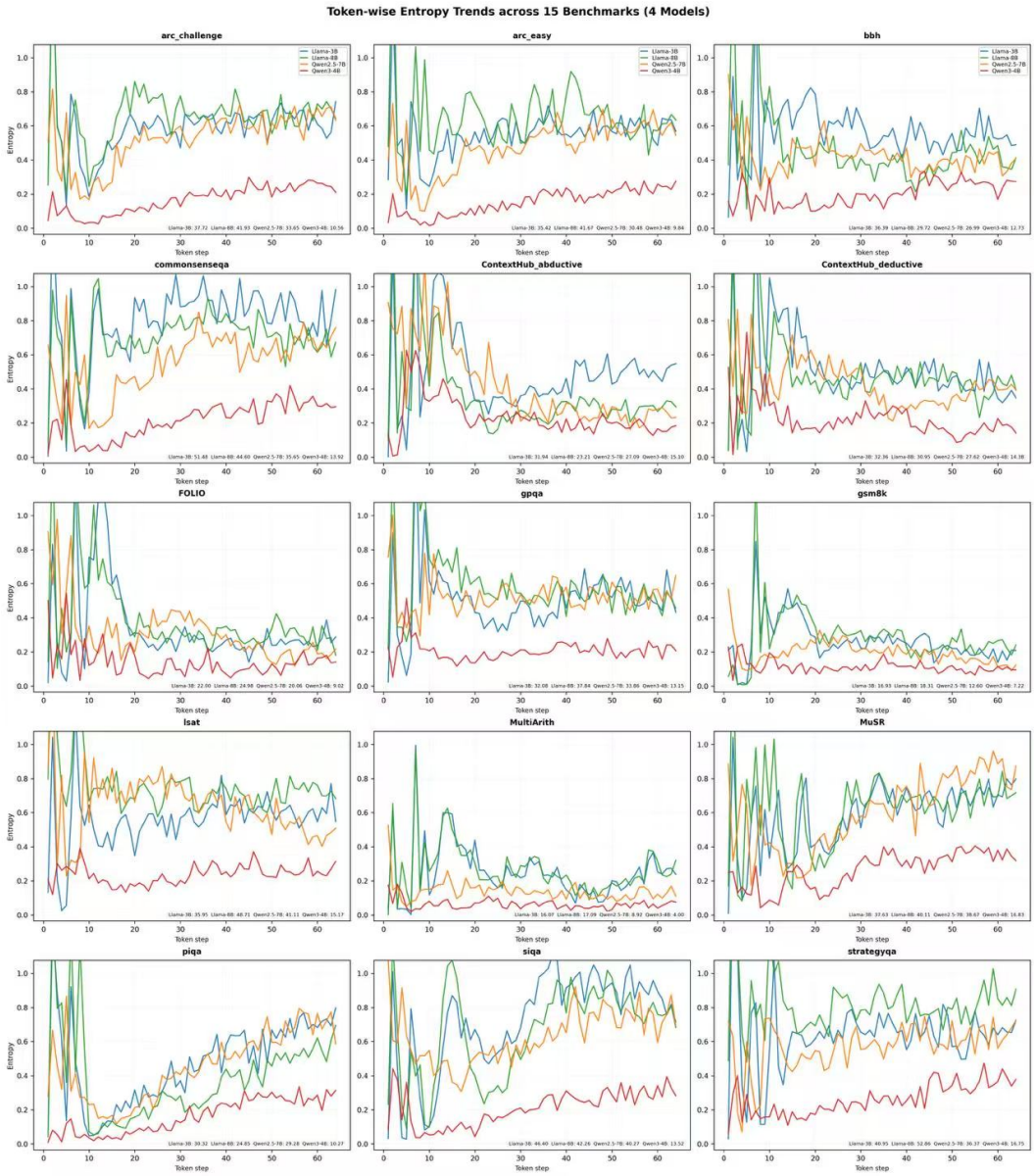

Figure 5: Entropy trajectories for all evaluated models and benchmarks. Each line represents the average entropy evolution for a specific model on the benchmark. Each subplot corresponds to a specific benchmark, with lines of different colors representing the average entropy trajectory for each model.

A.3 Visualization of Entropy Trajectories of All Models and Benchmarks

As shown in Figure 5, the entropy trajectories reveal distinct uncertainty dynamics across models. Qwen3-4B (red line) consistently maintains the lowest entropy levels across almost all benchmarks, exhibiting a stable trajectory with minimal fluctuation. This indicates high confidence in token generation. Conversely, Llama-3B (blue) and Llama-8B (green) generally display higher entropy with significant volatility, particularly in the early decoding steps. Qwen2.5-7B (orange) typically falls between these extremes.

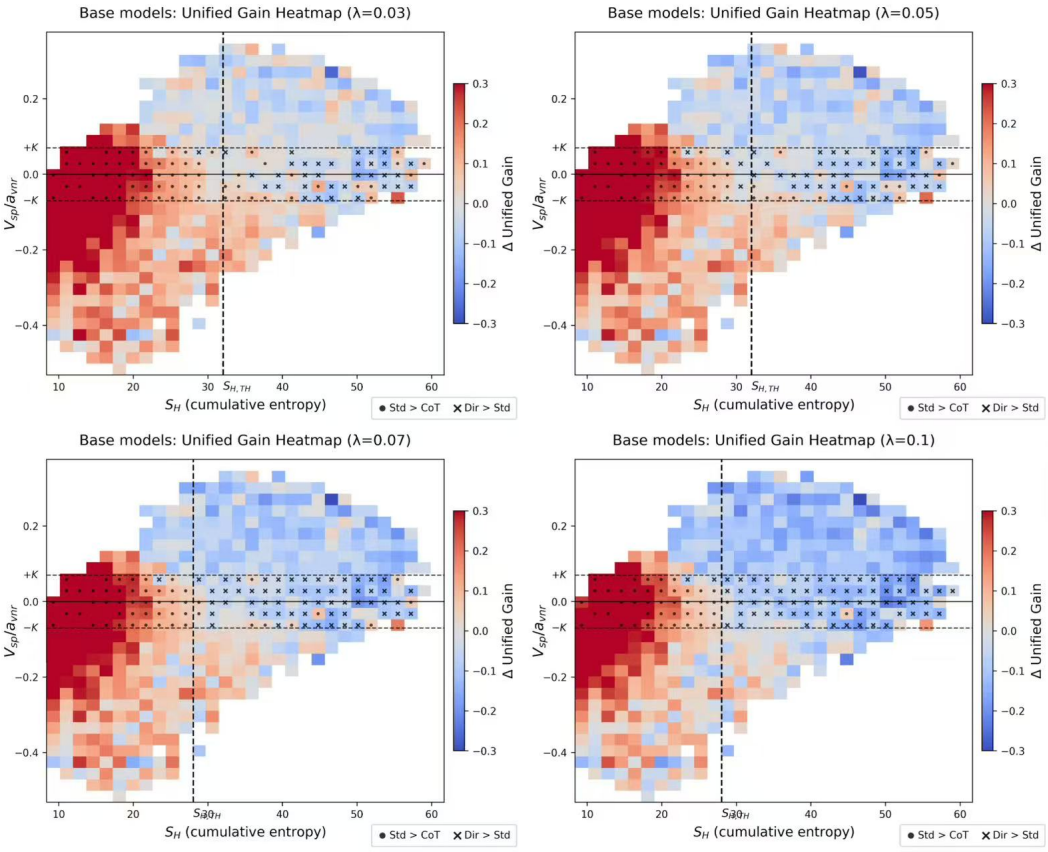

Figure 6: Unified Gain Heatmap with various λ (0.03,0.05,0.07,0.10) on **base models**.

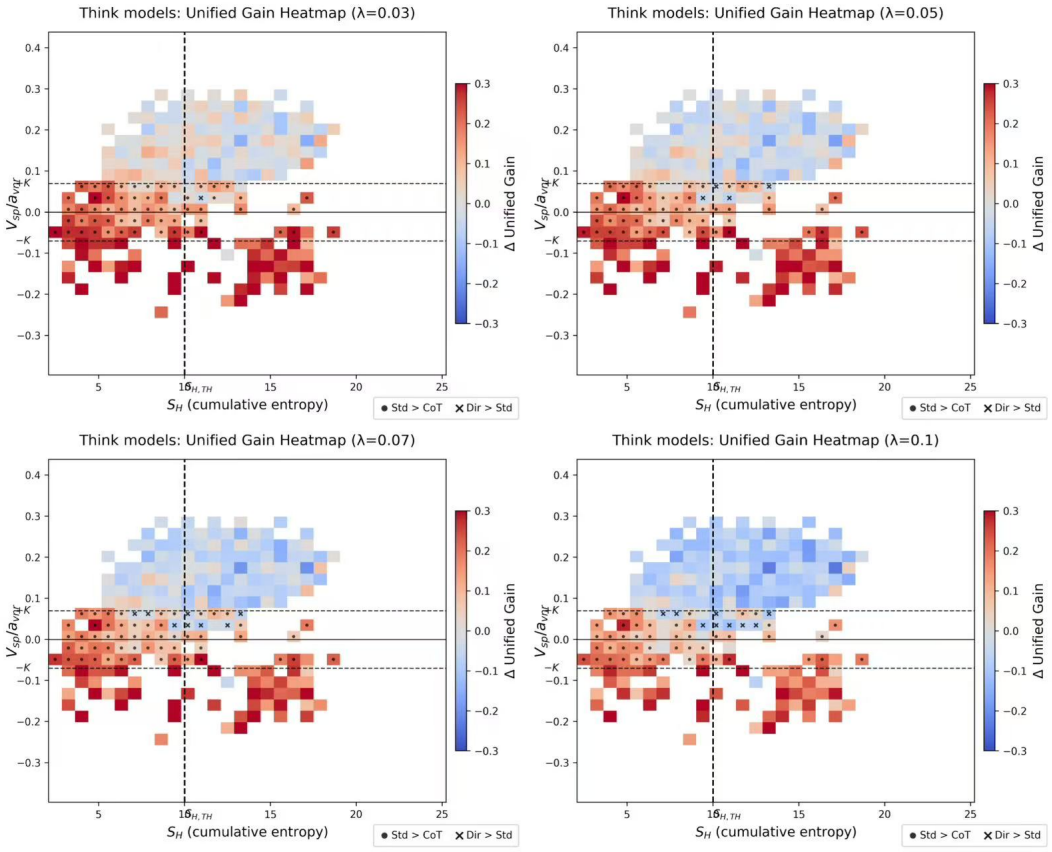

Figure 7: Unified Gain Heatmap with various λ (0.03,0.05,0.07,0.10) on **think models**.

A.4 UNIFIED GAIN HEATMAP

Figure 6 and 7 visualizes how the relative utility of *CoT* versus *Direct* varies across regions of the entropy-dynamics manifold. Since accuracy improvements alone do not reflect the substantial token overhead of explicit reasoning, we introduce a *Unified Gain* that jointly accounts for correctness and generation cost.

Motivation: why unified gain? Standard comparisons such as $\mathrm{Acc}(\mathrm{CoT}) - \mathrm{Acc}(\mathrm{Direct})$ ignore token usage, yet *CoT* often increases output length by orders of magnitude (Tables 1, 2). For deployment, the relevant question is not only "which is more accurate?" but "which is more cost-effective under a chosen accuracy–cost trade-off?" Unified Gain provides a single scalar objective aligned with this requirement, enabling a fine-grained visualization over the manifold coordinates.

Relation to the main-paper shorthand definition. In the main paper, we use a shorthand unified gain $U(m) = \mathrm{Acc}(m) - \lambda \cdot \mathrm{Cost}(m)$ for readability, where $\mathrm{Acc}(m)$ and $\mathrm{Cost}(m)$ can be understood as *already-aggregated* statistics (e.g., dataset-level averages). In this appendix, we restate the same objective at the *instance level* to remove ambiguity and to support heatmap construction: we first define a per-instance utility $U_x(m) = \mathbb{I}[\mathrm{correct}_m(x)] - \lambda \cdot \tilde{T}_m(x)$, and then aggregate by expectation/average over a set of instances. This formulation makes explicit (i) that accuracy corresponds to the mean of correctness indicators, and (ii) what token cost means and how it is normalized. When $\mathrm{Acc}(m) = \mathbb{E}_x[\mathbb{I}[\mathrm{correct}_m(x)]]$ and $\mathrm{Cost}(m) = \mathbb{E}_x[\tilde{T}_m(x)]$, the appendix definition reduces exactly to the main-paper shorthand. We normalize tokens as $\tilde{T}_m(x) = T_m(x)/1000$ so that λ is comparable across datasets and models and remains numerically interpretable.

Definition and computation. For an instance x and decoding mode $m \in \{\mathrm{Direct}, \mathrm{CoT}\}$, let $\mathbb{I}[\mathrm{correct}_m(x)] \in \{0, 1\}$ be the correctness indicator and $T_m(x)$ be the number of generated output tokens.[1] We define the per-instance utility as

$$U_x(m) = \mathbb{I}[\mathrm{correct}_m(x)] - \lambda \cdot \frac{T_m(x)}{1000}, \qquad \Delta U_x(\mathrm{CoT}, \mathrm{Direct}) = U_x(\mathrm{CoT}) - U_x(\mathrm{Direct}), \quad (8)$$

where $\lambda \geq 0$ is a cost-penalty coefficient and the token cost is normalized by 1000 to keep λ in a human-interpretable range.[2] A positive ΔU_x means *CoT* is more cost-effective than *Direct* on x under the chosen λ, and a negative value means the opposite.

From per-instance gain to a heatmap over the manifold. To construct Figure 2, we compute each instance's manifold coordinates using *Standard* probing:

$$\left(\frac{V_{\mathrm{sp}}(x)}{a_{\mathrm{vnr}}(x)},\ S_H(x) \right),$$

and then discretize the 2D plane into bins. Each cell reports the average unified gain among instances that fall into the corresponding bin:

$$\Delta U_{\mathrm{cell}} = \mathbb{E}\left[\Delta U_x(\mathrm{CoT}, \mathrm{Direct}) \mid x \in \mathrm{cell} \right].$$

This visualization directly links the observed efficiency frontier to entropy dynamics, without requiring any learned router.

How to read the heatmap (interpretation of the main-paper figure). The heatmap in Figure 2 shows a structured separation aligned with our routing logic:

- **Convergent region (large negative $V_{\mathrm{sp}}/a_{\mathrm{vnr}}$):** cells tend to have higher (often positive) ΔU, indicating that when entropy decreases reliably relative to volatility, *CoT* is more likely to justify its extra tokens.
- **Divergent/unstable region (positive $V_{\mathrm{sp}}/a_{\mathrm{vnr}}$):** cells are dominated by negative ΔU, consistent with the observation that *CoT* often overthinks and wastes tokens when uncertainty drifts upward.

[1] In our experiments, $T_m(x)$ corresponds to the recorded output token length in the evaluation logs.

[2] The main paper figure uses $\lambda = 0.05$ for base models as a representative operating point. Larger λ increasingly favors shorter outputs; smaller λ approaches accuracy-only comparison.

- **Uncertainty-load effect (S_H):** even when trend is not strongly positive, very large S_H corresponds to early-stage uncertainty overload, where the additional *CoT* budget is less likely to convert into correctness, pushing ΔU downward.

Overall, the heatmap provides an empirical justification for using the two-axis manifold $(V_{\text{sp}}/a_{\text{vnr}}, S_H)$ as the decision space: it exposes where reasoning is truly cost-effective and where it is not.

A.4.1 Cross-dataset Calibration of $S_{H,\text{TH}}$

Why calibration is needed? The uncertainty-overload threshold $S_{H,\text{th}}$ is *model-dependent*. Per-model *empirical thresholds* are obtained by binary search over candidate values, yielding $S_{H,\text{th}}$=32 for Llama-3.2-3B, Llama-3.1-8B, and Qwen2.5-7B, 39 for Qwen2.5-32B, and $10/30/20$ for the reasoning models Qwen3-4B-T, DeepSeek-R1-Distill-Qwen-7B, and Qwen3.5-35B-A3B, respectively. While such thresholds can work well for the models studied in this paper, they are obtained by analyzing a closed set of model–dataset results and therefore may not transfer to a new model with different calibration, decoding behavior, or built-in "thinking" biases. Repeating a full benchmark sweep to re-tune $S_{H,\text{th}}$ for every new model is costly and often impractical in cold-start deployment. To address this, we introduce a lightweight *calibration* procedure used by EDRM-Global/Inst-C: for any unseen model (and optionally a new collection of datasets), it estimates a suitable $S_{H,\text{th}}$ from a small number of sampled instances via a simple heuristic, enabling fast and flexible threshold selection without additional training or exhaustive evaluation.

This parts details the cross-dataset heuristic calibration used by EDRM-Global/Inst-C. Consider a fixed model evaluated on J datasets $\{\mathcal{D}_j\}_{j=1}^{J}$. For each dataset, we sample n instances, run N-step probing, and compute per-instance features. We then form dataset-level means

$$\mu_S^{(j)} = \frac{1}{n} \sum_{x \in \mathcal{D}_{\text{sample}}^{(j)}} S_H(x), \qquad \mu_V^{(j)} = \frac{1}{n} \sum_{x \in \mathcal{D}_{\text{sample}}^{(j)}} V_{\text{sp}}(x). \tag{9}$$

Let

$$M = \left| \left\{ j \in \{1, \ldots, J\} : \mu_V^{(j)} < 0 \right\} \right| \tag{10}$$

be the number of datasets whose average probing dynamics are convergent ($\mu_V^{(j)} < 0$). Sort $\{\mu_S^{(j)}\}_{j=1}^{J}$ in ascending order to obtain $s_{(1)} \le \cdots \le s_{(J)}$, and define

$$S_{H,\text{th}} = \left\lfloor s_{(M)} \right\rfloor, \tag{11}$$

where we clamp the index as $M \leftarrow \max(1, \min(M, J))$ to handle boundary cases. The full calibration procedure is presented in Algorithm 3.

A.5 Routing Decision Logic Analysis

This section analyzes the design rationale of EDRM's three-branch routing (*Direct*, *Standard*, *CoT*), the decision boundaries in Algorithm 1, and how these boundaries relate to the entropy-dynamics descriptors $(S_H, V_{\text{sp}}, a_{\text{vnr}})$.

Why three branches (Direct / Standard / CoT)? We adopt three regimes because they span the practical efficiency–reliability spectrum while remaining deployable across both base and think-enabled models:

- **Direct (minimum compute, maximum robustness against overthinking).** Direct is the lowest-cost mode and is empirically strong on retrieval-heavy or low-reasoning tasks, where step-by-step deliberation often yields marginal or negative gains while substantially increasing token cost. Direct is also a conservative choice under decoding drift risk.
- **Standard (neutral default and probing anchor).** Standard uses a minimally biased prompt (no explicit CoT instruction), providing a stable "native" decoding state for probing and serving as the middle option when the evidence for either extreme is insufficient.
- **CoT (maximum compute when progressive convergence is detected).** CoT is reserved for cases where early decoding shows stable uncertainty reduction (a reliable convergence signature). In such cases, additional structured reasoning is most likely to translate into accuracy gains.

Algorithm 3 Heuristic Calibration of $S_{H,\text{th}}$ Across Datasets

Require: Datasets $\{\mathcal{D}_j\}_{j=1}^{J}$, per-dataset sample size n, probe length N
Ensure: Calibrated threshold $S_{H,\text{th}}$

for $j = 1$ **to** J **do**
  Sample $\mathcal{D}_{\text{sample}}^{(j)} \subset \mathcal{D}_j$ with $|\mathcal{D}_{\text{sample}}^{(j)}| = n$
  for each $x \in \mathcal{D}_{\text{sample}}^{(j)}$ **do**
    Run N-step probing on x to obtain entropy sequence E_x
    Compute $S_H(x)$ and $V_{\text{sp}}(x)$
  end for
  $\mu_S^{(j)} \leftarrow \frac{1}{n} \sum_{x \in \mathcal{D}_{\text{sample}}^{(j)}} S_H(x)$
  $\mu_V^{(j)} \leftarrow \frac{1}{n} \sum_{x \in \mathcal{D}_{\text{sample}}^{(j)}} V_{\text{sp}}(x)$
end for
$M \leftarrow \left| \left\{ j : \mu_V^{(j)} < 0 \right\} \right|$
Sort $\{\mu_S^{(j)}\}_{j=1}^{J}$ ascending to obtain $s_{(1)} \le \cdots \le s_{(J)}$
$M \leftarrow \max(1, \min(M, J))$
$S_{H,\text{th}} \leftarrow \lfloor s_{(M)} \rfloor$
return $S_{H,\text{th}}$

This tri-partition is also operationally important: it avoids forcing a binary decision (CoT vs. Direct) when the model is neither clearly convergent nor clearly divergent, reducing misrouting on boundary cases.

Decision boundaries and their meaning in the entropy feature embedding. EDRM uses two complementary signals: a *direction–stability* coupling via $(V_{\text{sp}}, a_{\text{vnr}})$ and an *uncertainty-load* guardrail via S_H. Concretely, Algorithm 1 implements:

- **Divergence / drift region ⇒ Direct.** If $V_{\text{sp}} > k \cdot a_{\text{vnr}}$, the entropy trajectory has an overall *increasing* tendency, and the confidence does not improve with steps. This is a high drift-risk regime: extra reasoning tokens are likely to amplify exploration or produce verbose but ungrounded continuations. Therefore, we route to *Direct*.
- **Convergence region ⇒ CoT.** If $V_{\text{sp}} < -k \cdot a_{\text{vnr}}$, the trajectory exhibits a *stable monotone decrease* relative to its volatility. This indicates progressive uncertainty reduction, where structured reasoning is most likely to be beneficial; we route to *CoT*.
- **Uncertainty overload guardrail ⇒ Direct.** Even when V_{sp} is not strongly positive, if $V_{\text{sp}} > 0$ and $S_H > S_{H,\text{th}}$, the model accumulates large total uncertainty during early decoding. Empirically, this region often corresponds to poor calibration or inability to settle on a coherent reasoning path; CoT tends to be long and unreliable. We thus route to *Direct* to cap cost and reduce drift.
- **Otherwise ⇒ Standard.** When neither strong convergence nor strong divergence is detected, we select *Standard* as the default middle ground.

How k and $S_{H,\text{th}}$ interact with the descriptors. Parameter k controls how much negative/positive trend is required *relative to volatility* to trigger CoT/Direct. Intuitively, a_{vnr} serves as a noise-aware normalization: under high volatility, $|V_{\text{sp}}|$ must be larger to be trusted as a genuine monotone trend. Threshold $S_{H,\text{th}}$ acts as a capacity-dependent ceiling on early uncertainty load (base vs. reasoning models differ), preventing expensive CoT in regimes where the model is already "lost" in early decoding.

Why use the dataset mean of V_{sp} (i.e., $\bar{V}_{\text{sp}} = \mathbb{E}[V_{\text{sp}}(x)]$) instead of computing V_{sp} on the averaged entropy curve? For dataset-level routing, we aggregate descriptors as $\bar{S}_H = \frac{1}{|\mathcal{D}_s|} \sum_x S_H(x)$, $\bar{V}_{\text{sp}} = \frac{1}{|\mathcal{D}_s|} \sum_x V_{\text{sp}}(x)$, $\bar{a}_{\text{vnr}} = \frac{1}{|\mathcal{D}_s|} \sum_x a_{\text{vnr}}(x)$. A tempting alternative is to first average the token-level entropies across samples, $\tilde{H}_i = \frac{1}{|\mathcal{D}_s|} \sum_x H_i(x)$, and then compute $V_{\text{sp}}(\tilde{H}_{1:N})$. We intentionally avoid this for three reasons:

- **Non-commutativity (nonlinearity) of Spearman.** Spearman correlation is computed on *ranks* and is not a linear operator, hence in general

$$\mathrm{Spearman}\big(i, \mathbb{E}[H_i(x)]\big) \;\neq\; \mathbb{E}\big[\mathrm{Spearman}(i, H_i(x))\big].$$

  Averaging before ranking can change the ordering structure and distort the monotonicity signal.

- **Heterogeneity cancellation.** Many datasets contain mixed instance types (some clearly convergent, others divergent). Averaging token-level entropies across such a mixture often yields an artificially smooth curve that can appear weakly decreasing even when a large fraction of instances are increasing (or vice versa). In contrast, averaging per-instance $V_{\text{sp}}(x)$ preserves the sign/magnitude distribution of instance-level trends, which is exactly what dataset-level routing aims to summarize.

- **Robustness to irregularities and early termination.** In practice, some instances may terminate early ($T < N$) and are treated as confident (routed to *Standard* without descriptor-based decision). Constructing an averaged entropy curve requires additional alignment/padding choices that introduce bias. Computing $V_{\text{sp}}(x)$ per valid instance (with $T = N$) and then averaging avoids such artifacts and matches the actual routing semantics.

Empirically, we find $\bar{V}_{\text{sp}}$ computed as the mean of per-instance trends aligns better with dataset-level CoT gains and yields more stable routing under sampling (cf. low variances reported in Tables 2).

A.6 MLP Variants Design and Training Details

This subsection details the data construction, variant design, and training setup for **EDRM-MLP**. The router is trained as an instance-level classifier over three regimes $\mathcal{M} \in \{\textit{Direct}, \textit{Standard}, \textit{CoT}\}$ using entropy-dynamics signals extracted from an N=64-step *Standard* probing decode.

Training targets (multi-label vs. priority single-label). For each instance, we evaluate all three decoding regimes and derive correctness indicators $\mathbf{y} = [y_D, y_S, y_C] \in \{0,1\}^3$, where y_m=1 iff mode m answers correctly. This yields the **original multi-label target** (possibly multi-hot, e.g., $[1,1,0]$; or $[0,0,0]$ when all fail). To incorporate a token-efficiency preference, we also construct a **priority-constrained single-label target** $\mathbf{y}'$ by mapping multi-hot labels to a one-hot vector using the rule *Direct*>*Standard*>*CoT*; samples with $[0,0,0]$ are filtered out in this setting.

Input variants (3D / 64D / 67D). All variants share the same probing source: we take the token-level entropy sequence $E = \{H_1, \ldots, H_T\}$ recorded in *Standard* decoding logs and align it to a fixed length N=64 by truncation or zero-padding: if $T \geq N$, keep the first N entropies; otherwise pad E with zeros to length N. We then form three input representations: (i) **3D descriptors**: $(S_H, V_{\text{sp}}, a_{\text{vnr}})$, where $S_H = \sum_{i=1}^{N} H_i$, V_{sp} is the Spearman trend, and a_{vnr} is the Von Neumann ratio; (ii) **64D trajectory**: the aligned entropy vector $(H_1, \ldots, H_N)$; (iii) **67D hybrid**: concatenation of the 64D trajectory and the 3D descriptors. Combining the two label strategies with the three input forms yields the six ablation variants used in our experiments.

Data construction and split. We build per-instance features and labels by merging: (a) the entropy trajectory from *Standard* probing logs, and (b) per-instance descriptors and correctness labels computed by the EDRM evaluation pipeline. To keep training lightweight while preserving coverage, we perform **10% stratified sampling** for training: for each dataset, we group instances by the 8 possible label patterns (e.g., $000, 001, \ldots, 111$) and sample 10% from each group (at least one sample when available). The remaining instances are used for testing.

Model and optimization. We use a compact 3-layer MLP that outputs 3 logits. Inputs are standardized with a `StandardScaler` fitted on the training split. For **multi-label** training, we use `BCEWithLogitsLoss` with per-class `pos_weight` (computed as neg/pos on the training set) to mitigate label imbalance. For **single-label** training, we use `CrossEntropyLoss` with inverse-frequency class weights. Unless otherwise specified, we train with Adam, learning rate 10^{-3}, batch size 32, hidden dimension 128, weight decay 10^{-4}, and 100 epochs (we use 120 epochs for the 64D multi-label variant in our default scripts).

B Appendix: Experimental Details

B.1 Datasets and Models

We evaluate our method on 15 diverse benchmarks spanning four key reasoning categories: mathematical reasoning, commonsense and knowledge reasoning, scientific reasoning, and formal / logical reasoning. All datasets follow standard evaluation protocols in prior reasoning research.

Mathematical reasoning.

- **GSM8K**: Elementary school math problems requiring multi-step numerical reasoning.
- **MultiArith**: Arithmetic word problems testing basic numerical reasoning.
- **BBH**: A challenging subset of Big-Bench Hard focusing on complex symbolic and logical reasoning.

Commonsense & knowledge reasoning.

- **CommonsenseQA (CSQA)**: Commonsense understanding via multiple-choice questions.
- **StrategyQA**: Implicit multi-hop reasoning to answer binary questions.
- **PIQA**: Physical commonsense about plausible everyday actions.
- **SIQA**: Social commonsense and causal reasoning in interactive scenarios.
- **MuSR**: Murder mystery-style narrative causal inference.

Scientific reasoning.

- **ARC-Challenge (ARC-C)**: Hard multiple-choice science test questions.
- **ARC-Easy (ARC-E)**: Simpler science test questions.
- **GPQA**: Graduate-level scientific reasoning questions focused on biology and STEM.

Formal & deductive reasoning.

- **FOLIO**: First-order logic reasoning problems.
- **LSAT**: Logical reasoning tasks from the Law School Admission Test.
- **ContextHub_abductive (CH-Abd)**: Abductive reasoning from observations to explanations.
- **ContextHub_deductive (CH-Ded)**: Strict deductive reasoning from premises.

B.1.1 Models Details

We evaluate seven instruction-tuned LLMs spanning both *base* and *reasoning-enhanced* families to test EDRM's cross-model robustness. All models are used in their public Instruct checkpoints and are decoded with greedy generation for all regimes to ensure comparability.

Evaluated models.

- **Llama-3.2-3B-Instruct** Grattafiori et al. (2024): a lightweight 3B parameter base instruct model.
- **Llama-3.1-8B-Instruct** Grattafiori et al. (2024): an 8B parameter base instruct model.
- **Qwen2.5-7B-Instruct** Hui et al. (2024): a 7B parameter base instruct model from the Qwen family.
- **Qwen2.5-32B-Instruct** Hui et al. (2024): a 32B parameter base instruct model, used to validate cross-scale generalization.
- **Qwen3-4B-Instruct-2507** Yang et al. (2025): a 4B parameter *reasoning-enhanced* model with built-in `think` behavior, often producing verbose deliberation by default; we include it as an "over-reasoning" stress test.

- **DeepSeek-R1-Distill-Qwen-7B**: a 7B reasoning-distilled model built on the Qwen2.5-7B base, producing extremely long CoT traces.
- **Qwen3.5-35B-A3B**: a 35B MoE reasoning model representing the latest generation of verbose-reasoning architectures.

Decoding regimes and model-specific control. We apply the same three regimes (*Direct*, *Standard*, *CoT*) across all models (Section 3). For reasoning-enhanced models (Qwen3-4B-Instruct-2507, DeepSeek-R1-Distill-Qwen-7B, Qwen3.5-35B-A3B), we additionally control the internal thinking behavior:

- **Direct**: force final-answer-only output (and disable/avoid explicit `think` traces when supported by the model interface).
- **Standard**: use a neutral instruction without step-by-step cues; for reasoning models we also disable/avoid `think` so probing reflects minimally perturbed decoding dynamics.
- **CoT**: explicitly request step-by-step reasoning; for reasoning models we enable/allow `think` to realize the intended reasoning mode.

This design ensures that "CoT" corresponds to increased deliberation budget, while "Direct/Standard" reflect low-deliberation decoding, which is crucial for a fair assessment of routing under overthinking-prone models.

Generation and evaluation settings. All experiments use greedy decoding (temperature $= 0$) with identical stopping criteria across regimes. The maximum generation length is set to 4096 tokens. For instance-level routing, the probing phase uses $N{=}64$ steps under the *Standard* regime before the router selects the final decoding mode.

B.2 Prompts Design and Decoding Details

This subsection details the prompt templates and decoding configurations for the three regimes (*Direct*, *Standard*, *CoT*) across different model families.

Prompt templates by decoding regime. We design task-specific prompts for two categories: **Answer tasks** (mathematical reasoning: `gsm8k`, `MultiArith`, `MATH`) and **Choice tasks** (multi-choice QA: all other benchmarks). Table 5 summarizes the prompts for each decoding regime.

Table 5: Prompt templates for each decoding regime across task types.

Regime	Prompt Suffix
Direct	Answer: "Your answer must not include any reasoning step. You must only write your answer directly. You only output 'The answer is ¡answer¿'."
	Choice: "Your answer must not include any reasoning. Write the answer: 'Answer: ¡Your Answer Letter Choice¿'"
Standard	Both: "{question}" (no additional instruction)
CoT	Both: "Let's think step by step."

Model-specific chat templates. Table 6 shows the official chat templates used for each model family.

Handling reasoning-enhanced models. For models with built-in thinking capabilities (e.g., Qwen3-4B-Instruct-2507), we apply additional control via the `enable_thinking` parameter and system prompts, as summarized in Table 7.

Table 6: Chat templates for different model families.

Model Family	Chat Template
Llama-3.x Instruct	`<|begin_of_text|>...user...<|eot_id|>...assistant...`
Qwen2.5 / Qwen3	`<|im_start|>user{prompt}<|im_end|>...<|im_start|>assistant`

Table 7: Thinking mode control for reasoning-enhanced models.

Regime	`enable_thinking`	Additional Control
Direct / *Standard*	False	Qwen3.5: prepend "/no_think" system prompt
CoT	True	None (allow thinking traces)

This design ensures that the three decoding regimes correspond to distinct levels of deliberation budget: *Direct* forces minimal output, *Standard* provides neutral prompting, and *CoT* explicitly solicits extended reasoning.

B.3 Evaluation Metrics Details

This subsection details the computation of accuracy, token consumption, and consistency metrics used throughout our experiments.

Accuracy and token consumption. For a decoding mode $m \in \{\textit{Direct}, \textit{Standard}, \textit{CoT}\}$, we compute dataset-level accuracy and average token consumption as:

$$\text{Acc}(m) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbb{I}[\text{correct}_m(x)], \qquad \text{AvgTok}(m) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} T_m(x), \tag{12}$$

where $\mathbb{I}[\text{correct}_m(x)] \in \{0, 1\}$ indicates whether mode m produces a correct answer for instance x, and $T_m(x)$ denotes the number of generated tokens.

Dataset-level routing (EDRM-Global). When routing selects mode $m^*(x)$ for each instance, the effective accuracy and token cost are:

$$\text{Acc}_{\text{Global}} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbb{I}[\text{correct}_{m^*(x)}(x)], \qquad \text{AvgTok}_{\text{Global}} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} T_{m^*(x)}(x). \tag{13}$$

Since EDRM-Global probes only a small calibration subset (n=50 samples, < 5 tokens per instance on average), we do not include probing overhead in the reported token consumption.

Instance-level routing (EDRM-Inst). Instance-level routing requires probing every instance, so we strictly account for all token costs. For an instance x routed to mode $m^*(x)$, the total token cost is:

$$T_{\text{total}}(x) = T_{m^*(x)}(x) + T_{\text{probe}}(x) + T_{\text{fallback}}(x), \tag{14}$$

where:

- $T_{\text{probe}}(x) = N$ if $m^*(x) \neq$ *Standard* (probing tokens are consumed when Standard is not selected), otherwise $T_{\text{probe}}(x) = 0$;
- $T_{\text{fallback}}(x) = T_{\textit{Direct}}(x)$ if $m^*(x) \neq$ *Direct* (fallback compensation adds Direct branch), otherwise $T_{\text{fallback}}(x) = 0$.

The effective accuracy incorporates fallback compensation: if $m^*(x) \neq$ *Direct*, the instance is considered correct when either $m^*(x)$ or *Direct* produces a correct answer. Formally:

$$\mathbb{I}[\text{correct}_{\text{Inst}}(x)] = \begin{cases} \mathbb{I}[\text{correct}_{m^*(x)}(x)] \vee \mathbb{I}[\text{correct}_{\textit{Direct}}(x)], & m^*(x) \neq \textit{Direct} \\ \mathbb{I}[\text{correct}_{\textit{Direct}}(x)], & m^*(x) = \textit{Direct} \end{cases} \tag{15}$$

The dataset-level metrics are then:

$$\text{Acc}_{\text{Inst}} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbb{I}[\text{correct}_{\text{Inst}}(x)], \qquad \text{AvgTok}_{\text{Inst}} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} T_{\text{total}}(x). \tag{16}$$

Routing consistency. For experiments requiring multiple trials (8 random seeds $\{0, 1, 2, 3, 11, 12, 13, 14\}$), we report the routing decision distribution as the ratio $D : S : C$, where D, S, and C denote the number of trials (out of 8) that select *Direct*, *Standard*, and *CoT* respectively. This ratio directly reflects the stability of dataset-level routing decisions: a concentrated distribution (e.g., $8 : 0 : 0$) indicates high consistency across random seeds, while a dispersed distribution (e.g., $2 : 3 : 3$) indicates sensitivity to sampling variation.

B.4 Additional EDRM-Global Results

EDRM-Global with Llama-3.1-8B-Instruct. The following table presents the comprehensive per-dataset results of EDRM-Global (both E&C variants) for `Llama-3.1-8B-Instruct` across all 15 benchmarks. EDRM-Global-E achieves 67.27% accuracy with 164.4 average tokens, reducing token cost by 40.8% compared to CoT (277.8 tokens).

Table 8: EDRM-Global(E&C) Performance comparison of `Llama-3.1-8B-Instruct` across 15 benchmarks. EDRM-Global-E achieves 67.27% accuracy with 164.4 average tokens, reducing token cost by 40.8% compared to CoT (277.8 tokens).

Dataset	Direct		Standard		CoT		EDRM-Global-E			EDRM-Global-C		
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C
ARC-C	81.06	4.0	81.14	29.7	86.60	270.1	81.08	10.4	6:2:0	81.12	23.3	2:6:0
ARC-E	92.76	4.0	92.72	16.1	94.19	235.2	93.09	69.4	1:5:2	93.09	70.9	0:6:2
BBH	54.29	4.0	60.95	149.5	65.33	303.2	64.78	284.0	0:1:7	64.78	284.0	0:1:7
CSQA	74.04	4.0	71.99	47.2	74.61	230.1	74.04	4.0	8:0:0	72.76	31.0	3:5:0
CH-Abd	31.04	4.0	39.21	240.0	39.92	284.3	39.92	284.3	0:0:8	39.92	284.3	0:0:8
CH-Ded	36.33	4.0	54.13	230.3	56.04	240.1	54.85	234.0	0:5:3	54.84	234.0	0:5:3
FOLIO	48.42	4.0	52.24	245.5	56.98	337.7	56.98	337.7	0:0:8	56.98	337.7	0:0:8
GPQA	36.16	11.5	33.71	327.2	27.01	1005.8	31.20	581.7	0:5:3	31.19	581.7	0:5:3
GSM8K	14.86	16.8	84.46	227.2	82.64	234.8	84.46	227.2	0:8:0	84.46	227.2	0:8:0
LSAT	50.74	4.0	50.74	213.7	49.85	491.7	50.63	222.2	1:6:1	50.63	222.2	1:6:1
MultiArith	42.50	7.0	98.83	116.9	98.00	125.7	98.83	116.9	0:8:0	98.83	116.9	0:8:0
MuSR	51.85	4.0	50.26	165.6	51.32	291.7	51.85	4.0	8:0:0	50.26	165.6	0:8:0
PIQA	81.99	4.0	82.59	42.9	83.19	216.8	82.89	146.7	1:2:5	82.89	146.7	1:2:5
SIQA	70.32	4.0	70.01	48.3	70.68	214.4	70.32	4.0	8:0:0	70.32	4.0	8:0:0
StratQA	62.62	4.0	62.71	15.6	69.78	255.7	65.34	102.8	2:3:3	65.35	104.2	1:4:3
Overall	57.31	5.0	66.08	127.1	68.11	277.8	67.27	164.4	–	67.15	172.1	–

EDRM-Global with Qwen2.5-7B-Instruct. The following table presents the per-dataset results of EDRM-Global (both E&C variants) for `Qwen2.5-7B-Instruct` across all 15 benchmarks. EDRM-Global-E achieves 71.66% accuracy with 191.8 average tokens, reducing token cost by 41.9% compared to CoT (330.3 tokens).

EDRM-Global with Qwen3-4B-Instruct-2507. The following table presents the per-dataset results of EDRM-Global (both E&C variants) for `Qwen3-4B-Instruct-2507` across all 15 benchmarks. EDRM-Global-E achieves 76.35% accuracy with 335.1 average tokens, reducing token cost by 47.8% compared to CoT (642.5 tokens).

EDRM-Global with Qwen2.5-32B-Instruct. The following table presents the per-dataset results of EDRM-Global (both E&C variants) for `Qwen2.5-32B-Instruct` across all 15 benchmarks. EDRM-Global-E achieves 78.89% accuracy with 194.6 average tokens, reducing token cost by 36.6% compared to CoT (306.7 tokens).

EDRM-Global with DeepSeek-R1-Distill-Qwen-7B. The following table presents the per-dataset results of EDRM-Global (both E&C variants) for `DeepSeek-R1-Distill-Qwen-7B` across all

Table 9: EDRM-Global(E&C) Performance comparison of `Qwen2.5-7B-Instruct` across 15 benchmarks. EDRM-Global-E achieves 71.66% accuracy with 191.8 average tokens, reducing token cost by 41.9% compared to CoT (330.3 tokens).

Dataset	Direct		Standard		CoT		EDRM-Global-E			EDRM-Global-C		
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C
ARC-C	90.44	3.0	90.87	198.3	91.72	289.7	90.49	27.4	7:1:0	90.50	27.4	7:1:0
ARC-E	95.75	3.0	96.51	157.8	96.21	270.5	95.75	3.0	8:0:0	95.75	3.0	8:0:0
BBH	53.91	3.4	68.47	223.7	71.07	330.4	70.09	290.4	0:3:5	70.09	290.4	0:3:5
CSQA	81.08	3.0	81.00	120.6	80.84	252.3	81.08	3.0	8:0:0	81.08	3.0	8:0:0
CH-Abd	34.21	3.1	52.08	375.5	52.71	432.5	52.71	432.5	0:0:8	52.71	432.5	0:0:8
CH-Ded	46.92	3.0	53.21	337.2	52.88	365.8	52.88	365.8	0:0:8	52.88	365.8	0:0:8
FOLIO	53.82	3.0	67.61	356.5	69.44	398.4	69.44	398.4	0:0:8	69.44	398.4	0:0:8
GPQA	39.51	3.0	36.16	670.9	29.91	806.3	35.38	687.8	0:7:1	35.38	687.8	0:7:1
GSM8K	22.59	6.6	89.76	292.8	89.23	299.9	89.36	298.2	0:2:6	89.37	298.2	0:2:6
LSAT	61.35	3.1	56.39	492.1	57.68	545.9	57.68	545.9	0:0:8	57.68	545.9	0:0:8
MultiArith	61.17	5.8	98.83	174.7	98.83	182.3	98.83	182.3	0:0:8	98.83	182.3	0:0:8
MuSR	54.89	3.0	53.84	182.3	52.78	384.8	54.89	3.0	8:0:0	54.89	3.0	8:0:0
PIQA	86.62	3.4	82.43	119.4	87.32	235.5	86.62	3.4	8:0:0	86.62	3.4	8:0:0
SIQA	73.39	3.1	72.67	101.9	74.46	261.7	73.39	3.1	8:0:0	73.39	3.1	8:0:0
StratQA	62.10	3.0	66.11	151.4	68.47	259.7	62.10	3.0	8:0:0	63.10	40.1	6:2:0
Overall	62.28	3.4	71.43	240.6	72.44	330.3	71.66	191.8	–	71.76	195.5	–

Table 10: EDRM-Global(E&C) Performance comparison of `Qwen3-4B-Instruct-2507` across 15 benchmarks. EDRM-Global-E achieves 76.35% accuracy with 335.1 average tokens, reducing token cost by 47.8% compared to CoT (642.5 tokens).

Dataset	Direct		Standard		CoT		EDRM-Global-E			EDRM-Global-C		
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C
ARC-C	90.78	6.3	94.37	254.8	94.45	416.2	90.78	6.3	8:0:0	90.78	6.3	8:0:0
ARC-E	96.09	5.9	97.98	190.1	98.32	344.9	96.09	5.9	8:0:0	96.09	5.9	8:0:0
BBH	57.23	8.2	82.93	556.4	84.09	664.6	63.66	145.3	6:2:0	82.93	556.4	0:8:0
CSQA	75.92	6.3	80.43	256.2	80.43	414.5	75.92	6.3	8:0:0	75.92	6.3	8:0:0
CH-Abd	29.96	8.1	65.00	869.9	63.83	943.2	63.83	943.2	0:0:8	63.83	943.2	0:0:8
CH-Ded	54.00	6.2	82.46	634.4	82.87	676.6	82.87	676.6	0:0:8	82.88	676.6	0:0:8
FOLIO	60.71	4.7	77.41	816.7	76.83	865.3	76.83	865.3	0:0:8	76.83	865.3	0:0:8
GPQA	38.17	6.5	48.88	2611.3	47.99	2762.7	47.54	2285.7	1:7:0	48.88	2611.3	0:8:0
GSM8K	27.22	7.5	90.22	321.8	90.83	339.1	90.68	334.8	0:2:6	90.67	334.8	0:2:6
LSAT	66.70	4.8	82.16	1420.3	83.05	1550.7	76.36	889.5	3:5:0	82.16	1420.3	0:8:0
MultiArith	81.83	5.7	97.83	107.9	97.83	129.4	97.83	116.0	0:5:3	97.83	116.0	0:5:3
MuSR	58.33	5.7	57.80	1059.3	62.83	1401.7	58.33	5.7	8:0:0	58.33	5.7	8:0:0
PIQA	83.35	5.0	84.71	167.9	88.14	425.4	83.35	5.0	8:0:0	83.35	5.0	8:0:0
SIQA	71.90	5.6	74.87	239.8	77.02	398.6	71.90	5.6	8:0:0	71.90	5.6	8:0:0
StratQA	62.10	5.1	67.07	206.5	68.52	326.1	62.10	5.1	8:0:0	62.72	30.3	7:1:0
Overall	63.75	6.2	79.90	518.4	80.75	642.5	76.35	335.1	–	78.42	404.2	–

Table 11: EDRM-Global(E&C) Performance comparison of `Qwen2.5-32B-Instruct` across 15 benchmarks. EDRM-Global-E achieves 78.89% accuracy with 194.6 average tokens, reducing token cost by 36.6% compared to CoT (306.7 tokens).

Dataset	Direct		Standard		CoT		EDRM-Global-E			EDRM-Global-C		
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C
ARC-C	92.75	4.1	93.94	151.7	94.11	263.7	92.90	22.6	7:1:0	93.49	96.3	3:5:0
ARC-E	94.61	4.1	94.53	116.2	94.36	245.5	94.60	18.1	7:1:0	94.60	18.1	7:1:0
BBH	66.83	4.2	84.62	244.5	86.64	281.9	85.63	263.2	0:4:4	85.63	263.2	0:4:4
CSQA	86.98	4.1	85.42	103.7	85.01	223.1	86.78	16.6	7:1:0	86.78	16.6	7:1:0
CH-Abd	43.46	4.4	60.75	414.1	60.71	441.2	60.71	441.2	0:0:8	60.71	441.2	0:0:8
CH-Ded	62.29	4.4	65.71	303.5	64.58	325.0	64.58	325.0	0:0:8	64.58	325.0	0:0:8
FOLIO	64.29	4.0	75.58	380.4	76.08	392.8	76.08	392.8	0:0:8	76.08	392.8	0:0:8
GPQA	37.05	5.3	43.08	672.2	45.76	707.7	45.09	698.8	0:2:6	45.09	698.8	0:2:6
GSM8K	38.13	7.6	94.01	293.3	94.16	304.3	94.10	300.2	0:3:5	94.11	300.2	0:3:5
LSAT	76.61	4.7	72.05	462.4	72.84	508.8	72.74	503.0	0:1:7	72.75	503.0	0:1:7
MultiArith	94.83	6.6	98.83	174.1	98.83	182.4	98.83	178.3	0:4:4	98.83	178.3	0:4:4
MuSR	59.52	4.3	59.66	179.0	59.66	371.5	59.52	4.3	8:0:0	59.52	4.3	8:0:0
PIQA	90.75	5.7	89.83	124.2	91.95	221.0	90.29	65.0	4:4:0	90.29	65.0	4:4:0
SIQA	78.20	4.7	77.07	143.9	78.15	255.0	78.06	22.1	7:1:0	78.06	22.1	7:1:0
StratQA	74.37	4.0	70.87	145.5	68.73	211.5	73.93	21.7	7:1:0	72.18	92.5	3:5:0
Overall	71.35	4.7	78.37	238.7	78.57	306.7	78.89	194.6	–	78.75	205.3	–

15 benchmarks. EDRM-Global-E achieves 69.98% accuracy with 1342.9 average tokens; EDRM-Global-C routes all benchmarks to Standard (70.20% / 1389.0 tokens).

Table 12: EDRM-Global(E&C) Performance comparison of `DeepSeek-R1-Distill-Qwen-7B` across 15 benchmarks. EDRM-Global-E achieves 69.98% accuracy with 1342.9 average tokens; EDRM-Global-C routes all benchmarks to Standard (70.20% / 1389.0 tokens).

Dataset	Direct		Standard		CoT		EDRM-Global-E			EDRM-Global-C		
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C
ARC-C	69.11	2.9	80.89	1007.1	83.87	897.1	80.89	1007.1	0:8:0	80.89	1007.1	0:8:0
ARC-E	81.27	3.0	88.72	739.9	88.47	702.4	88.72	739.9	0:8:0	88.72	739.9	0:8:0
BBH	40.41	3.4	72.81	2550.8	71.31	2219.3	72.81	2550.8	0:8:0	72.81	2550.8	0:8:0
CSQA	57.90	3.0	65.60	1151.6	66.83	1010.8	65.60	1151.6	0:8:0	65.60	1151.6	0:8:0
CH-Abd	30.88	5.2	56.83	1915.5	58.33	1780.4	56.83	1915.5	0:8:0	56.83	1915.5	0:8:0
CH-Ded	56.29	4.8	65.87	924.3	67.79	885.4	65.87	924.3	0:8:0	65.88	924.3	0:8:0
FOLIO	52.82	2.7	73.09	1730.3	73.01	1791.6	73.09	1730.3	0:8:0	73.09	1730.3	0:8:0
GPQA	33.71	4.2	47.77	4964.9	52.90	4650.8	47.77	4964.9	0:8:0	47.77	4964.9	0:8:0
GSM8K	21.38	4.2	90.22	842.4	91.05	870.5	90.22	842.4	0:8:0	90.22	842.4	0:8:0
LSAT	40.73	2.6	52.23	2877.9	52.43	2850.9	52.23	2877.9	0:8:0	52.23	2877.9	0:8:0
MultiArith	58.67	2.8	97.67	354.6	97.17	418.0	97.67	354.6	0:8:0	97.67	354.6	0:8:0
MuSR	51.59	3.2	54.89	2052.6	53.17	2440.1	54.07	1540.3	2:6:0	54.89	2052.6	0:8:0
PIQA	71.33	3.0	77.80	964.4	77.86	877.6	76.18	724.1	2:6:0	77.80	964.4	0:8:0
SIQA	55.12	3.0	62.74	943.7	63.61	856.6	62.74	943.7	0:8:0	62.74	943.7	0:8:0
StratQA	56.68	3.0	61.54	820.3	62.07	744.5	60.93	718.2	1:7:0	61.48	820.3	0:8:0
Overall	53.28	3.5	70.20	1389.0	70.83	1314.2	69.98	1342.9	–	70.20	1389.0	–

EDRM-Global with Qwen3.5-35B-A3B. The following table presents the per-dataset results of EDRM-Global (both E&C variants) for `Qwen3.5-35B-A3B` across all 15 benchmarks. EDRM-Global-E achieves 83.13% accuracy with 1711.3 average tokens; EDRM-Global-C achieves 83.03% with 1651.0 tokens (3.1% savings vs. CoT 1703.4).

Table 13: EDRM-Global(E&C) Performance comparison of `Qwen3.5-35B-A3B` across 15 benchmarks. EDRM-Global-E achieves 83.13% accuracy with 1711.3 average tokens; EDRM-Global-C achieves 83.03% with 1651.0 tokens (3.1% savings vs. CoT 1703.4).

Dataset	Direct		Standard		CoT		EDRM-Global-E			EDRM-Global-C		
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C	$\overline{\text{Acc}}$(%)	$\overline{\text{Tok}}$	D:S:C
ARC-C	94.62	6.2	95.56	583.6	95.65	630.0	95.44	511.4	1:7:0	95.33	439.3	2:6:0
ARC-E	94.87	6.3	94.91	507.3	95.03	729.7	94.91	507.3	0:8:0	94.90	382.0	2:6:0
BBH	71.22	7.1	91.80	1691.9	92.29	1119.0	91.80	1691.9	0:8:0	91.80	1691.9	0:8:0
CSQA	84.11	6.3	85.75	614.7	85.75	1120.7	84.11	6.3	8:0:0	84.11	6.3	8:0:0
CH-Abd	45.33	8.1	62.29	4404.0	61.04	2087.6	62.29	4404.0	0:8:0	62.29	4404.0	0:8:0
CH-Ded	68.04	7.5	87.96	2623.5	87.67	2311.3	87.96	2623.5	0:8:0	87.96	2623.5	0:8:0
FOLIO	68.19	6.0	81.06	3237.0	81.15	3045.3	81.06	3237.0	0:8:0	81.06	3237.0	0:8:0
GPQA	60.94	6.3	77.68	6410.3	75.67	6294.9	77.68	6410.3	0:8:0	77.68	6410.3	0:8:0
GSM8K	42.76	6.6	93.56	491.0	93.56	708.6	93.56	491.0	0:8:0	93.56	491.0	0:8:0
LSAT	79.09	6.0	93.95	2177.8	92.86	1764.4	93.95	2177.8	0:8:0	93.95	2177.8	0:8:0
MultiArith	95.33	5.7	98.67	153.5	98.67	233.1	98.67	153.5	0:8:0	98.67	153.5	0:8:0
MuSR	58.99	6.1	66.67	3561.1	64.42	2441.7	64.75	2672.3	2:6:0	61.87	1339.2	5:3:0
PIQA	92.44	6.1	93.85	693.9	94.34	1119.8	92.44	6.1	8:0:0	92.44	6.1	8:0:0
SIQA	77.99	6.0	79.63	870.1	80.60	1749.3	78.20	114.0	7:1:0	78.20	114.0	7:1:0
StratQA	74.32	6.0	63.10	2321.5	62.66	2576.5	63.10	2321.5	0:8:0	63.10	2321.5	0:8:0
Overall	73.66	6.6	83.52	1895.2	83.34	1703.4	83.13	1711.3	–	83.03	1651.0	–

B.5 Additional EDRM-Instance Results

This subsection presents instance-level routing results for the remaining six models: `Llama-3.1-8B-Instruct`, `Qwen2.5-7B-Instruct`, `Qwen3-4B-Instruct-2507`, `Qwen2.5-32B-Instruct`, `DeepSeek-R1-Distill-Qwen-7B`, and `Qwen3.5-35B-A3B`.

EDRM-Instance with Llama-3.1-8B-Instruct. Table 14 presents the instance-level routing performance on `Llama-3.1-8B-Instruct`. EDRM-Inst-E achieves 70.04% accuracy with 153.9 tokens. EDRM-MLP achieves 69.14% accuracy with 119.9 tokens (best token efficiency among instance-level variants).

EDRM-Instance with Qwen2.5-7B-Instruct. Table 15 presents the instance-level routing performance on `Qwen2.5-7B-Instruct`. EDRM-Inst-E achieves 76.07% accuracy with 242.7 tokens. EDRM-Inst-C achieves 77.19% accuracy with 237.0 tokens.

EDRM-Instance with Qwen3-4B-Instruct-2507. Table 16 presents the instance-level routing performance on the reasoning-enhanced `Qwen3-4B-Instruct-2507`. Despite the model's inherent over-reasoning bias, EDRM effectively suppresses redundant deliberation. EDRM-Inst-E achieves 78.54% accuracy with 401.1 tokens (37.6% token reduction vs. CoT). EDRM-MLP achieves 79.33% accuracy with 422.0 tokens.

EDRM-Instance with Qwen2.5-32B-Instruct. Table 17 presents the instance-level routing performance on `Qwen2.5-32B-Instruct`. EDRM-Inst-C achieves 82.25% accuracy with 243.0 tokens (best accuracy among instance-level variants). EDRM-MLP achieves 81.31% accuracy with 215.8 tokens.

Table 14: EDRM instance-level Performance on `Llama-3.1-8B-Instruct`. EDRM-Inst-E achieves 70.04% accuracy with 153.9 tokens. EDRM-MLP achieves 69.14% accuracy with 119.9 tokens.

Dataset	Baselines						Ours							
	Direct		Standard		CoT		Token-Sign		EDRM-Inst-E		EDRM-Inst-C		EDRM-MLP	
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$ (%)	$\overline{\text{Tok}}$	Acc(%)	Tok
ARC-C	81.06	4.0	81.14	29.7	86.60	270.1	86.60	289.3	84.56	32.8	84.56	33.1	84.30	33.2
ARC-E	92.76	4.0	92.72	16.1	94.19	235.2	94.32	243.0	93.94	20.2	93.94	20.5	93.86	19.9
BBH	54.29	4.0	60.95	149.5	65.33	303.2	65.33	357.1	69.53	213.1	69.80	214.2	66.54	137.8
CSQA	74.04	4.0	71.99	47.2	74.61	230.1	74.61	263.4	75.51	56.5	75.60	57.3	76.00	46.7
CH-Abd	31.04	4.0	39.21	240.0	39.92	284.3	39.92	348.2	50.50	285.0	50.82	287.8	44.50	198.4
CH-Ded	36.33	4.0	54.13	230.3	56.04	240.1	56.04	303.7	53.21	222.8	54.26	229.7	53.83	215.6
FOLIO	48.42	4.0	52.24	245.5	56.98	337.7	56.98	394.6	63.95	323.2	64.00	323.7	57.64	211.1
GPQA	36.16	11.5	33.71	327.2	27.01	1005.8	36.16	52.1	43.08	363.4	43.50	378.3	41.96	363.9
GSM8K	14.86	16.8	84.46	227.2	82.64	234.8	82.64	298.8	76.57	245.4	76.90	246.7	84.08	250.1
LSAT	50.74	4.0	50.74	213.7	49.85	491.7	50.55	62.4	58.28	341.2	58.28	343.2	55.50	156.3
MultiArith	42.50	7.0	98.83	116.9	98.00	125.7	98.00	189.2	95.50	156.1	95.67	156.4	94.33	127.4
MuSR	51.85	4.0	50.26	165.6	51.32	291.7	51.32	352.8	56.08	124.0	56.37	134.1	57.01	174.9
PIQA	81.99	4.0	82.59	42.9	83.19	216.8	82.43	59.5	86.78	103.5	86.78	103.5	85.04	40.5
SIQA	70.32	4.0	70.01	48.3	70.68	214.4	70.62	252.8	71.65	45.0	71.76	45.3	72.88	58.5
StratQA	62.62	4.0	62.71	15.6	69.78	255.7	69.78	267.0	66.46	20.7	66.50	20.7	66.51	17.3
Overall	57.31	5.0	66.08	127.1	68.11	277.8	68.26	264.3	70.04	153.9	**71.67**	151.1	69.14	<u>119.9</u>

Table 15: EDRM instance-level Performance on `Qwen2.5-7B-Instruct`. EDRM-Inst-E achieves 76.07% accuracy with 242.7 tokens. EDRM-Inst-C achieves 77.19% accuracy with 237.0 tokens.

Dataset	Baselines						Ours							
	Direct		Standard		CoT		Token-Sign		EDRM-Inst-E		EDRM-Inst-C		EDRM-MLP	
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$ (%)	$\overline{\text{Tok}}$	Acc(%)	Tok
ARC-C	90.44	3.0	90.87	198.3	91.72	289.7	91.81	350.6	92.41	130.8	92.41	133.9	92.15	150.4
ARC-E	95.75	3.0	96.51	157.8	96.21	270.5	96.13	332.6	96.42	104.8	96.42	106.8	96.51	152.7
BBH	53.91	3.4	68.47	223.7	71.07	330.4	71.02	384.2	74.54	252.1	74.63	253.3	72.85	238.2
CSQA	81.08	3.0	81.00	120.6	80.84	252.3	80.84	313.5	81.82	93.3	82.13	96.0	82.64	89.8
CH-Abd	34.21	3.1	52.08	375.5	52.71	432.5	52.79	495.5	61.92	463.6	62.11	466.3	57.79	345.9
CH-Ded	46.92	3.0	53.21	337.2	52.88	365.8	52.88	429.8	65.25	377.9	65.49	382.3	63.38	327.3
FOLIO	53.82	3.0	67.61	356.5	69.44	398.4	69.44	462.3	74.00	410.2	74.09	410.6	73.42	405.3
GPQA	39.51	3.0	36.16	670.9	29.91	806.3	39.51	67.0	48.21	579.2	48.27	592.7	44.87	406.9
GSM8K	22.59	6.6	89.76	292.8	89.23	299.9	89.23	363.9	87.11	327.9	87.11	327.9	90.07	361.2
LSAT	61.35	3.1	56.39	492.1	57.68	545.9	61.05	249.6	68.68	464.8	68.68	465.0	66.60	282.1
MultiArith	61.17	5.8	98.83	174.7	98.83	182.3	98.83	246.3	97.67	210.9	97.67	210.9	99.17	249.1
MuSR	54.89	3.0	53.84	182.3	52.78	384.8	53.04	311.2	56.48	101.3	56.58	104.4	57.94	105.9
PIQA	86.62	3.4	82.43	119.4	87.32	235.5	87.32	298.9	87.70	105.1	87.70	105.7	88.47	136.8
SIQA	73.39	3.1	72.67	101.9	74.46	261.7	74.46	324.0	75.03	92.5	75.27	93.8	74.26	92.8
StratQA	62.10	3.0	66.11	151.4	68.47	259.7	68.47	323.2	66.90	133.7	67.15	135.8	66.42	139.4
Overall	62.28	3.4	71.43	240.6	72.44	330.3	72.78	356.7	76.07	242.7	**77.19**	237.0	75.35	<u>224.3</u>

Table 16: EDRM instance-level Performance on `Qwen3-4B-Instruct-2507`. EDRM-Inst-E achieves 78.54% accuracy with 401.1 tokens, reducing token cost by 37.6% compared to CoT (642.5 tokens).

Dataset	Baselines						Ours							
	Direct		Standard		CoT		Token-Sign		EDRM-Inst-E		EDRM-Inst-C		EDRM-MLP	
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$ (%)	$\overline{\text{Tok}}$	Acc(%)	Tok
ARC-C	90.78	6.3	94.37	254.8	94.45	416.2	94.45	480.1	92.15	114.3	92.32	129.2	93.17	194.4
ARC-E	96.09	5.9	97.98	190.1	98.32	344.9	98.32	408.1	96.55	96.4	96.79	102.8	97.43	155.0
BBH	57.23	8.2	82.93	556.4	84.09	664.6	84.09	728.6	77.82	403.2	78.80	425.0	82.16	491.7
CSQA	75.92	6.3	80.43	256.2	80.43	414.5	80.43	478.3	76.66	94.5	77.31	103.5	78.95	168.8
CH-Abd	29.96	8.1	65.00	869.9	63.83	943.2	63.83	1007.2	65.75	887.9	67.58	921.0	63.92	742.5
CH-Ded	54.00	6.2	82.46	634.4	82.87	676.6	82.88	740.6	84.50	674.9	85.18	688.1	81.71	581.8
FOLIO	60.71	4.7	77.41	816.7	76.83	865.3	76.83	929.3	79.65	809.1	81.21	854.6	79.40	706.9
GPQA	38.17	6.5	48.88	2611.3	47.99	2762.7	48.21	2555.2	51.56	1790.2	52.43	1971.7	54.46	1972.4
GSM8K	27.22	7.5	90.22	321.8	90.83	339.1	90.83	403.0	83.70	337.1	85.06	349.8	85.06	336.2
LSAT	66.70	4.8	82.16	1420.3	83.05	1550.7	83.05	1614.7	78.39	863.7	79.98	989.6	80.77	1015.6
MultiArith	81.83	5.7	97.83	107.9	97.83	129.4	97.83	193.4	95.17	142.6	95.17	142.6	95.50	137.4
MuSR	58.33	5.7	57.80	1059.3	62.83	1401.7	62.30	894.9	60.85	315.2	61.23	399.6	63.23	618.5
PIQA	83.35	5.0	84.71	167.9	88.14	425.4	88.14	489.3	84.28	122.6	84.39	123.8	84.71	150.0
SIQA	71.90	5.6	74.87	239.8	77.02	398.6	77.02	462.5	74.00	135.7	74.35	144.1	74.87	180.8
StratQA	62.10	5.1	67.07	206.5	68.52	326.1	68.52	390.0	64.24	121.5	65.00	133.9	66.20	152.9
Overall	63.75	6.2	79.90	518.4	**80.75**	642.5	80.74	682.3	78.54	<u>401.1</u>	79.88	426.7	79.33	422.0

Table 17: EDRM instance-level Performance on `Qwen2.5-32B-Instruct`. EDRM-Inst-C achieves 82.25% accuracy with 243.0 tokens. EDRM-MLP achieves 81.31% accuracy with 215.8 tokens.

Dataset	Baselines						Ours							
	Direct		Standard		CoT		Token-Sign		EDRM-Inst-E		EDRM-Inst-C		EDRM-MLP	
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$ (%)	$\overline{\text{Tok}}$	Acc(%)	Tok
ARC-C	92.75	4.1	93.94	151.7	94.11	263.7	94.11	326.6	93.43	142.4	93.52	146.2	93.69	145.1
ARC-E	94.61	4.1	94.53	116.2	94.36	245.5	94.61	65.7	94.78	111.1	94.78	113.7	94.78	121.5
BBH	66.83	4.2	84.62	244.5	86.64	281.9	86.64	345.9	83.85	236.2	83.85	236.9	85.34	222.4
CSQA	86.98	4.1	85.42	103.7	85.01	223.1	86.98	63.3	88.21	116.6	88.29	118.0	88.29	107.3
CH-Abd	43.46	4.4	60.75	414.1	60.71	441.2	60.71	505.2	68.00	495.8	68.04	496.5	62.79	369.6
CH-Ded	62.29	4.4	65.71	303.5	64.58	325.0	65.00	376.0	75.33	340.4	75.64	346.2	72.83	259.6
FOLIO	64.29	4.0	75.58	380.4	76.08	392.8	76.08	456.8	76.33	370.8	76.50	371.5	76.16	283.1
GPQA	37.05	5.3	43.08	672.2	45.76	707.7	45.76	771.7	51.79	602.6	51.79	615.3	49.55	480.1
GSM8K	38.13	7.6	94.01	293.3	94.16	304.3	94.16	368.3	92.87	331.1	92.87	331.1	94.77	350.2
LSAT	76.61	4.7	72.05	462.4	72.84	508.8	76.61	68.7	80.87	432.1	80.87	433.2	80.48	321.6
MultiArith	94.83	6.6	98.83	174.1	98.83	182.4	98.83	246.4	99.33	212.6	99.33	212.6	99.33	237.9
MuSR	59.52	4.3	59.66	179.0	59.66	371.5	59.52	67.8	60.71	123.0	60.81	126.7	62.57	159.1
PIQA	90.75	5.7	89.83	124.2	91.95	221.0	91.95	284.2	92.00	133.3	92.15	134.6	92.38	138.3
SIQA	78.20	4.7	77.07	143.9	78.15	255.0	78.97	194.1	81.12	139.9	81.43	141.8	80.35	124.8
StratQA	74.37	4.0	70.87	145.5	68.73	211.5	74.37	67.2	76.64	146.3	76.83	149.3	76.81	136.8
Overall	71.35	4.7	78.37	238.7	78.57	306.7	79.54	266.9	81.87	248.6	**82.25**	243.0	81.31	<u>215.8</u>

EDRM-Instance with DeepSeek-R1-Distill-Qwen-7B. Table 18 presents the instance-level routing performance on `DeepSeek-R1-Distill-Qwen-7B`. EDRM-Inst-C achieves 77.71% accuracy. EDRM-MLP achieves 74.26% accuracy with 1008.3 tokens, the best token efficiency among instance-level variants on this over-reasoning model.

Table 18: EDRM instance-level Performance on `DeepSeek-R1-Distill-Qwen-7B`. EDRM-Inst-C achieves 77.71% accuracy. EDRM-MLP achieves 74.26% accuracy with 1008.3 tokens (best token efficiency).

Dataset	Baselines						Ours							
	Direct		Standard		CoT		Token-Sign		EDRM-Inst-E		EDRM-Inst-C		EDRM-MLP	
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$ (%)	$\overline{\text{Tok}}$	Acc(%)	Tok
ARC-C	69.11	2.9	80.89	1007.1	83.87	897.1	83.87	961.1	85.49	892.3	85.92	919.3	81.48	649.6
ARC-E	81.27	3.0	88.72	739.9	88.47	702.4	88.47	766.4	89.65	668.1	89.91	693.2	87.96	517.3
BBH	40.41	3.4	72.81	2550.8	71.31	2219.3	71.31	2283.3	69.62	1661.8	71.03	1761.7	73.67	1678.3
CSQA	57.90	3.0	65.60	1151.6	66.83	1010.8	66.83	1074.8	72.89	958.7	73.10	968.6	72.97	964.6
CH-Abd	30.88	5.2	56.83	1915.5	58.33	1780.4	58.33	1844.4	60.25	1549.1	62.09	1633.1	62.67	1581.0
CH-Ded	56.29	4.8	65.87	924.3	67.79	885.4	67.79	949.4	73.71	809.2	74.97	851.0	74.96	795.4
FOLIO	52.82	2.7	73.09	1730.3	73.01	1791.6	73.01	1855.6	82.64	1727.6	82.64	1728.8	77.91	1495.3
GPQA	33.71	4.2	47.77	4964.9	52.90	4650.8	52.90	4714.8	58.48	4511.1	59.24	4673.9	55.13	3013.1
GSM8K	21.38	4.2	90.22	842.4	91.05	870.5	91.05	934.5	89.69	837.8	90.26	849.5	90.60	833.8
LSAT	40.73	2.6	52.23	2877.9	52.43	2850.9	52.43	2914.9	61.45	2726.6	62.61	2803.1	58.97	2268.8
MultiArith	58.67	2.8	97.67	354.6	97.17	418.0	97.17	482.0	98.33	390.3	98.35	390.9	97.83	379.4
MuSR	51.59	3.2	54.89	2052.6	53.17	2440.1	51.59	67.2	63.36	1599.2	66.50	1756.7	63.10	1071.8
PIQA	71.33	3.0	77.80	964.4	77.86	877.6	77.86	941.6	81.72	628.1	82.49	674.6	81.39	616.6
SIQA	55.12	3.0	62.74	943.7	63.61	856.6	63.61	920.6	71.08	790.0	71.83	824.7	66.53	588.2
StratQA	56.68	3.0	61.54	820.3	62.07	744.5	61.97	808.5	67.47	553.0	68.42	590.5	66.55	453.8
Overall	53.28	3.5	70.20	1389.0	70.83	1314.2	70.76	1298.4	74.78	1139.4	**77.71**	1182.1	74.26	<u>1008.3</u>

EDRM-Instance with Qwen3.5-35B-A3B. Table 19 presents the instance-level routing performance on `Qwen3.5-35B-A3B`. EDRM-Inst-C achieves 86.81% accuracy. EDRM-MLP achieves 86.00% accuracy with 1423.1 tokens.

Table 19: EDRM instance-level Performance on `Qwen3.5-35B-A3B`. EDRM-Inst-C achieves 86.81% accuracy. EDRM-MLP achieves 86.00% accuracy with 1423.1 tokens.

Dataset	Baselines						Ours							
	Direct		Standard		CoT		Token-Sign		EDRM-Inst-E		EDRM-Inst-C		EDRM-MLP	
	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	Acc(%)	Tok	$\overline{\text{Acc}}$ (%)	$\overline{\text{Tok}}$	Acc(%)	Tok
ARC-C	94.62	6.2	95.56	583.6	95.65	630.0	95.65	694.0	95.48	405.3	95.48	399.7	95.65	520.8
ARC-E	94.87	6.3	94.91	507.3	95.03	729.7	95.03	793.7	95.16	355.1	95.14	348.7	95.29	519.5
BBH	71.22	7.1	91.80	1691.9	92.29	1119.0	92.29	1183.0	91.27	1405.8	91.25	1404.6	91.27	1070.5
CSQA	84.11	6.3	85.75	614.7	85.75	1120.7	85.75	1184.7	85.50	368.2	85.47	360.3	85.83	484.4
CH-Abd	45.33	8.1	62.29	4404.0	61.04	2087.6	61.04	2151.6	55.29	1170.4	55.23	1164.9	63.04	3234.0
CH-Ded	68.04	7.5	87.96	2623.5	87.67	2311.3	87.67	2375.3	87.54	1984.1	87.39	1976.8	88.62	2224.2
FOLIO	68.19	6.0	81.06	3237.0	81.15	3045.3	81.15	3109.3	84.72	3079.9	84.69	3078.1	84.39	2616.3
GPQA	60.94	6.3	77.68	6410.3	75.67	6294.9	75.67	6358.9	80.80	5125.2	80.64	5080.8	80.80	5345.9
GSM8K	42.76	6.6	93.56	491.0	93.56	708.6	93.56	772.6	88.10	413.1	87.90	411.9	92.65	626.9
LSAT	79.09	6.0	93.95	2177.8	92.86	1764.4	92.86	1828.4	94.45	1673.5	94.34	1654.7	93.86	1659.5
MultiArith	95.33	5.7	98.67	153.5	98.67	233.1	98.67	297.1	99.17	173.6	99.17	173.6	99.17	262.3
MuSR	58.99	6.1	66.67	3561.1	64.42	2441.7	64.42	2505.7	67.46	2437.7	67.21	2375.6	65.08	1690.6
PIQA	92.44	6.1	93.85	693.9	94.34	1119.8	94.34	1183.8	93.63	458.5	93.61	456.8	94.12	522.2
SIQA	77.99	6.0	79.63	870.1	80.60	1749.3	80.60	1813.3	80.60	670.3	80.60	664.5	82.45	851.4
StratQA	74.32	6.0	63.10	2321.5	62.66	2576.5	74.32	70.0	88.03	2112.2	87.99	2104.2	82.62	1291.6
Overall	73.66	6.6	83.52	1895.2	83.34	1703.4	84.50	1512.2	85.25	<u>1276.1</u>	**86.81**	1380.2	86.00	1423.1

Summary of cross-model instance-level performance. Across all seven models, instance-level EDRM variants achieve meaningful accuracy–efficiency trade-offs. On base models (Llama-3.2-3B, Llama-3.1-8B, Qwen2.5-7B, Qwen2.5-32B), EDRM-Inst-E improves accuracy over the corresponding Global-E routing while keeping token consumption well below CoT. On reasoning models (Qwen3-4B, DeepSeek-R1, Qwen3.5-35B), EDRM-MLP delivers the largest token savings, and the calibratable Inst-C variant (which uses ground truth and thus serves as an Oracle-style upper bound) achieves the highest accuracy, demonstrating robust routing even when models are biased toward verbose deliberation.

B.6 Ablation Study on EDRM-MLP Variants

We conduct comprehensive ablation studies on **EDRM-MLP** across two design dimensions: (i) *input representation* (3D statistical descriptors, 64D entropy trajectory, or 67D hybrid concatenation), and (ii) *label strategy* (original multi-label vs. priority-constrained single-label). Table 20 reports average accuracy and token consumption across 15 reasoning benchmarks under four complementary evaluation settings that jointly disentangle the effect of the Direct Boost fallback and of the training/test split: (i) **w/FB (full)**: routing with Direct Boost fallback, evaluated on the full benchmark sets—the setting used for the EDRM-MLP column of Table 1; (ii) **w/o-FB (full)**: pure MLP routing decisions without fallback, fully deployable, on the full sets; (iii)/(iv) the same two routing variants evaluated on the **held-out 90% test set** (MLP trained on the 10% stratified split, tested on the remaining 90%), which isolates generalization from any tuning-set leakage.

Table 20: EDRM-MLP variants comparison (average accuracy and token consumption across 15 benchmarks) under four settings: with/without Direct Boost fallback (w/FB, w/o-FB) × full-set vs. held-out 90% test evaluation. Each cell reports *Acc* / *Token*. Best accuracy per model–setting is **bolded**; lowest token count is underlined. The w/FB (full) rows correspond to the EDRM-MLP column in Table 1.

Model	Setting	ML-3D		ML-64D		ML-67D		SL-3D		SL-64D		SL-67D	
		Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok
Llama-3.2-3B	w/FB (full)	63.72	170.2	64.91	180.8	64.62	176.0	**65.51**	194.3	63.26	148.4	63.33	149.7
	w/o-FB (full)	59.04	167.7	**59.74**	178.0	59.28	173.3	59.54	191.4	59.15	146.3	59.09	147.6
	w/FB (90%)	63.65	170.2	64.37	181.4	64.04	176.6	**65.44**	194.2	62.22	148.3	62.29	149.8
	w/o-FB (90%)	58.95	167.7	58.89	178.6	58.38	173.8	**59.43**	191.3	57.92	146.1	57.85	147.7
Llama-3.1-8B	w/FB (full)	70.36	123.7	69.14	119.9	69.07	112.1	**71.41**	148.0	67.89	106.6	68.81	114.4
	w/o-FB (full)	65.67	119.4	65.16	115.8	65.19	108.1	**65.71**	143.3	63.96	103.0	64.55	110.6
	w/FB (90%)	70.36	123.4	68.90	119.4	68.79	112.3	**71.40**	147.9	67.08	106.1	68.07	114.4
	w/o-FB (90%)	65.64	119.1	64.80	115.4	64.76	108.4	**65.71**	143.2	63.00	102.5	63.62	110.6
Qwen2.5-7B	w/FB (full)	74.01	188.9	**75.35**	224.3	74.57	191.6	73.35	163.2	74.03	167.6	73.36	159.4
	w/o-FB (full)	71.07	187.2	**71.81**	222.2	71.42	189.7	70.52	161.6	71.35	166.1	70.80	158.0
	w/FB (90%)	74.02	189.1	**75.18**	224.0	74.47	191.7	73.34	163.3	73.21	166.8	72.49	158.3
	w/o-FB (90%)	71.07	187.4	**71.52**	222.0	71.24	189.7	70.47	161.7	70.48	165.3	69.89	156.9
Avg.	w/FB (full)	69.36	160.9	69.80	175.0	69.42	159.9	**70.09**	168.5	68.39	140.9	68.50	141.1
	w/o-FB (full)	65.26	158.1	**65.57**	172.0	65.30	157.0	65.26	165.5	64.82	138.5	64.81	138.7
	w/FB (90%)	69.34	160.9	69.48	174.9	69.10	160.2	**70.06**	168.4	67.50	140.4	67.62	140.8
	w/o-FB (90%)	**65.22**	158.1	65.07	172.0	64.79	157.3	65.20	165.4	63.80	138.0	63.79	138.4

Key findings.

- **Input dimensionality exhibits model-dependent scaling.** For mid-scale models (Qwen2.5-7B), richer representations consistently outperform compact descriptors: ML-64D achieves the highest accuracy (75.35), exceeding ML-3D by +1.34%. Conversely, for Llama-3.1-8B and Llama-3.2-3B, the lightweight 3D input yields the best result (SL-3D: 71.41 and 65.51 respectively), suggesting that stronger backbones can extract sufficient routing signals from statistical summaries alone. This reveals a *capacity-efficiency trade-off*: trajectory-level features benefit capacity-limited routers, while concise descriptors suffice for larger models.
- **Label strategy interacts with model scale and input form.** The priority-constrained single-label formulation (SL) yields more stable gains on Llama-3.2-3B and Llama-3.1-8B, where simplifying the prediction target reduces optimization ambiguity. However, on Qwen2.5-7B with high-dimensional inputs (64D/67D), the original multi-label target (ML) slightly outperforms SL, implying that larger routers can effectively leverage richer supervision without confusion from label correlations.

- **Hybrid 67D features do not universally dominate.** While 67D occasionally differs from 64D (e.g., Qwen2.5-7B ML-67D: 74.57 vs. ML-64D: 75.35), it does not consistently improve and underperforms in several configurations (e.g., Llama-3.1-8B SL-67D: 68.81 vs. SL-3D: 71.41). We hypothesize that naively concatenating statistical and trajectory features may introduce redundancy or optimization interference for certain model-label combinations.
- **Token efficiency favors compact representations.** Across all models and settings, SL-64D/SL-67D achieve the lowest average token consumption (138.0–141.1), while ML-64D incurs the highest overhead (172.0–175.0). This suggests that single-label training encourages more decisive routing behavior, reducing unnecessary fallback to expensive decoding regimes.
- **Full-set and held-out evaluation agree closely, but fallback dominates the gap.** Comparing the 90% test rows against their full-set counterparts, accuracy drops by at most 1.24% (Llama-3.2 SL-67D w/o-FB: 59.09 vs. 57.85) and typically by 0.1–0.5%, confirming that the 10% stratified tuning split does not inflate the reported MLP results. In contrast, the w/FB vs. w/o-FB gap is large and systematic (+2.6 to +6.0% on every model–variant pair, at only 1.4–4.7 extra tokens), reconfirming that the Direct Boost fallback—not the routing rule itself—accounts for most of the accuracy headroom.

Practical recommendation. Considering both accuracy and computational efficiency, we recommend **SL-3D** as the default configuration for resource-constrained deployments: it achieves the best average accuracy (70.09 w/FB; 65.26 w/o-FB) with moderate token overhead (168.5/165.5). For scenarios prioritizing absolute accuracy with mid-scale models (e.g., 7B class), **ML-64D** provides the highest accuracy (75.35) at the cost of $\sim$34% additional tokens.

Limitations and future directions. Our ablation focuses on fixed-length (N=64) entropy trajectories and static feature concatenation. Future extensions could explore: (i) adaptive trajectory length selection based on instance difficulty; (ii) learnable feature fusion mechanisms (e.g., cross-attention between 3D and 64D branches); (iii) joint optimization of router and decoder to enable end-to-end adaptation of entropy dynamics.

ML-64D on reasoning models: inference-time calibration. For the two reasoning models added in the rebuttal stage, DeepSeek-R1-Distill-Qwen-7B and Qwen3.5-35B-A3B, the MLP router is trained with exactly the same ML-64D recipe as above (same architecture, 10% stratified split, multi-label BCE targets). The difference lies entirely in *inference-time* decision rules, which we summarize in Table 21 and describe here. First, both models apply a **probe-cost guard**: instances whose *Standard* decode costs fewer than N=64 tokens are forced to *Standard*, since probing them cannot pay for itself. Second, for DeepSeek-R1-Distill-Qwen-7B we apply a **Direct margin calibration**: reasoning-distilled models have a weakened Direct branch, and the raw three-way argmax frequently misroutes to *Direct*. We therefore require the Direct logit to exceed $\max(\text{logit}_{\textit{Standard}}, \text{logit}_{\textit{CoT}})$ by a margin β before *Direct* is adopted; otherwise the instance falls back to the better of *Standard*/*CoT*. We swept $\beta \in [0, 5]$ without retraining and selected $\beta = 0.30$, which maximizes routed accuracy while suppressing spurious Direct selections. Qwen3.5-35B-A3B retains $\beta = 0$ (raw argmax) because its Direct branch is strong enough not to require suppression. The w/o-FB rows report pure routing accuracy under these calibrated rules; the w/FB rows additionally apply the Direct Boost fallback shared with all other models.

Table 21: ML-64D router on the two reasoning models (average across 15 benchmarks), under the same four settings as Table 20. R1-Distill-Qwen-7B applies Direct margin calibration with β=0.30 at inference; Qwen3.5-35B-A3B uses raw argmax (β=0); both apply the probe-cost guard and Direct Boost fallback (w/FB rows). The w/FB (full) values correspond to the EDRM-MLP column in Table 1. Each cell reports *Acc* / *Token*.

Setting	DeepSeek-R1-Distill-Qwen-7B	Qwen3.5-35B-A3B
w/FB (full)	74.26 / 1008.3	**86.00** / 1423.1
w/o-FB (full)	68.73 / 1005.7	82.85 / <u>1418.3</u>
w/FB (90%)	73.63 / 1009.7	85.84 / 1429.9
w/o-FB (90%)	67.71 / <u>1007.2</u>	82.60 / 1425.0
CoT (reference)	70.83 / 1314.2	83.34 / 1703.4

Two observations follow. (i) **Calibration, not capacity, is the bottleneck on distilled models:** with fallback, the calibrated router reaches 74.26% (+3.43% over CoT) at 23.3% token savings; without fallback, the same router attains only 68.73% (−2.10% vs. CoT), showing that the margin rule alone cannot recover the weakened Direct branch of distilled models—fallback remains essential. (ii) **The strong reasoner needs no calibration:** on Qwen3.5-35B-A3B, raw argmax with fallback yields 86.00% (+2.66% over CoT) at 16.5% savings, and even the fully deployable w/o-FB variant (82.85%, 1418.3 tokens) sits within 0.5% of CoT at 16.7% lower cost, indicating that for models with a strong Direct branch the learned router generalizes without any inference-time adjustment.

B.7 Hyper-parameter sensitivity k for $V_{\text{SP}}/a_{\text{VNR}}$ control

The following experiments are all on EDRM-Global-E.

Evaluation protocol: population-level statistics (no sampling). In the standard Global-E deployment (§4), the routing thresholds are calibrated on a small random subset of 50 samples per benchmark (1%), and all reported Global-E numbers are averaged over 8 random seeds—this reflects the practical constraint that one cannot probe the full benchmark set at deployment time. For this ablation, we deliberately *remove* this source of sampling variation to isolate the effect of the hyperparameter k itself. Instead of estimating the feature means $\bar{S}_H, \bar{V}_{\text{sp}}, \bar{a}_{\text{vnr}}$ from a 50-sample calibration subset, we compute them over the **entire benchmark set** (population statistics), so the only quantity that changes across the table is k. Each reported cell is therefore a single deterministic evaluation rather than an 8-seed average. This is a controlled-attribution device, not a deployable configuration: it ensures that any accuracy or token difference between two columns is attributable to k alone, free of calibration-subset noise. Consequently, the numbers in Table 22 are not directly comparable to the 8-seed Global-E entries of Table 1 (e.g., Llama-3.2-3B at k=0.07: 59.79/107.7 here vs. 59.87/113.2 in the main table, where the 50-sample calibration and seed averaging shift both routing decisions and token counts).

Table 22: Routing accuracy and token consumption with different k values (weighted mean across 15 benchmarks). **Protocol:** population-level feature statistics computed over the full benchmark sets (no 50-sample calibration, single deterministic run per cell), so that differences across columns are attributable to k alone; see the protocol paragraph above and §B.14 for the deployable 8-seed Global-E protocol used in the main tables. Each k setting reports *Accuracy* / *Avg. Tokens*. Best accuracy per model is **bolded**; lowest token count is underlined.

Model	k=0.10		k=0.07		k=0.05		k=0.03	
	Acc	Token	Acc	Token	Acc	Token	Acc	Token
Llama-3.2-3B	59.99	110.1	59.79	<u>107.7</u>	59.79	<u>107.7</u>	**60.08**	114.6
Llama-3.1-8B	66.58	<u>123.9</u>	**66.98**	137.7	**66.98**	164.3	66.93	178.6
Qwen2.5-7B	71.55	<u>183.8</u>	**71.75**	194.0	71.63	196.6	71.63	196.6
Qwen3-4B-T	76.02	<u>351.4</u>	**76.06**	352.4	**76.06**	353.0	76.04	355.9
Avg.	68.54	<u>192.3</u>	**68.65**	197.9	68.62	205.4	68.67	211.4

Table 22 shows the trade-off between routing accuracy and token efficiency across four k thresholds. Two key patterns emerge:

(1) Accuracy is robust across k. Across all models, routing accuracy varies by at most 0.4% across $k \in \{0.03, 0.05, 0.07, 0.10\}$ (avg. 68.54–68.67), demonstrating that EDRM's routing performance is not sensitive to this hyperparameter. Moderate values (k=0.07) achieve the best average accuracy (68.65) with near-optimal token consumption.

(2) Token savings scale monotonically with k. Larger k values trigger routing more aggressively, resulting in lower token consumption (avg. 192.3 tokens at k=0.10 vs. 211.4 at k=0.03). However, the accuracy gain from k=0.10 → 0.07 (+0.11%) and the stable accuracy across all k values justify our default choice of k=0.07 as a balanced operating point.

(3) Robustness holds under the deployable protocol. Because the population-level protocol above removes sampling noise by construction, one may ask whether the k-robustness conclusion survives the 50-sample calibration used in deployment. The main-table Global-E entries (Table 1)—computed with 1% calibration subsets and averaged over 8 seeds—differ from the corresponding cells here by at most 0.29% in accuracy across the four models (e.g., Llama-3.2-3B: 59.79 vs. 59.87; Qwen2.5-7B: 71.75 vs. 71.66), confirming that calibration-subset noise is a second-order effect relative to the k-insensitivity reported here. Token counts shift more visibly (e.g., Llama-3.2-3B: 107.7 vs. 113.2) because different routing decisions on borderline benchmarks redistribute work between Direct and Standard/CoT, but the qualitative conclusion is unchanged: k controls the accuracy–cost operating point smoothly, and no choice of $k \in [0.03, 0.10]$ destabilizes routing.

B.8 Change-Point Detection Results

This section provides the complete change-point detection results across all 4 models and 3 routing groups (12 model–group pairs), as referenced in §3.2.

Method details. We apply two change-point detection algorithms to each sample's first-64-token entropy sequence: (1) **CUSUM** (cumulative sum): a sequential analysis technique that detects shifts in the mean of a sequence by monitoring cumulative deviations from the overall mean. We use a two-sided CUSUM with a detection threshold set at the 95th percentile of the permutation null distribution. (2) **BinSeg** (binary segmentation): a recursive algorithm that exhaustively searches for the single change point maximizing the between-segment variance reduction, then reports the optimal split location. For each model, we partition all evaluation samples into three groups based on their Global-E routing decisions (Direct, Standard, CoT) and analyze change-point distributions within each group. To assess statistical significance, we perform permutation tests (B=1000) comparing the between-group variance explained by detected change points against that of randomly assigned split points.

Evaluation protocol: population-level routing, label-free. The routing groups used here are assigned by the same *population-level* (no-sampling) protocol as the k-sensitivity analysis (Appendix B.7): each sample is routed once, deterministically, from its full-set feature statistics, rather than from a 1% calibration subset averaged over 8 seeds as in the main-table Global-E. This choice is deliberate and has two consequences. First, the analysis is *entirely label-free*: routing decisions depend only on the entropy-derived features $(S_H, V_{\text{sp}}, a_{\text{vnr}})$, and change-point detection reads only the raw 64-token entropy sequences—correctness labels are never consulted, so the result is independent of any benchmark label correction. Second, group memberships on borderline benchmarks may differ slightly from the 8-seed Global-E routing (cf. the token-count discussion in Appendix B.7); we use the deterministic partition here because a fixed, reproducible group assignment is a controlled-attribution device for testing *where* in the entropy sequence the regime shift occurs, not a deployable routing configuration.

Results summary. Table 23 presents the complete results. Key findings: (1) BinSeg-detected change points explain 2.55–4.38× more between-group variance than random split points (all $p \ll 0.001$). (2) Change-point location correlates strongly with routing semantics: CoT-routed samples exhibit the earliest change points (BinSeg median 9–29 tokens), Direct-routed samples the latest (median 33–38 tokens) or comparable to Standard-routed samples, which lie between the two extremes on three of four models. (3) Pre/post change-point entropy means differ significantly in 11/12 model–group pairs (t-test, $p < 10^{-5}$; exception: Qwen2.5-7B-Standard, $p = 0.15$).

B.9 Probe Length Ablation

This section presents the full probe length ablation across $N \in \{16, 32, 48, 64, 80, 96\}$, as referenced in §3.3 and §4.1.

Analysis. Accuracy enters a plateau region at $N \geq 48$: w/o-Fallback accuracy varies by at most 1.2% and Oracle by at most 0.9% within $N \in [48, 96]$. N=64 sits at the center of this plateau, providing a robust operating point. Token consumption is non-monotonic in N: longer probes cause the router to more confidently select low-cost modes, partially offsetting the probing overhead. The

Table 23: Complete change-point detection results (12 model–group pairs). Groups are assigned by the deterministic population-level routing protocol (label-free; see protocol note above), not the 8-seed Global-E calibration of the main tables. Permutation ratio > 1 indicates detected change points are more informative than random; all $p \ll 0.001$.

Model	**Group**	n	**CUSUM det.**	**BinSeg med.**	**Δ Entropy**	**t-test p**	**Perm. ratio**
Llama-3.2-3B	Direct	10813	21.8%	33	-0.390	$< 10^{-5}$	2.99×
	Standard	5210	9.2%	33	$+0.140$	$< 10^{-5}$	3.37×
	CoT	6808	42.2%	24	$+0.480$	$< 10^{-5}$	2.80×
Llama-3.1-8B	Direct	4691	18.3%	36	-0.268	$< 10^{-5}$	3.45×
	Standard	4307	16.2%	20	$+0.247$	$< 10^{-5}$	4.38×
	CoT	11773	30.9%	9	$+0.594$	$< 10^{-5}$	2.71×
Qwen2.5-7B	Direct	8626	23.3%	33	-0.479	$< 10^{-5}$	2.78×
	Standard	3974	6.3%	36	$+0.008$	0.15	3.24×
	CoT	8171	35.9%	28	$+0.475$	$< 10^{-5}$	2.65×
Qwen3-4B-T	Direct	10122	30.4%	38	-0.286	$< 10^{-5}$	2.55×
	Standard	3435	8.4%	38	-0.015	$< 10^{-5}$	3.72×
	CoT	7214	25.2%	29	$+0.222$	$< 10^{-5}$	2.62×

Table 24: Probe length ablation: Instance-level w/o-Fallback / Oracle accuracy (%) and average tokens across 4 models. H_{th} values are adaptively calibrated per N (e.g., Llama-3.2-3B: 30.0 at N=64, vs. the bin-searched 32 used in the main-table Inst-E), so the N=64 row is an independently calibrated run and its values (58.87/63.54/175.5) differ slightly from Table 4 (58.87/63.63/179.4, token includes amortized probe overhead).

N	**Llama-3.2-3B**			**Llama-3.1-8B**			**Qwen2.5-7B**			**Qwen3-4B-T**		
	w/oFB	Oracle	Tok	w/oFB	Oracle	Tok	w/oFB	Oracle	Tok	w/oFB	Oracle	Tok
16	51.97	55.33	140.8	60.59	65.85	190.9	70.21	76.01	291.4	75.66	78.66	496.0
32	56.76	61.56	173.4	63.14	68.72	170.9	69.91	74.60	240.4	76.86	79.29	458.4
48	58.59	63.12	175.6	64.29	69.35	152.5	70.92	75.34	237.2	76.39	78.34	404.3
64	58.87	63.54	175.5	65.11	70.01	149.5	71.60	75.99	239.6	76.91	78.65	400.6
80	58.70	63.46	172.5	64.98	69.85	144.4	71.74	76.18	241.4	76.61	78.26	388.1
96	58.69	63.50	170.8	64.88	69.71	138.3	71.69	76.18	241.7	76.64	78.24	381.8

H_{th} threshold adapts to N via our calibration procedure (e.g., Llama-3.2: H_{th} = 6.0/30.0/43.0 for N = 16/64/96), confirming that the cumulative entropy threshold correctly scales with probe length.

B.10 Deployable Selector: Full Results and Information-Theoretic Bounds

This section provides the complete selector results referenced in §4.1. As discussed in the main text, Fallback Compensation (Eq. 15) is an Oracle ceiling requiring ground truth and is not deployable. We exhaustively evaluated 6 families of deployable selectors that operate only on the disagreement subset (where the routed answer differs from the Direct answer) without test labels. Table 25 summarizes all schemes.

Design space. We explored three signal families: (i) **Posterior signals** (reusing decoding logs): PAE (tail-entropy comparison), PSC (tail-surprisal comparison), DCC (Direct min-probability threshold via leave-one-benchmark-out); (ii) **Task priors**: 1A (D-ratio prior, label-free), 1B (global-accuracy prior); (iii) **Per-benchmark calibration**: the per-bench d_{mp} selector (5-fold CV on 50 calibration samples per benchmark, no test labels), which is the best-performing scheme and the one reported in the main-text Table 4 (Selected-FB column).

Table 25: Selector comparison on EDRM-Inst-E (Acc% / Tok). Token counts include amortized probe overhead. All selector variants share the same token cost as Oracle, since the Direct fallback branch is determined by routing, not by the answer-selection logic.

Scheme	Llama-3.2-3B	Llama-3.1-8B	Qwen2.5-7B	Qwen3-4B-T
w/o-FB (baseline)	58.87 / 177.1	65.11 / 149.8	71.63 / 240.7	76.81 / 397.9
Global LOO d_{mp}	59.17 / 179.4	65.31 / 153.9	70.93 / 242.7	76.81 / 401.1
Per-bench d_{mp}	**59.59 / 179.4**	**65.44 / 153.9**	**71.77 / 242.7**	**76.71 / 401.1**
Oracle (w/ FB)	63.63 / 179.4	70.04 / 153.9	76.07 / 242.7	78.54 / 401.1

Exhaustive scheme exploration. Beyond the two d_{mp}-based selectors in Table 25, we evaluated four additional signal families: (i) **posterior signals**—PAE (tail-entropy comparison), PSC (tail-surprisal comparison), both reusing decoding logs; (ii) **task priors**—1A (D-ratio prior, label-free), 1B (global-accuracy prior). None of these achieved consistent positive gap recovery: PAE/PSC degrade Qwen3-4B by 6–10% (tail-entropy signals fail under over-reasoning), and 1A/1B generalize poorly across scales. The d_{mp} signal is the single best discriminator (AUC 0.55–0.72), but per-benchmark calibration is essential—the global LOO variant fails on Qwen2.5-7B (-15.6% gap recovery) due to cross-benchmark distribution shift.

Why most selectors fail. On the disagreement subset, Direct is structurally disadvantaged (D_d:$D_m \approx$ 1:3 to 1:8.6). Any selector that unconditionally switches to Direct incurs a net loss. The posterior signals (PAE, PSC) achieve AUC 0.55–0.72, far below the ≈ 0.85 needed for net-positive recovery under such base-rate imbalance.

Information-theoretic upper bound. Even with *oracle* per-benchmark threshold selection (i.e., selecting the optimal threshold with access to test labels), the recoverable gap is bounded by the mutual information in the available signals. Table 26 reports this bound: the best single signal (d_{mp}) recovers at most 0.07–0.89% out of a total Oracle gap of 1.73–4.93%, and combining all five signals via logistic regression raises the bound only marginally (at most 1.18%, still $\leq$25% of the gap). This confirms that the limited Oracle recovery is an inherent signal-information limit, not a methodological flaw.

B.11 Complete Bootstrap and Permutation Test Results

This section reports the full per-model confidence intervals and p-values underlying the statistical analysis in §4.6.

Table 26: Information-theoretic upper bound on selector recovery (% of Oracle gap recoverable with an oracle per-benchmark threshold selected with access to test labels, recomputed on the label-corrected evaluation data). "5-signal LR" combines d_{mp}, tail entropy, mean probability, answer agreement, and token length via per-benchmark logistic regression with an oracle probability threshold.

Signal	**Llama-3.2-3B**	**Llama-3.1-8B**	**Qwen2.5-7B**	**Qwen3-4B-T**
d_{mp} (best single)	+0.89	+0.50	+0.58	+0.07
5-signal LR	+1.18	+0.69	+0.70	+0.20
Oracle gap (total)	4.75	4.93	4.44	1.73

Table 27: Instance-level per-sample bootstrap: full 95% CIs and paired p-values (n=23061/model, 1000 resamples). Δ is relative to CoT unless noted.

Model	**Method**	**Acc(%)**	**CI lo**	**CI hi**	**Δ vs CoT**	p
Llama-3.2-3B	CoT	60.20	59.56	60.84	–	–
	Standard	59.89	59.25	60.51	−0.31	–
	w/o-FB	58.87	58.22	59.54	−1.32	0.001 (**)
	Oracle	63.63	63.00	64.25	+3.43	0.001 (**)
	Oracle vs w/o-FB	–	–	–	*+4.75*	*0.001 (**)*
Llama-3.1-8B	CoT	68.11	67.47	68.70	–	–
	Standard	66.08	65.45	66.70	−2.03	–
	w/o-FB	65.11	64.54	65.74	−3.00	0.001 (**)
	Oracle	70.04	69.48	70.61	+1.93	0.001 (**)
	Oracle vs w/o-FB	–	–	–	*+4.93*	*0.001 (**)*
Qwen2.5-7B	CoT	72.44	71.89	73.01	–	–
	Standard	71.43	70.89	71.99	−1.01	–
	w/o-FB	71.63	71.06	72.23	−0.81	0.001 (**)
	Oracle	76.07	75.55	76.63	+3.63	0.001 (**)
	Oracle vs w/o-FB	–	–	–	*+4.44*	*0.001 (**)*
Qwen3-4B-T	CoT	80.75	80.26	81.26	–	–
	Standard	79.90	79.39	80.44	−0.85	–
	w/o-FB	76.81	76.27	77.37	−3.94	0.001 (**)
	Oracle	78.54	78.01	79.07	−2.21	0.001 (**)
	Oracle vs w/o-FB	–	–	–	*+1.73*	*0.001 (**)*

Table 28: Global-level benchmark paired permutation tests (n=15, B=10000). Token savings are significant on all models ($p < 0.001^{***}$).

Model	Comparison	ΔAcc (%)	p	Tok saving	p
Llama-3.2-3B	w/o-FB vs CoT	−0.88	0.13 (ns)	29.6%	$< 0.001^{***}$
	w/o-FB vs Std	−0.73	0.27 (ns)		
	Oracle vs CoT	+3.86	0.005 (**)		
Llama-3.1-8B	w/o-FB vs CoT	−2.42	0.014 (*)	46.1%	$< 0.001^{***}$
	w/o-FB vs Std	−1.06	0.15 (ns)		
	Oracle vs CoT	+2.63	0.11 (ns)		
Qwen2.5-7B	w/o-FB vs CoT	−0.38	0.54 (ns)	27.1%	$< 0.001^{***}$
	w/o-FB vs Std	+0.13	0.77 (ns)		
	Oracle vs CoT	+4.04	0.014 (*)		
Qwen3-4B-T	w/o-FB vs CoT	−4.01	< 0.001 (***)	38.1%	$< 0.001^{***}$
	w/o-FB vs Std	−3.15	< 0.001 (***)		
	Oracle vs CoT	−2.12	0.025 (*)		

B.12 Token Amortization of the Global Probe

The global-level routing probe requires decoding 50 short probe sequences per benchmark. We amortize this overhead across all test samples of the benchmark:

$$\text{amortized}_j = \frac{N_{\text{probe}} \times \overline{\text{probe_tok}}_j}{n_j}, \tag{17}$$

where N_{probe}=50, $\overline{\text{probe_tok}}_j$ is the mean decode length of the 50 probe samples on benchmark j (capped at N=64), and n_j is the number of test samples on benchmark j.

Table 29: Amortized probe overhead per sample. The overhead is 1.46–2.08 tokens/sample ($< 1.8\%$ of total token cost), negligible in all settings. Instance-level routing embeds the probe in the predecode ($\min(64, T_{\text{std}})$), so no additional amortization is needed.

Model	Amortized	Min	Max	n total
Llama-3.2-3B	1.97	1.25	6.46	23061
Llama-3.1-8B	1.46	0.31	5.29	23061
Qwen2.5-7B	2.07	1.33	7.14	23061
Qwen3-4B-T	2.08	1.33	7.14	23061

The per-benchmark amortized cost ranges from 0.31 (large benchmarks such as ARC-Easy) to 7.14 tokens (small benchmarks such as GPQA with n=448). Even the worst case (7.14 tokens on GPQA) is under 3% of that benchmark's average CoT length. Instance-level routing does not incur this overhead, as its probe is the $\min(64, T_{\text{std}})$ predecode already counted in the Standard token cost.

API deployment path. When the API exposes top-K logprobs, the entropy trajectory can be approximated directly, enabling zero-overhead deployment. When logprobs are unavailable, inverse-logit estimation from the model's verbalized probability estimates can serve as a proxy. These paths extend EDRM to closed-source settings where raw logits are inaccessible.

B.13 TOKEN-SIGNATURE BASELINE SPECIFICATION

To ensure a fair comparison, the Token-Signature (TS) baseline receives the *same* 64-token probe signal and the *same* prompt as EDRM. The differences are confined to the routing logic:

- **Signal**: TS uses a single variable V_{sp} (the Spearman rank correlation of the entropy sequence), whereas EDRM uses three descriptors $(S_H, V_{\text{sp}}, a_{\text{vnr}})$.
- **Routing rule**: $V_{\text{sp}} < 0 \rightarrow$ CoT (entropy rising $\Rightarrow$ uncertain, needs reasoning); $V_{\text{sp}} > 0 \rightarrow$ Direct (entropy falling $\Rightarrow$ confident, skip reasoning); $V_{\text{sp}} \approx 0 \rightarrow$ Standard.
- **Token accounting**: identical to EDRM (probe tokens included).
- **Calibration**: TS uses the same per-benchmark 50-sample probe set to determine the neutral band around $V_{\text{sp}} = 0$.

This design isolates the effect of the feature set: TS is a univariate baseline against EDRM's multivariate $(S_H, V_{\text{sp}}, a_{\text{vnr}})$ routing. The consistent improvement of EDRM over TS (Table 1) thus reflects the value of the additional cumulative-entropy and volatility descriptors.

B.14 CALIBRATION PROTOCOL AND DATA PARTITIONING

To address concerns about data leakage, we clarify the three-role partitioning used across all EDRM variants. Table 30 summarizes the roles.

Table 30: Three-role data partitioning. Only the MLP router uses labels (for training); calibration and test are label-free w.r.t. routing decisions.

Role	Used for	Source	Labels?
Tuning (MLP only)	Train MLP router	10% stratified per benchmark	Yes
Calibration	Calibrate H_{th}, θ_{bench}	50 samples per benchmark	No
Test	Report all numbers	Remaining 90% (MLP) / all (heuristic)	Eval only

Calibration is label-free. The H_{th} threshold for the heuristic router (EDRM-Global-E, Inst-E) is calibrated using only entropy-dynamics features (S_H) from the 50 probe samples—it does not read answer correctness. Similarly, the per-benchmark d_{mp} selector threshold θ_{bench} is selected by 5-fold cross-validation on the 50 calibration samples using only the minimum token probability d_{mp}, without test labels.

MLP evaluation protocol. For EDRM-MLP, 10% stratified sampling per benchmark is used for training. All MLP numbers in the main text and appendix tables are evaluated on the full benchmark sets (all samples, including the tuning split); the difference between this full-set evaluation and the held-out 90% test set is $\leq 0.63\%$ across all 7 models (with direct boost; $\leq 1.02\%$ without fallback), confirming that the 10% tuning split does not inflate results. The full comparison under both evaluation settings and both fallback configurations is reported in Tables 20 and 21.