How Hard Does It Think? Analyzing Step-Aware Reasoning Energy in LLM Chain-of-Thought Trajectories

Anonymous ACL submission

Abstract

Understanding how computational effort is allocated across individual reasoning steps in chain-of-thought (CoT) trajectories is a key open challenge for LLM interpretability, yet existing methods either rely on output-level signals or collapse processing depth into a single trajectory-level scalar, leaving step-wise reasoning effort opaque. We propose **Step-Aware Reasoning Energy (SARE)**, a geometric framework that quantifies computational effort at the granularity of individual CoT steps via Centered Kernel Alignment (CKA) between Gram matrices of token hidden states across adjacent transformer layers. Unlike token-level or output-based proxies, SARE captures inter-token relational structure without requiring eigenvector alignment or cluster correspondence, and further contextualizes energy within the semantic progression of reasoning by modeling CoT trajectories as transitions among latent semantic states. Experiments across six reasoning benchmarks and three open-weight LLMs reveal three consistent findings: reasoning energy is highly non-uniform across semantic step types, exhibiting structured phase-like transitions invisible to trajectory-level metrics; incorrect trajectories are associated with systematically lower energy at critical reasoning junctions; and SARE-based features match or outperform output-based confidence baselines across most evaluated settings, demonstrating that internal geometric dynamics encode predictive information beyond surface-level signals.

1 Introduction

Chain-of-Thought (CoT) prompting (Wei et al., 2022) has become one of the most effective techniques for eliciting multi-step reasoning from large language models(LLMs), substantially improving performance on mathematical, logical, andcommonsense benchmarks. By generating explicit intermediate reasoning steps, CoT provides a window into the model's reasoning process. Yet despite this apparent transparency, we have little understanding of how computational effort is actually allocated across those steps (e.g., which steps demand deep internal processing, and which are resolved trivially), leaving the black-box nature of LLM reasoning largely intact.

Existing attempts to explain CoT reasoning operate primarily at the surface level. Output-based approaches examine token log-probabilities (Hwang et al., 2026) or train classifiers on trajectory text (Madaan et al., 2023), but treat the transformer's internal computations as opaque. Interpretability research has begun probing transformer internals (Belrose et al., 2023; Chuang et al., 2023), yet typically at the level of individual tokens or collapsed aggregate representations, which is too coarse to capture the computational dynamics of entire reasoning steps. Recently, Chen et al. (2026) propose the Deep Thinking Ratio (DTR) to measure reasoning effort via token-level layer-wise prediction stability, but DTR aggregates depth into a single trajectory-level scalar and treats tokens independently, discarding the relational structure among tokens within a step. Neither line of work adequately connects the model's layer-wise computation to the semantic progression of the reasoning chain.

To bridge this gap, we propose **Step-Aware Reasoning Energy (SARE)**, a geometric framework that quantifies computational effort at the granularity of individual CoT reasoning steps. For each step and each pair of adjacent transformer layers, we compute Centered Kernel Alignment (CKA) (Kornblith et al., 2019) between Gram matrices constructed from token hidden states, measuring how much the step's internal token relationship geometry reorganizes during the forward pass. A step that continues to reorganize across many layers reflects genuine computational effort; one that stabilizes early indicates minimal processing. Unlike token-level depth measures, CKA operates on pairwise token similarity structure, preserving inter-token re-

lational information without requiring eigenvector alignment or cluster correspondence across layers. We further contextualize SARE within the semantic trajectory of reasoning by modeling CoT steps as transitions among latent semantic states identified via unsupervised clustering, enabling joint analysis of how energy varies across semantic roles and how it evolves across the chain.

We evaluate SARE across six reasoning benchmarks spanning mathematical, commonsense, and multi-hop domains (i.e., GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), CSQA (Talmor et al., 2019), StrategyQA (Geva et al., 2021), HotpotQA (Yang et al., 2018), and MuSiQue (Trivedi et al., 2022)) using three open-weight LLMs: LLaMA-3.2-3B (Grattafiori et al., 2024), Phi-4-mini (Abouelenin et al., 2025), and Gemma-3-4B (Team et al., 2025). Our analysis addresses two core research questions:

- **RQ1:** Do different semantic reasoning states exhibit distinct step-level reasoning energy profiles, and do these profiles differ between correct and incorrect trajectories?
- **RQ2:** Can step-level energy dynamics predict reasoning failures without access to ground-truth labels?

Our results reveal three consistent findings. First, reasoning energy is highly non-uniform across semantic step types: early setup and final synthesis steps anchor the energy extremes while mid-trajectory factual retrieval steps exhibit stable, moderate energy, exposing a structured phase-like allocation of computational effort invisible to trajectory-level metrics. Second, incorrect trajectories are associated with lower reasoning energy at specific reasoning junctions, particularly in final verification and compositional reasoning states. Third, SARE, combined with token count as a complementary signal, match or outperform output-based confidence baselines including token log-probability, entropy, and perplexity on most evaluated benchmarks and models.

In summary, our primary contributions are:

- We propose SARE, a geometry-grounded framework that quantifies step-level reasoning effort in CoT trajectories via CKA on token hidden state Gram matrices, capturing inter-token relational dynamics that token-level measures discard.
- We establish that reasoning energy is structured and non-uniform across semantic step types, and that incorrect trajectories are consistently associated with lower energy at critical reasoning junctions.
- We show that SARE-based features are competitive with or superior to output-based confidence baselines for offline reasoning failure detection across diverse benchmarks, models, and reasoning domains.

2 Related Work

2.1 Understanding and Explaining Chain-of-Thought Reasoning

The emergence of CoT prompting (Wei et al., 2022) and its variants has sparked extensive research into how LLMs solve complex multi-step problems. While empirical results show that CoT substantially enhances reasoning performance, understanding the internal mechanism behind this capability is an active area of study. A significant portion of existing work focuses entirely on the final representations or the generated output layer. For instance, methods like Self-Consistency (Wang et al., 2022) sample multiple reasoning paths and aggregate semantic clusters over the final answers to increase reliability. Parallel efforts have analyzed reasoning errors by examining token log-probabilities (Kauf et al., 2024) or by classifying reasoning types directly from the textual output trajectory (Madaan et al., 2023). Although these approaches offer practical ways to evaluate the consistency of a reasoning path, they operate entirely on the surface level, leaving the step-to-step computational effort within the deep layer representations of the model unexplored. As a result, surface-level methods fall short in pinpointing the exact internal location where an error begins.

2.2 Internal Representation Analysis in Transformers

Analyses of transformer internal mechanics have traditionally investigated how structural information is built progressively across layers. Tools such as "tuned lenses" (Belrose et al., 2023) and early exiting strategies (Schwartz et al., 2020; Teerapittayanon et al., 2016) attempt to map hidden states to final vocabulary predictions or determine when computation can be terminated. Recent studies, such as DoLa (Chuang et al., 2023), focus on decoding strategies that optimize information difference

between specific intermediate layers to mitigate hallucinations. However, these methods primarily focus on trajectory-level representation or trace representations per-token across layers. A token-level lens often fails to capture the aggregate semantic value of a contiguous reasoning step (e.g., an equation or a logical deduction).

Our work bridges the gap between text-level CoT analysis and token-level layer probing. By defining **Step-Aware Reasoning Energy (SARE)**, we aggregate representational transitions across the specific span of tokens corresponding to a reasoning step. Furthermore, we cluster the final representations into distinct *reasoning states*, moving beyond raw token probabilities. This novel approach quantifies the computational effort of step transitions (measured via layer-wise alignment) and grounds it in the broader semantic topology of CoT reasoning (measured via state transitions based on the final layer). Consequently, our framework provides multiple dimensions: stepwise representation changes and semantic state evolution, to explain and evaluate CoT reasoning capabilities.

3 Preliminaries: Modeling Reasoning as State Transitions

We first formalize the CoT trajectory as a stochastic process to analyze the evolution of internal reasoning logic.

3.1 Step-Aware Formalization

A CoT reasoning trajectory $\mathcal{T}$ is segmented into a sequence of T discrete textual reasoning steps: $\mathcal{T} = [s_1, \ldots, s_T]$, where each step s_t ($t = 1, \ldots, T$) consists of n_t tokens. To capture the internal semantic relationships among tokens at a given depth, we define the Gram matrix $\boldsymbol{G}_t^{(l)} = \boldsymbol{H}_t^{(l)}(\boldsymbol{H}_t^{(l)})^\top$, where $\boldsymbol{H}_t^{(l)} \in \mathbb{R}^{n_t \times d}$ denotes the matrix of token hidden states for step s_t at layer l.

3.2 Semantic State Clustering

Following Yu et al. (2025), we interpret the progression of reasoning steps as transitions among latent semantic states. Each step s_t is represented by a spectral embedding derived from the eigenvalue spectrum of its cumulative token Gram matrix, computed from the last-layer hidden states of the LLM. These embeddings are then grouped via K-Means clustering to infer reasoning clusters (macro-states) $C \in \{C_1, \ldots, C_K\}$, each capturing a distinct conceptual function such as problem framing, intermediate verification, or factual retrieval. Each step is assigned a hard cluster label, and the resulting cluster sequence defines the trajectory's state sequence for downstream analysis. Full implementation details are provided in Appendix A.1.

3.3 Markovian Transition Framework

We model the sequence of reasoning clusters as a first-order Markov chain. The dynamics of the reasoning process are governed by a transition probability matrix P, where the probability of transitioning from cluster C_i to C_j is defined as:

$$P_{ij} = P(s_{t+1} = C_j \mid s_t = C_i) \tag{1}$$

This formulation allows us to track cross-step energy velocities $\Delta E(s_t \rightarrow s_{t+1})$ and analyze the stability of the reasoning trajectory through the lens of state-space transitions. In practice, these cross-step energy dynamics are operationalized in our downstream analysis through the *volatility*, *peaks*, and *valleys* statistics in the trajectory-level feature vector (Section 5.3), which collectively capture the magnitude and direction of energy changes between consecutive steps.

4 Quantifying Reasoning Energy via Geometric Dissimilarity

Having established the semantic structure of reasoning trajectories, we now turn to measuring the computational effort expended within each step.

4.1 Layer-to-Layer Dissimilarity via Centered Kernel Alignment (CKA)

A key insight from mechanistic interpretability is that transformer layers do not process all tokens equally: some tokens require sustained representational revision across many layers before their contextual role is resolved, while others stabilize early and require little further computation (Chuang et al., 2023; Chen et al., 2026). We operationalize this observation at the *step level*: a reasoning step that undergoes substantial reorganization of its internal token relationship geometry across consecutive layers is one the model actively "works on," reflecting genuine computational effort. Conversely, a step whose geometry stabilizes rapidly indicates that the model resolved it with minimal processing. Crucially, this geometric perspective operates on raw hidden states rather than

output-level signals, capturing computational effort that may never surface in the model's token predictions.

Formulation. Specifically, We apply CKA (Kornblith et al., 2019) to the centered Gram matrix $\tilde{\boldsymbol{G}}_t^{(l)} = \boldsymbol{M}_{n_t}\boldsymbol{G}_t^{(l)}\boldsymbol{M}_{n_t}$, where $\boldsymbol{M}_{n_t} = \boldsymbol{I}_{n_t} - \frac{1}{n_t}\mathbf{1}\mathbf{1}^\top$ is the standard centering matrix and n_t is the number of tokens in step s_t. $\boldsymbol{I}_{n_t} \in \mathbb{R}^{n_t \times n_t}$ denotes the identity matrix and $\mathbf{1} \in \mathbb{R}^{n_t}$ is an all-ones vector. The layer-wise dissimilarity score for step s_t between adjacent layers l and $l+1$ is then:

$$D_t^{(l)} = 1 - \mathrm{CKA}\big(\tilde{\boldsymbol{G}}_t^{(l)}, \tilde{\boldsymbol{G}}_t^{(l+1)}\big), \text{for } l = 1, \ldots, L-1 \tag{2}$$

where

$$\mathrm{CKA}\big(\tilde{\boldsymbol{G}}_t^{(l)}, \tilde{\boldsymbol{G}}_t^{(l+1)}\big) = \frac{\mathrm{tr}\big(\tilde{\boldsymbol{G}}_t^{(l)}\tilde{\boldsymbol{G}}_t^{(l+1)}\big)}{\big\|\tilde{\boldsymbol{G}}_t^{(l)}\big\|_F \big\|\tilde{\boldsymbol{G}}_t^{(l+1)}\big\|_F}. \tag{3}$$

$D_t^{(l)}$ is high when the pairwise token similarity structure reorganizes substantially between layers l and $l+1$, and low when it remains stable. The full depth profile $\boldsymbol{D}_t = [D_t^{(1)}, \ldots, D_t^{(L-1)}]$ traces the geometric evolution of step s_t throughout the entire forward pass.

Total Step-Aware Reasoning Energy We define the *Step-Aware Reasoning Energy* (E_t) as the total accumulated geometric dissimilarity across all layer transitions:

$$E_t = \sum_{l=1}^{L-1} D_t^{(l)}. \tag{4}$$

A high E_t indicates extensive cross-layer reorganization, reflecting that the model expended significant computational effort to refine the internal relational structure of that step. Conversely, a low E_t implies early structural stabilization, characteristic of trivial transitions or routine factual recall.

4.2 Why CKA over Alternative Measures.

Several natural alternatives exist for measuring layer-to-layer geometric change, each of which we argue is insufficient for our purpose.

First, one could directly compare the eigenvalue spectra of $G_t^{(l)}$ and $G_t^{(l+1)}$, inspired by Yu et al. (2025). However, the eigenvectors of each Gram matrix are derived independently at their respective layers, spanning different principal directions with no guaranteed cross-layer correspondence; comparing eigenvalues without aligning their eigenvectors is therefore geometrically meaningless.

Second, one could compute the Jensen-Shannon Divergence (JSD) between token-level next-token distributions across adjacent layers, as adopted in DoLa (Chuang et al., 2023) and DTR (Chen et al., 2026), and aggregate these scores to the step level by averaging. While this approach is computationally efficient, it treats each token independently and discards all token-token relationship information, precisely the relational structure that the Gram matrix is designed to capture. Since our goal is to measure how the *collective* semantic geometry of a reasoning step evolves across layers, a token-wise measure that ignores inter-token dependencies is fundamentally misaligned with this objective.

Third, one could compare soft cluster distributions $P(C_m \mid s_t)$ defined in Section 3.2 via JSD across layers. However, since clusters are derived independently at each layer, there is no guaranteed correspondence between cluster identities: the same cluster index may refer to entirely different semantic groupings at different layers. While this misalignment could in principle be resolved by permuting cluster assignments to find an optimal correspondence, doing so requires solving a combinatorial assignment problem at every layer transition, which is computationally prohibitive at scale.

CKA circumvents all three limitations by operating directly on the Gram matrices, whose (i, j) entries always refer to the same token pair across all layers by construction. This preserves the full pairwise token relationship structure without requiring any eigenvector alignment, cluster correspondence, or permutation. Furthermore, CKA is invariant to orthogonal transformations and isotropic scaling of hidden states, making it robust to the layer-wise normalization and rotation ambiguities that are common in deep networks.

5 Experiments

In this section, we empirically validate our unified framework. By integrating layer-wise geometric dissimilarity with Markovian state transitions, we analyze how reasoning energy varies across semantic roles, evolves over the course of a reasoning trajectory, and whether its patterns can indicate reasoning failure of CoT trajectories independently of the final answer.

5.1 Data and Models

We evaluate our framework on six established benchmarks across three reasoning domains, all

of which are widely used to assess LLM reasoning ability: (1) **Math:** *GSM8K* (Cobbe et al., 2021) and *MATH* (Hendrycks et al., 2021), focusing on grade-school and advanced numerical problem-solving. (2) **Commonsense:** *CSQA* (Talmor et al., 2019) and *StrategyQA* (Geva et al., 2021), challenging the model's intuitive reasoning and implicit factual deductions. (3) **Multi-Hop:** *HotpotQA* (Yang et al., 2018) and *MuSiQue* (Trivedi et al., 2022), requiring multi-step factual inference over textual evidence.

For the LLM backbones, we generate CoT trajectories and extract internal hidden states using three recent, highly-capable models: **LLaMA-3.2-3B** (Grattafiori et al., 2024), **Phi-4-mini** (Abouelenin et al., 2025), and **Gemma-3-4B** (Team et al., 2025).

For each dataset we sample 800 examples per model (687 for StrategyQA, where the full test set is used), drawing from the standard test split for GSM8K, MATH, and StrategyQA and the validation split for CSQA, HotpotQA, and MuSiQue. Full protocol details (i.e., decoding parameters, per-dataset answer normalization, and class balance statistics) are provided in Appendix A.3.

5.2 RQ1: Reasoning Energy Profiles Across Reasoning States

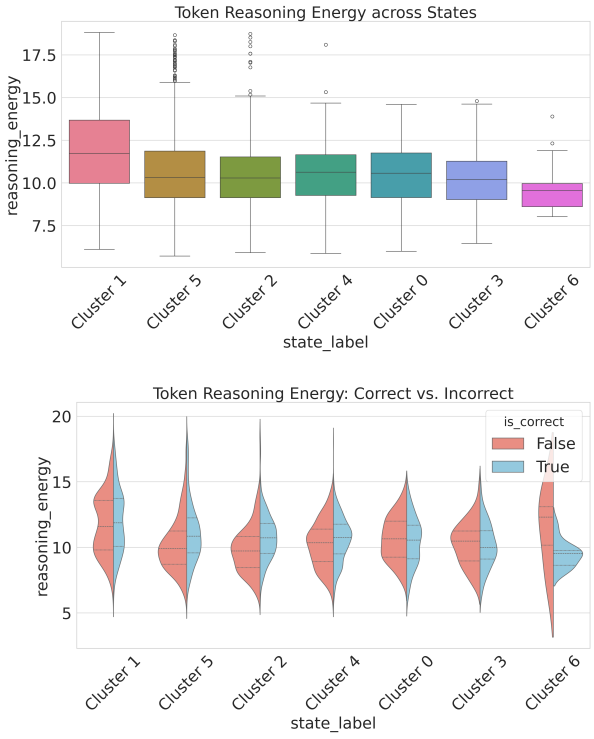

Figure 1: Reasoning energy profiles for each cluster on **GSM8K** with **Phi-4-mini**, shown unconditioned (left) and conditioned on trajectory correctness (right).

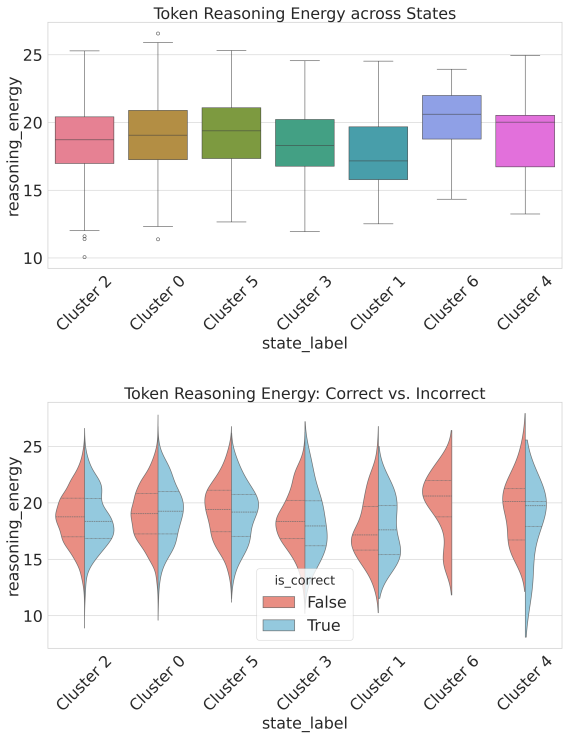

Figure 2: Reasoning energy profiles for each cluster on **HotpotQA** with **Gemma-3-4B**, shown unconditioned (left) and conditioned on trajectory correctness (right).

Our first goal is twofold: (1) to examine whether different reasoning states exhibit distinct step-aware reasoning energy (SARE) profiles, and (2) to investigate whether the energy profiles within each reasoning state differ between correct and incorrect trajectories.

To address these questions, we cluster all reasoning steps generated by the model into K groups ($K = 7$ in our experiments) using the method described in Section 3.2, and then compute the step-level reasoning energy defined in Section 4. To answer the first question, we visualize the SARE distribution of each cluster using *box plots*; to answer the second, we compare the distributions of correct and incorrect trajectories within each cluster using *violin plots*. Figures 1 to 3 present representative results for Phi-4 on GSM8K, Gemma-3 on HotpotQA and Llama-3.2 on StrategyQA, respectively. Clusters are arranged according to their most frequently observed positions in the reasoning trajectory.

Our analysis of the SARE profiles across three diverse cognitive domains reveals that internal hidden states are not merely abstract representations, but correspond to distinct functional stages of the reasoning process. By combining the quantitative

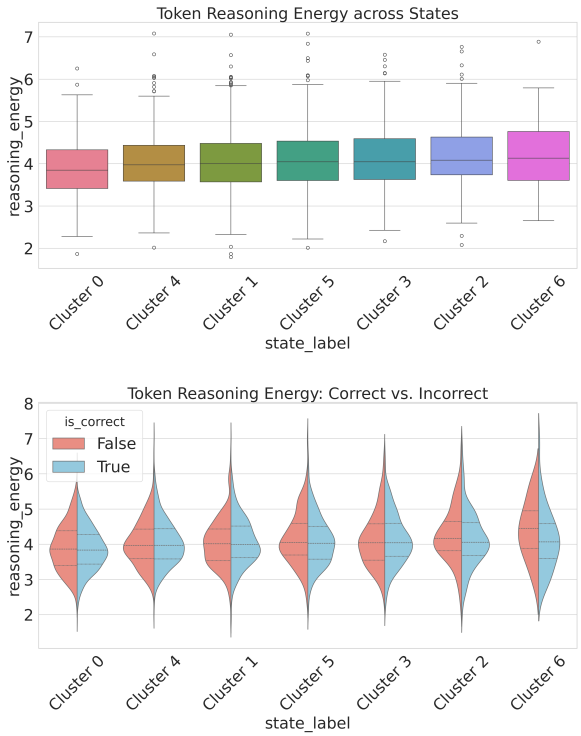

Figure 3: Reasoning energy profiles for each cluster on **StrategyQA** with **LLaMA-3.2-3B**, shown unconditioned (left) and conditioned on trajectory correctness (right).

distributions from box and violin plots with the qualitative examples from our curated clusters, we identify several key findings regarding the energetic cost of CoT generation.

Finding 1: Semantic Interpretation of Reasoning States A cross-model examination of the $K = 7$ clusters reveals a broadly consistent functional taxonomy of reasoning states. Early-stage clusters typically correspond to **initial setup and information extraction**, where the model parses the prompt and identifies key variables or entities (e.g., extracting quantities in math problems or identifying relevant entities in multi-hop questions). In *Phi-4-mini* on *GSM8K*, this stage exhibits relatively *high* median energy, suggesting that translating natural language into a structured mathematical representation requires substantial internal transformation. In contrast, for *LLaMA-3.2-3B* on *StrategyQA*, the corresponding early-stage clusters show comparatively *lower* energy than later reasoning states.

Mid-trajectory clusters are generally associated with **factual retrieval and intermediate comparison**, such as accessing world knowledge or performing local logical inferences. These states tend to exhibit stable, *moderate* energy levels across models. Final-stage clusters typically correspond to **synthesis and termination**, including integrative reasoning steps and final answer formulation. Notably, in *Gemma-3-4B* and *LLaMA-3.2-3B*, the final cluster often exhibits the *highest* energy, suggesting that the synthesis of disparate evidence is the most computationally demanding stage in multi-hop and commonsense reasoning. By contrast, in *Phi-4-mini*, the final cluster consistently shows the *lowest* energy, indicating that in mathematical reasoning, the last step is often comparatively lightweight once the core reasoning has already been completed.

Finding 2: Energy-Based Signatures of Reasoning Failure The violin plots reveal that incorrect trajectories are associated with distinctive energy patterns: they tend to exhibit lower energy than correct trajectories in specific reasoning states, most notably in final verification or synthesis states, where erroneous trajectories skew toward lower-energy distributions. For example, in Phi-4-mini on GSM8K, incorrect final reasoning steps show a clear shift toward lower SARE. We note that this is a correlational observation: the analysis does not establish whether lower energy precedes or contributes to failure, but rather that the two co-occur at specific reasoning junctions.

This divergence is not limited to the final stage. In LLaMA-3.2, the energy distributions of correct and incorrect trajectories largely overlap in early retrieval states, but begin to diverge in later quantification and compositional reasoning states. This suggests that the correlation between lower energy and incorrect outcomes is localized to specific junctions in the trajectory rather than being uniformly distributed.

Finding 3: Model-Specific Energetic Profiles We observe that the "dynamic range" of reasoning energy is heavily dependent on model architecture. *Gemma-3-4B* utilizes a much wider energy spectrum (10.0–27.5) than *Llama-3.2-3B* (2.0–7.0) or *Phi-4-mini* (6.0–18.0). This implies that larger or differently architectural models may engage in more intensive representational re-writing across transformer layers.

5.3 RQ2: Predicting Reasoning Failure via Latent Dynamics

To evaluate the predictive utility of our internal reasoning signals, we define a binary classification

task to detect reasoning errors (incorrect final answers) in an offline setting.

Experimental Design and Evaluation Protocol Following common practice in constructing trajectory features (Dempster et al., 2019; Wang et al., 2017), we aggregate step-aware energy and state information into a 12-dimensional trajectory-level vector consisting of seven **energy-intensity** statistics (mean, median, standard deviation, range, volatility, peaks, and valleys, where volatility, peaks, and valleys directly operationalize the cross-step energy velocity $\Delta E(s_i \rightarrow s_{i+1})$ introduced in Section 3.3 by capturing the magnitude, local maxima, and local minima of step-to-step energy changes) and five **state-topology** statistics (cluster entropy, state revisits, unique transitions, transition diversity, and most frequent state ratio). We then adopt a *stratified* 70/10/20 train/validation/test split based on final answer correctness. A Logistic Regression model with ℓ_2 regularization is trained on the 70% training set, and the classification threshold is tuned on the 10% validation set to maximize F1-score. The final performance is then evaluated on the held-out 20% test set.

Baselines We compare SARE against five established baselines: (1) **Token Count** (Guo et al., 2025; Yang et al., 2025; Zhong et al., 2024): the total length of the CoT trajectory; (2) **Mean Log-Probability** (Kauf et al., 2024; Zhang and Liu, 2025): the average log-likelihood of generated tokens; (3) **Negative Entropy** (Zhao, 2026; Buffa and Del Corro, 2026): the mean negative Shannon entropy of the token distribution, we report negative entropy since larger values indicate higher confidence; (4) **Negative Perplexity** (Zhou et al., 2025; Geng et al., 2024): calculated as $-\exp(-\text{mean log-likelihood})$, we report negative perplexity since larger values correspond to higher confidence; and (5) **Self-Certainty** (Kang et al., 2025): the model's internal confidence score, computed as the KL-divergence between the token distribution and a uniform distribution. For every baseline, we apply the same threshold-tuning protocol on the validation set to ensure fair comparison.

Reasoning Energy and Token Count as Complementary Signals In our main results, we combine the trajectory-level vector described in Section 5.3 with **Token Count**, as the two provide complementary information and consistently improve performance. In the appendix, we report an ablation study without Token Count. Although absolute performance slightly decreases, the overall trends remain consistent with the results presented here (see Appendix Table 5-7).

A key design choice of our method is the integration of *Reasoning Energy* and *Token Count*. We view internal energy dynamics and trajectory length as complementary signals of reasoning correctness. While standard probabilistic baselines (e.g., Entropy or Log-Prob) primarily capture final-layer representation properties, which are already implicitly represented in our layer-wise Gram matrices, they often remove the token count information by averaging the token-wise information over the entire trajectory. By incorporating both internal dynamics and overall trajectory length, our method provides a more complete characterization of the model's reasoning process.

Empirical Findings (F1 Performance): Across three LLM families and six datasets (Tables 1–3), SARE demonstrates competitive discriminative power for reasoning error detection, matching or outperforming most baseline metrics.

A notable pattern is that four probabilistic baselines (Mean Log-Probability, Negative Entropy, Negative Perplexity, and Self-Certainty) report F1 = 0 on StrategyQA across all three models, which is not an implementation error, but a fundamental limitation of output-based confidence for binary True/False tasks, where token-level probability distributions are near-uniform across trajectories regardless of correctness and threshold tuning collapses to predicting all trajectories correct. Neither SARE nor Token Count exhibits this failure, retaining F1 of 0.46–0.53 and 0.46–0.58 respectively, as both operate on full trajectory features rather than final-token probabilities.

We note, however, that Token Count matches or outperforms SARE on StrategyQA for LLaMA-3.2-3B and Gemma-3-4B, suggesting that for short binary-answer tasks trajectory length captures much of the available signal; comparisons with the collapsed baselines on this dataset should be interpreted accordingly. This discriminative advantage over probabilistic baselines persists when trajectory length is excluded from the feature set, indicating that internal geometric dynamics encode predictive information beyond surface-level heuristics. AUPRC results (Appendix Tables 8–10) confirm the same overall trends.

Table 1: F1 Comparison: SARE vs. All Baselines for **LLaMA-3.2-3B**

Dataset	SARE	Token	LogProb	NegEnt	NegPerp	SelfCert
GSM8K	**0.532**	0.514	0.512	0.507	0.512	0.505
MATH	**0.801**	0.789	0.792	0.792	0.792	0.792
CSQA	**0.454**	0.448	0.436	0.431	0.436	0.423
HotpotQA	**0.900**	0.900	**0.900**	**0.900**	**0.900**	**0.900**
MuSiQue	**0.958**	0.947	0.954	0.954	0.954	0.947
StrategyQA	0.491	**0.580**	0.000	0.000	0.000	0.000

Table 2: F1 Comparison: SARE vs. All Baselines for **Gemma-3-4B**

Dataset	SARE	Token	LogProb	NegEnt	NegPerp	SelfCert
GSM8K	**0.293**	0.200	0.237	0.262	0.236	0.282
MATH	**0.746**	0.691	0.640	0.634	0.640	0.673
CSQA	**0.473**	0.471	0.451	0.415	0.451	0.465
HotpotQA	**0.893**	0.885	**0.893**	0.864	**0.893**	0.889
MuSiQue	**0.964**	**0.964**	0.961	**0.964**	0.961	**0.964**
StrategyQA	0.459	**0.460**	0.000	0.000	0.000	0.000

Table 3: F1 Comparison: SARE vs. All Baselines for **Phi-4-Mini**

Dataset	SARE	Token	LogProb	NegEnt	NegPerp	SelfCert
GSM8K	**0.621**	0.617	0.615	0.609	0.615	0.615
MATH	**0.769**	**0.769**	0.755	0.764	0.755	0.755
CSQA	0.462	**0.464**	0.462	0.462	0.462	**0.464**
HotpotQA	**0.908**	0.907	0.904	**0.908**	0.904	0.891
MuSiQue	**0.974**	0.968	**0.974**	**0.974**	**0.974**	0.968
StrategyQA	**0.526**	0.514	0.000	0.000	0.000	0.000

6 Conclusion

In this paper, we proposed **Step-Aware Reasoning Energy (SARE)**, a geometric framework that quantifies computational effort at the level of individual reasoning steps in chain-of-thought trajectories. By applying Centered Kernel Alignment (CKA) to token hidden state Gram matrices across consecutive transformer layers, SARE measures how much each step's internal token relationship geometry reorganizes during the forward pass, without requiring eigenvector alignment, cluster correspondence, or token-level aggregation. Experiments across six reasoning benchmarks and three open-source LLMs reveal that reasoning energy is highly non-uniform across semantic step types, that incorrect trajectories exhibit systematically lower energy at critical reasoning junctions prior to failure, and that energy dynamics encode predictive information about reasoning correctness beyond surface-level confidence measures.

In its current form, SARE is primarily a diagnostic and interpretability tool: it reveals *where* and *how much* computational effort is expended across a reasoning trajectory, and identifies energetic signatures that correlate with incorrect outcomes. Translating these diagnostic signals into actionable interventions is a natural next step. SARE scores could serve as a training-time signal to encourage the model to sustain higher geometric reorganization at critical reasoning junctions, analogous to process-reward models that supervise intermediate steps. At inference time, low-energy steps at high-stakes junctions could trigger targeted resampling or beam search expansion, and SARE profiles could inform early-exit strategies for steps that stabilize quickly. We hope the structured energy patterns identified here provide a useful foundation for geometry-aware reasoning improvement.

Limitations

Model scale. All experiments use open-weight LLMs in the 3–4B parameter range, spanning three architecturally distinct families (LLaMA-3.2-3B, Phi-4-mini, Gemma-3-4B). Whether the observed energy profiles and failure-detection patterns generalize to larger models is an open empirical question we leave to future work. That said, the consistency of findings across three families differing in training objectives, tokenizers, and attention configurations already provides initial cross-architecture evi-

dence that the geometric signal captured by CKA is not model-specific, offering some basis for expecting broader generalizability.

Computational overhead. SARE requires extracting hidden states at every transformer layer for every reasoning step, making it best suited to offline analysis in its current form. Approximations such as layer subsampling or mini-batch CKA estimation are natural directions for reducing this cost in large-scale settings.

LLM Usage. An LLM coding assistant was used to support portions of the implementation.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Pedro Memoli Buffa and Luciano Del Corro. 2026. Entropy sentinel: Continuous llm accuracy monitoring from decoding entropy traces in stem. *arXiv preprint arXiv:2601.09001*.

Wei-Lin Chen, Liqian Peng, Tian Tan, Chao Zhao, Blake JianHang Chen, Ziqian Lin, Alec Go, and Yu Meng. 2026. Think deep, not just long: Measuring llm reasoning effort via deep-thinking tokens. *arXiv preprint arXiv:2602.13517*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Angus Dempster, François Petitjean, and Geoffrey I Webb. 2019. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *arXiv preprint arXiv:1910.13051*.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jaehui Hwang, Dongyoon Han, Sangdoo Yun, and Byeongho Heo. 2026. Oops, wait: Token-level signals as a lens into llm reasoning. *arXiv preprint arXiv:2601.17421*.

Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*.

Carina Kauf, Emmanuele Chersoni, Alessandro Lenci, Evelina Fedorenko, and Anna A Ivanova. 2024. Log probabilities are a reliable estimate of semantic plausibility in base and instruction-tuned language models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 263–277.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A Smith. 2020. The right tool for the job: Matching model

and instance complexities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6640–6651.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Zhiguang Wang, Weizhong Yan, and Tim Oates. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Sheldon Yu, Yuxin Xiong, Junda Wu, Xintong Li, Tong Yu, Xiang Chen, Ritwik Sinha, Jingbo Shang, and Julian J. McAuley. 2025. Explainable chain-of-thought reasoning: An empirical analysis on state-aware reasoning dynamics. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 16660–16667. Association for Computational Linguistics.

Zihao Zhang and Fei Liu. 2025. Cost-augmented monte carlo tree search for llm-assisted planning. *arXiv preprint arXiv:2505.14656*.

Xinghao Zhao. 2026. Entropy trajectory shape predicts llm reasoning reliability: A diagnostic study of uncertainty dynamics in chain-of-thought. *arXiv preprint arXiv:2603.18940*.

Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Zeyu Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, and 1 others. 2024. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486*.

Zhi Zhou, Tan Yuhao, Zenan Li, Yuan Yao, Lan-Zhe Guo, Xiaoxing Ma, and Yu-Feng Li. 2025. Bridging internal probability and self-consistency for effective and efficient llm reasoning. *arXiv preprint arXiv:2502.00511*.

A Appendix

A.1 Semantic State Clustering: Implementation Details

Following Yu et al. (2025), we construct a spectral embedding for each reasoning step in three stages.

Step segmentation. CoT trajectories are generated using structured prompts (see Appendix A.2) that instruct the model to begin each reasoning step with the delimiter `Step X:` on its own line. Steps are extracted by matching this header pattern via regular expression.

Spectral embedding. For each sample, the full prompt-plus-trajectory text is passed through the LLM to obtain last-layer token hidden states. The hidden states are projected to 128 dimensions via a fixed random linear projection. For step s_t, the corresponding token span is identified via character-offset mapping, and a cumulative feature-covariance matrix is computed as $M_t = \sum_{i=1}^{t} \boldsymbol{H}_i^\top \boldsymbol{H}_i \in \mathbb{R}^{128\times128}$, where $\boldsymbol{H}_i$ is the projected token embedding matrix of step s_i. The top-64 eigenvalues of M_t by magnitude, computed via sparse eigendecomposition, form the 64-dimensional spectral embedding of s_t.

Clustering. K-Means clustering (K=7, Euclidean distance, `n_init=10`, `random_state=42`) is applied to the step embeddings pooled across all steps and samples in the dataset, yielding a hard cluster label for each step. The cluster sequence of a trajectory defines its state sequence for the Markov transition model.

A.2 Prompt Templates for CoT Generation

All CoT trajectories are generated using structured prompts that enforce explicit step delimiters. Each model is instructed to begin every reasoning step with `Step X:` on its own line, and to conclude with `The final answer is: {answer}`. Steps are subsequently extracted by matching this header pattern via regular expression `(Step\s*\d+\s*:.*?)(?=Step\s*\d+\s*:|$)`. Below we show representative prompts for each reasoning domain. Prompts for LLaMA-3.2-3B and Phi-4-mini are identical in format; Gemma-3-4B prompts additionally include a one-shot worked example to improve format compliance.

Math (LLaMA-3.2-3B / Phi-4-mini, e.g. GSM8K).

```
Solve the problem step-by-step. Your response
must follow these strict format rules:
1. Start your response directly with 'Step
1:'. Do NOT include any introductory text or
pleasantries.
2. Each logical step must begin with 'Step X:
' (where X is the sequential number).
3. Each step must be on a new line.
4. Each step must be on a single line.
5. End the entire solution with the phrase:
'The final answer is: {number}'.
6. Do not include any additional text or
explanation after the final answer.
7. Do not repeat the problem or your response
once you get the first final answer.

Problem: {question}

Solution:
```

Math (Gemma-3-4B, e.g. GSM8K) — includes one-shot example.

```
Solve the problem step-by-step. Keep your
reasoning concise and to the point.
[same format rules 1-4 as above]
5. End the entire solution with the phrase:
'The final answer is: {number}'.
[rules 6-8: no extra text, no repeated output,
stop after final answer]

- Example -
Problem: If John has 5 apples and eats 2, how
many does he have left?
Solution:
Step 1: Identify the initial number of apples,
which is 5.
Step 2: Identify the number of apples eaten,
which is 2.
Step 3: Subtract the number of apples eaten
from the initial number: 5 - 2 = 3.
The final answer is: 3
- End of Example -

- Task -
Problem: {question}

Solution:
```

Multi-hop QA (LLaMA-3.2-3B / Phi-4-mini, e.g. HotpotQA / MuSiQue).

```
Solve the multi-hop reasoning question
step-by-step. Your response must follow these
strict format rules:
[same format rules 1-4]
5. End the entire solution with the phrase:
'The final answer is: [Answer]'
where [Answer] is your short final result.
[rules 6-7: no extra text, no repeated output]

Question: {question}

Solution:
```

Commonsense QA (LLaMA-3.2-3B / Phi-4-mini, e.g. CSQA / StrategyQA).

```
Solve the commonsense question step-by-step
```

```
using the provided options.
[same format rules 1-4]
5. End the entire solution with the phrase:
'The final answer is: [Letter]'
where [Letter] is the single uppercase
character corresponding to your choice.
[rules 6-7: no extra text, no repeated output]

Question: {question}

Solution:
```

A.3 Experimental Protocol Details

Dataset splits and sample sizes. We use the standard test split for GSM8K, MATH, and StrategyQA, and the validation split for CSQA, HotpotQA, and MuSiQue. For each dataset we randomly sample 800 examples per model, with the exception of StrategyQA where the full test set of 687 examples is used.

Decoding. All CoT trajectories are generated with sampling (`temperature=0.7`, `top_p=0.9`, `max_new_tokens=512`). Up to five retries are attempted per sample if the output fails format validation (i.e., does not contain a parseable "The final answer is:" terminator).

Correctness evaluation. For math benchmarks (GSM8K, MATH), predicted and ground-truth answers are normalized by stripping punctuation and currency symbols and casting to float, then compared by exact match. For CSQA and StrategyQA, answers are normalized to the choice letter (A–E) and True/False respectively, then compared by exact match. For open-ended multi-hop benchmarks (HotpotQA, MuSiQue), string normalization is applied and a substring containment check is used as a fallback.

A.4 Dataset Statistics and Class Balance

Table 4 reports the number of evaluated samples and the fraction of correct trajectories for each dataset–model combination. Correctness rates vary substantially across datasets and models, from 5.0% on MuSiQue (Phi-4-mini) to 83.9% on GSM8K (Gemma-3-4B), motivating the use of stratified splits and threshold tuning in all classification experiments.

Table 4: Sample sizes and correct trajectory rates per dataset and model.

Dataset	Model	N	% Correct
GSM8K	Gemma-3-4B	800	83.9
GSM8K	LLaMA-3.2-3B	800	65.2
GSM8K	Phi-4-mini	800	55.1
MATH	Gemma-3-4B	800	49.1
MATH	LLaMA-3.2-3B	800	34.1
MATH	Phi-4-mini	800	38.0
CSQA	Gemma-3-4B	800	69.6
CSQA	LLaMA-3.2-3B	800	70.4
CSQA	Phi-4-mini	800	70.1
StrategyQA	Gemma-3-4B	687	69.7
StrategyQA	LLaMA-3.2-3B	687	59.1
StrategyQA	Phi-4-mini	687	64.6
HotpotQA	Gemma-3-4B	800	17.0
HotpotQA	LLaMA-3.2-3B	800	17.1
HotpotQA	Phi-4-mini	800	16.4
MuSiQue	Gemma-3-4B	800	7.5
MuSiQue	LLaMA-3.2-3B	800	8.2
MuSiQue	Phi-4-mini	800	5.0

Table 5: F1 Comparison: SARE (w/o Token Count) vs. All Baselines for **LLaMA-3.2-3B**

Dataset	SARE	Token	LogProb	NegEnt	NegPerp	SelfCert
GSM8K	**0.527**	0.514	0.512	0.507	0.512	0.505
MATH	**0.801**	0.789	0.792	0.792	0.792	0.792
CSQA	**0.454**	0.448	0.436	0.431	0.436	0.423
HotpotQA	**0.900**	0.900	**0.900**	**0.900**	**0.900**	**0.900**
MuSiQue	**0.958**	0.947	0.954	0.954	0.954	0.947
StrategyQA	0.536	**0.580**	0.000	0.000	0.000	0.000

Table 6: F1 Comparison: SARE (w/o Token Count) vs. All Baselines for **Gemma-3-4B**

Dataset	SARE	Token	LogProb	NegEnt	NegPerp	SelfCert
GSM8K	**0.305**	0.200	0.237	0.262	0.236	0.282
MATH	**0.717**	0.691	0.640	0.634	0.640	0.673
CSQA	**0.473**	0.471	0.451	0.415	0.451	0.465
HotpotQA	**0.893**	0.885	**0.893**	0.864	**0.893**	0.889
MuSiQue	**0.964**	**0.964**	0.961	**0.964**	0.961	**0.964**
StrategyQA	**0.462**	0.460	0.000	0.000	0.000	0.000

Table 7: F1 Comparison: SARE (w/o Token Count) vs. All Baselines for **Phi-4-mini**

Dataset	SARE	Token	LogProb	NegEnt	NegPerp	SelfCert
GSM8K	**0.628**	0.617	0.615	0.609	0.615	0.615
MATH	**0.769**	**0.769**	0.755	0.764	0.755	0.755
CSQA	0.462	**0.464**	0.462	0.462	0.462	**0.464**
HotpotQA	**0.908**	0.907	0.904	**0.908**	0.904	0.891
MuSiQue	**0.974**	0.968	**0.974**	**0.974**	**0.974**	0.968
StrategyQA	0.494	**0.514**	0.000	0.000	0.000	0.000

Table 8: AUPRC Comparison: SARE vs. All Baselines for **LLaMA-3.2-3B**

Dataset	SARE	Token	LogProb	NegEnt	NegPerp	SelfCert
GSM8K	**0.479**	0.420	0.286	0.257	0.286	0.246
MATH	**0.760**	0.728	0.585	0.560	0.585	0.548
CSQA	0.329	**0.331**	0.242	0.251	0.242	0.252
HotpotQA	0.861	**0.869**	0.730	0.708	0.730	0.698
MuSiQue	**0.921**	0.900	0.881	0.870	0.881	0.868
StrategyQA	0.382	**0.451**	0.000	0.000	0.000	0.000

Table 9: AUPRC Comparison: SARE vs. All Baselines for **Gemma-3-4B**

Dataset	SARE	Token	LogProb	NegEnt	NegPerp	SelfCert
GSM8K	**0.217**	0.199	0.159	0.156	0.159	0.210
MATH	**0.838**	0.779	0.511	0.539	0.511	0.600
CSQA	**0.500**	0.454	0.318	0.280	0.318	0.374
HotpotQA	**0.837**	0.813	0.718	0.731	0.718	0.751
MuSiQue	**0.926**	0.909	0.893	0.917	0.893	0.904
StrategyQA	**0.419**	0.390	0.000	0.000	0.000	0.000

Table 10: AUPRC Comparison: SARE vs. All Baselines for **Phi-4-mini**

Dataset	SARE	Token	LogProb	NegEnt	NegPerp	SelfCert
GSM8K	**0.644**	0.395	0.380	0.333	0.380	0.312
MATH	**0.702**	0.600	0.528	0.504	0.528	0.498
CSQA	0.372	**0.428**	0.260	0.246	0.260	0.225
HotpotQA	**0.894**	0.882	0.770	0.755	0.770	0.756
MuSiQue	**0.969**	0.958	0.964	0.946	0.964	0.940
StrategyQA	**0.367**	0.347	0.000	0.000	0.000	0.000