

Partial Reasoning in Language Models: Search and Refinement Guided by Uncertainty

Murilo da Luz^{1,2,*}, Bruno Brandão^{1,2}, Luana Martins^{1,2}, Gustavo Oliveira^{1,2}, Bryan de Oliveira^{1,2}, Luckeciano Melo^{1,3} and Telma Soares^{1,2}

¹Advanced Knowledge Center for Immersive Technologies (AKCIT)

²Federal University of Goiás, Brazil

³OATML, University of Oxford

Abstract

The use of Large Language Models (LLMs) for reasoning and planning tasks has drawn increasing attention in Artificial Intelligence research. Despite their remarkable progress, these models still exhibit limitations in multi-step inference scenarios, particularly in mathematical and logical reasoning. We introduce PREGU (Partial Reasoning Guided by Uncertainty). PREGU monitors the entropy of the output distribution during autoregressive generation and halts the process whenever entropy exceeds a defined threshold, signaling uncertainty. From that point, a localized search is performed in the latent space to refine the partial reasoning and select the most coherent answer, using the Soft Reasoning method. Experiments conducted with LLaMA-3-8B, Mistral-7B, and Qwen2-7B across four reasoning benchmarks (GSM8K, GSM-Hard, SVAMP, and StrategyQA) showed performance greater than or similar to Soft Reasoning, indicating that entropy can serve as an effective signal to trigger selective refinement during reasoning.

Keywords

Uncertainty, Entropy, Latent-space search, Soft Reasoning, LLM reasoning

1. Introduction

Large Language Models (LLMs) have demonstrated notable reasoning capabilities, often enhanced by techniques like Chain-of-Thought (CoT) prompting [1]. However, their performance often falters in tasks demanding complex, multi-step structured reasoning, such as advanced mathematics or contextual planning. This limitation stems from the challenge LLMs face in simulating long-term outcomes and exploring alternative reasoning paths, a process humans manage effectively [2].

Traditional reasoning strategies often operate in the vast space of tokens, which can be inefficient due to the immense number of possible token combinations. An alternative is performing search in the latent space, where representations are more abstract, compressed, and lower in dimensionality, avoiding successive natural language encoding and decoding.

The Soft Reasoning (SR) [3] method recently introduced exploration in the latent space by iteratively adjusting the input embedding (specifically, the first token) to influence the subsequent deterministic text generation. While effective, SR's efficacy is limited by its strong reliance on the starting point—optimizing the latent space only from the initial prompt. Uncertainty in LLMs is dynamic, typically manifesting in intermediate steps when the model must integrate facts or choose between plausible hypotheses. Restricting optimization to the beginning of the sequence risks focusing computational effort on regions where the model is already confident.

This research proposes PREGU (Figure 1), an adaptive extension of Soft Reasoning. PREGU dynamically identifies points of uncertainty during generation and triggers a focused search in the latent space only at those specific critical junctures.

LaCATODA 2026: The 10th Linguistic and Cognitive Approaches to Dialog Agents Workshop at the 40th AAAI conference, January 20–27, 2026, Singapore Expo

*Corresponding author.

✉ muriloluz@ufg.br (M. d. Luz)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

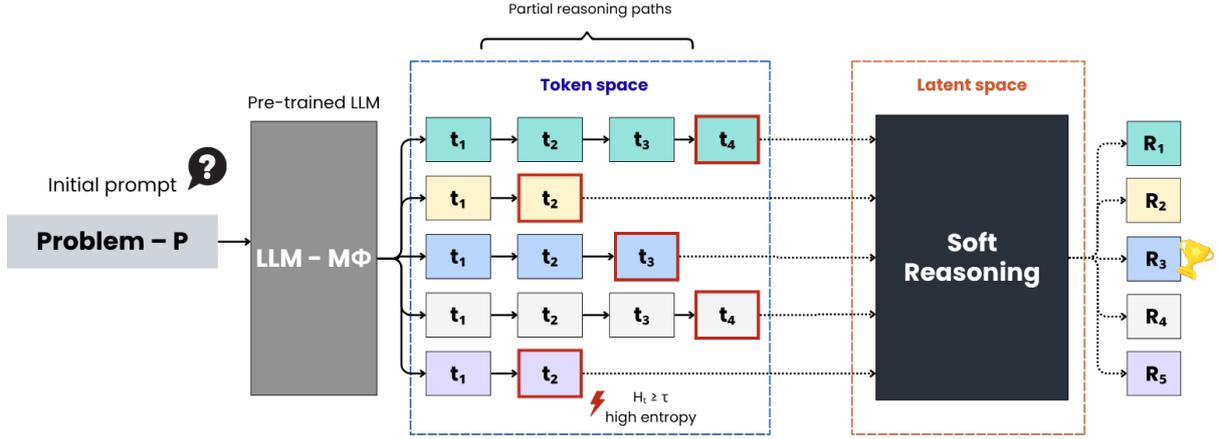


Figure 1: Overview of the PREGU architecture. The pre-trained LLM (M_ϕ) generates multiple partial reasoning paths in the token space. When the entropy of the output distribution (H_t) exceeds the threshold (τ), the generation halts, marking a region of uncertainty. Each partial reasoning sequence is then refined in the latent space using the Soft Reasoning method, producing candidate responses (R_1, R_2, \dots, R_n), from which the highest-reward answer is selected.

2. Background

2.1. Latent Space and Soft Reasoning

The latent space (or embedding space in the context of Large Language Models) is an abstract, multi-dimensional mathematical representation of input data (such as words or phrases) that encodes semantic, syntactic, and contextual information. Operating within this space allows LLM agents to function more efficiently by using compressed and abstract representations of environmental dynamics.

Soft Reasoning performs a search by applying controlled Gaussian perturbations ($\sigma\epsilon_i$) to the embedding of the first token, z . This exploration is guided by Bayesian Optimization using the *Expected Improvement* (EI) acquisition function [4]. The quality of the generated sequence (y), derived from a perturbed embedding (x), is evaluated through a reward function defined as:

$$f(x) = r_{\text{verifier}}(y) + r_{\text{coherence}}(y), \quad (1)$$

where r_{verifier} determines correctness – typically by employing the LLM itself as a black-box verifier through the *Multi-Generate* approach [3] – and $r_{\text{coherence}}$ assesses the semantic and syntactic fluency of the generated text. SR thus enables the model to refine its reasoning efficiently by exploring smooth variations in generation trajectories within the latent space.

2.2. Entropy as a Measure of Uncertainty

A standard method for quantifying an LLM’s internal uncertainty is by analyzing the probability distribution of its output tokens. The metric employed here is the *Shannon Entropy* (H_t), defined as:

$$H_t = - \sum_{j=1}^K p_t^{(j)} \log_2 p_t^{(j)}. \quad (2)$$

A higher entropy value signifies a flatter or more uniform probability distribution, indicating greater ambiguity or indecision in selecting the next token. By setting an entropy threshold (τ), the model’s autoregressive process can be interrupted when high uncertainty is detected, thereby activating an auxiliary reasoning mechanism. This approach mimics human dual cognition models (*System 1*: fast/intuitive; *System 2*: slow/analytical), where a slower, analytical process is triggered only when intuition encounters an impasse or high doubt [5].

Inspired by dual-process theories of cognition, PREGU mirrors the interaction between intuitive and analytical reasoning systems: a fast, heuristic mode during confident segments and a slower, deliberative intervention when uncertainty peaks. In this view, the entropy threshold τ acts as a computational analogue of metacognitive control, triggering a shift from automatic generation to reflective reasoning.

3. The PREGU Method

Unlike prompt-based diversification strategies such as Chain-of-Thought [6], Self-Consistency [7], ReAct [8], or Tree of Thoughts [9], which primarily vary surface-level text to elicit reasoning, PREGU reallocates computation toward points of uncertainty detected via Shannon entropy. This direction follows latent-space reasoning explored by Soft Reasoning [3], emphasizing optimization over internal representations rather than prompt reformulation.

PREGU (Partial Reasoning Guided by Uncertainty) is designed to overcome the limitation of Soft Reasoning (SR), which optimizes only the initial token embedding. PREGU integrates entropy-based detection with latent space refinement in a two-stage process (Figure 1).

Stage 1: Partial Reasoning Generation (Token Space)

We denote by M_ϕ a pretrained language model with parameters ϕ . The language model M_ϕ performs autoregressive generation to produce N candidate reasoning paths. At each decoding step t , we monitor the Shannon entropy H_t of the next-token distribution. For each path, generation is interrupted *at most once*, exactly at the *first* token whose entropy crosses the threshold τ (i.e., the first t such that $H_t \geq \tau$), yielding a partial reasoning prefix.

Entropy is estimated over the top- K tokens (with $K = 50$), balancing computational cost and representativeness; very low-probability tokens contribute minimally to the total uncertainty.

Minimum Prefix (t_{\min}): the model must generate at least t_{\min} tokens before uncertainty detection is allowed, ensuring sufficient semantic context.

Stage 2: Refinement (Latent Space)

Each partial reasoning sequence generated in Stage 1 is treated as an extended prompt root (p_i). This partial reasoning is then refined independently using the Soft Reasoning method. The search is focused on the latent space starting from the point of interruption, allowing Bayesian Optimization to specifically address the ambiguity that triggered the halt.

The process yields multiple candidate answers (R_1, R_2, \dots, R_n), and the final solution (a^*) is selected based on the highest reward computed by the equation 1.

This structure combines *breadth exploration* (through multiple partial reasoning paths) with *focused depth exploration* (via latent space refinement), enabling a more adaptive and uncertainty-aware reasoning process.

4. Experimental Setup

4.1. Configuration

Experiments were conducted using three intermediate-scale (7–8 billion parameter) open-source LLMs under a zero-shot setting: **LLaMA-3-8B** [10], **Mistral-7B** [11], and **Qwen2-7B** [12]. These models present distinct architectural characteristics: LLaMA-3 emphasizes generalization, Mistral focuses on architectural efficiency through *Sliding Window Attention* (SWA), and Qwen2 prioritizes robustness in mathematical reasoning tasks.

Benchmarks. The methodology was evaluated against four reasoning benchmarks:

- **GSM8K:** Grade-school mathematical word problems [13].
- **GSM-Hard:** A more challenging variant of GSM8K, requiring multi-step reasoning and compositional inference [14].
- **SVAMP:** Tests semantic robustness by introducing superficial structural variations in mathematical problems [15].
- **StrategyQA:** Evaluates strategic and commonsense reasoning, requiring the decomposition of implicit sub-hypotheses [16].

Baselines. We compare PREGU and Soft Reasoning against standard prompting and decoding baselines on GSM8K, GSM-Hard, SVAMP, and StrategyQA. **CoT** (Chain-of-Thought) encourages step-by-step reasoning by prompting the model to generate intermediate rationale before the final answer [17]. **SC** (Self-Consistency) samples multiple CoT solutions (e.g., using different temperatures) and selects the final answer by consensus (majority vote) [7]. **FIRE** increases diversity by applying a high temperature only to the first generated token while decoding subsequent tokens with regular execution sampling [18]. **CoT-Decoding** further diversifies reasoning by starting generation from the top- k most likely first tokens and decoding a completion from each start [19]. For these baselines, we adopt the same evaluation protocol and report the baseline results as presented in the Soft Reasoning study [3].

Hyperparameters. PREGU was evaluated using a standard hyperparameter configuration informed by preliminary studies of token-level entropy (see Sec. 5.1), including:

- Entropy threshold: $\tau = 3.0$ bits
- Token sample size for entropy estimation: $K = 50$

Table 1

Average accuracy (%) — mean \pm std over five runs. Bold marks best per benchmark.

Model	Method	GSM8K	GSM-Hard	SVAMP	StrategyQA
		<i>Zero-Shot</i>	<i>Zero-Shot</i>	<i>Zero-Shot</i>	<i>Zero-Shot</i>
LLaMA-3-8B	CoT	53.0 \pm 0.0	14.0 \pm 0.0	61.0 \pm 0.0	58.5 \pm 0.0
	SC($T=0.4$)	73.0 \pm 1.6	25.7 \pm 0.4	79.1 \pm 1.2	64.7 \pm 0.7
	SC($T=0.6$)	73.6 \pm 2.5	24.5 \pm 1.1	76.1 \pm 3.9	59.9 \pm 2.0
	SC($T=0.8$)	65.0 \pm 2.0	21.8 \pm 1.3	69.6 \pm 2.0	54.4 \pm 2.6
	FIRE	73.8 \pm 2.3	25.2 \pm 3.0	81.5 \pm 0.8	63.0 \pm 3.7
	CoT-Decoding	73.9 \pm 1.9	24.8 \pm 1.3	83.2 \pm 1.2	64.6 \pm 1.6
	Soft Reasoning	79.4 \pm 1.2	28.2 \pm 1.8	88.2 \pm 1.3	67.2 \pm 0.7
	PREGU (ours)	82.6 \pm 1.6	35.2 \pm 3.4	87.4 \pm 0.9	68.6 \pm 0.7
Qwen2-7B	CoT	64.5 \pm 0.0	40.0 \pm 0.0	43.5 \pm 0.0	63.0 \pm 0.0
	SC($T=0.4$)	81.2 \pm 0.6	47.5 \pm 1.4	72.3 \pm 2.0	67.1 \pm 1.5
	SC($T=0.6$)	80.2 \pm 1.9	46.2 \pm 1.9	77.3 \pm 1.2	67.5 \pm 0.7
	SC($T=0.8$)	80.0 \pm 0.9	47.3 \pm 1.3	78.6 \pm 2.1	67.0 \pm 1.0
	CoT	42.0 \pm 0.0	14.5 \pm 0.0	52.0 \pm 0.0	62.0 \pm 0.0
	SC($T=0.4$)	52.9 \pm 0.5	19.5 \pm 1.0	67.4 \pm 2.5	69.3 \pm 1.5
	SC($T=0.6$)	55.1 \pm 3.6	20.7 \pm 1.5	69.7 \pm 1.6	64.2 \pm 1.0
	SC($T=0.8$)	50.2 \pm 2.6	19.1 \pm 2.0	68.3 \pm 0.9	64.9 \pm 1.0
Mistral-7B	FIRE	47.2 \pm 2.9	18.1 \pm 1.9	67.1 \pm 1.9	64.2 \pm 1.0
	CoT-Decoding	47.3 \pm 3.0	16.6 \pm 0.7	69.4 \pm 2.5	63.5 \pm 1.5
	Soft Reasoning	61.4 \pm 2.5	25.8 \pm 1.8	72.2 \pm 2.2	66.1 \pm 1.9
	PREGU (ours)	66.0 \pm 1.9	32.7 \pm 1.4	74.8 \pm 0.8	68.1 \pm 1.9

- Search width: $N = 5$ partial reasoning paths
- *Soft Reasoning* parameters: $k = 5$ samples, projected latent dimension $d = 50$

5. Results

The results in Table 1 indicate that PREGU generally matches or improves upon Soft Reasoning across the evaluated benchmarks. We observe small regressions in isolated settings (e.g., Qwen-2-7B on GSM8K and LLaMA-3-8B on SVAMP), while maintaining gains on GSM-Hard, StrategyQA, and most remaining cases. Across five runs, the method remained stable and did not exhibit systematic performance degradation, suggesting that combining entropy-guided interruption with localized latent-space refinement may improve reasoning consistency across domains, especially on multi-step inference chains such as GSM-Hard.

Notably, performance on the GSM-Hard dataset improved, where PREGU showed robustness on long, multi-step inference chains. Even with the search being initiated only at the first point of uncertainty, the method successfully refined reasoning trajectories, suggesting entropy is an effective critical-point selection metric.

5.1. Empirical Validation of Uncertainty

An analysis of the detected points of uncertainty revealed a clear linguistic correlation. Tokens exhibiting the highest average entropy—such as “For”, “Given”, and “Since”—frequently marked logical transitions, the introduction of sub-problems, or causal relations. Other high-entropy tokens, including “First” and “According”, were also observed to initiate new reasoning steps. These tokens correspond precisely to structural moments in reasoning where the model must choose among multiple plausible logical continuations.

At these critical junctures, the probability distribution over output tokens becomes notably dispersed, reflecting the model’s indecision regarding the most coherent semantic direction.

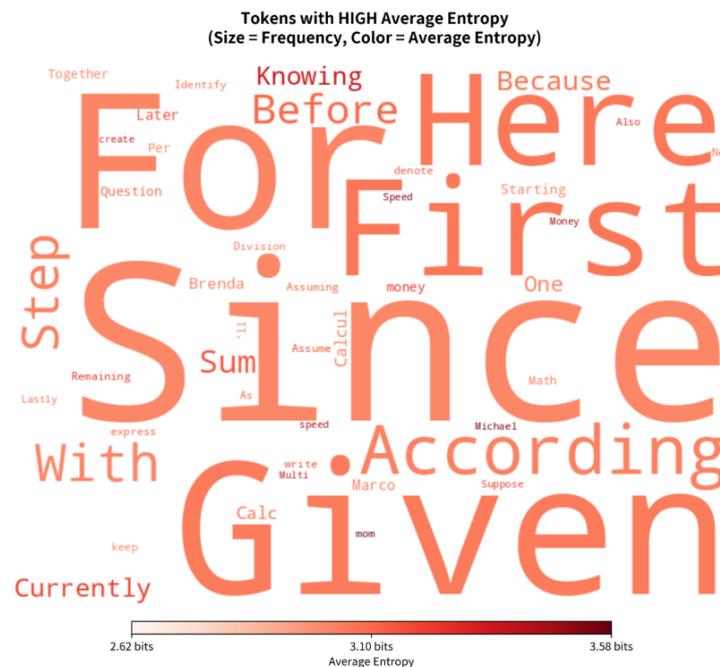


Figure 2: Word cloud of tokens with high average entropy observed in the SVAMP dataset. Token size represents frequency and color encodes the average entropy (in bits), highlighting terms such as “For”, “Since”, and “Given”, which are associated with logical transitions in reasoning.

These entropy spikes frequently coincide with discourse connectives (e.g., “for”, “since”, “given”), suggesting that the model’s uncertainty aligns with structural transitions in linguistic reasoning. Such correspondence supports the view that entropy can reveal moments of linguistic introspection, where the model implicitly evaluates competing semantic paths before proceeding.

This pattern supports the hypothesis that entropy functions as an indicator of *local cognitive complexity*: points where fast, automatic generation (*System 1*) gives way to slower, more analytical reasoning (*System 2*). Thus, the entropy-based mechanism effectively monitors internal signals of cognitive hesitation, transforming them into opportunities for reflective refinement.

The distribution of high-entropy tokens is visualized through a word cloud (Figure 2), highlighting terms associated with logical transitions in reasoning. This linguistic correspondence suggests that the method captures internal traces of hesitation within the model’s own generative process, approximating a primitive form of introspection.

6. Conclusion and Future Work

PREGU successfully achieved its objective by developing an adaptive architecture that integrates partial reasoning generation in the token space with localized refinement in the latent space, guided by entropy. The approach validates the hypothesis that internal uncertainty can serve as a metacognitive control mechanism to direct computational effort toward regions offering the highest potential for informational gain.

Known Limitations

PREGU inherits certain limitations from the original *Soft Reasoning* approach. First, the scope of Bayesian Optimization remains restricted, as it operates only on the initial embedding following a detected uncertainty point, thereby constraining refinement to a localized region of the latent space. Second, the method’s reliance on the internal verifier (r_{verifier})—which employs the LLM itself to assess the correctness of its own outputs—can introduce bias, particularly in complex mathematical verification tasks. Finally, the fixed entropy threshold (τ) constitutes a sensitive hyperparameter: inadequate calibration may either cause premature fragmentation of reasoning sequences or reduce the frequency of uncertainty detection, effectively reverting the behavior toward that of the original Soft Reasoning method.

Future Work

Future research should aim to expand the scope of optimization within the latent space to encompass a window of multiple embeddings following the point of uncertainty, rather than a single one. This broader search may enable smoother and more contextual refinements. Another promising direction involves designing new reward metrics that reduce dependence on the LLM’s self-evaluation, for instance by integrating external or symbolic verifiers for mathematical reasoning. Additionally, dynamically calibrating the entropy threshold (τ)—adapting it to the problem’s complexity and context—and quantifying explicit cost-benefit metrics (e.g., execution time, inference cost per correct solution) represent important steps toward improving PREGU’s efficiency, interpretability, and robustness.

Acknowledgments

This work has been fully/partially funded by the project Research and Development of Digital Agents Capable of Planning, Acting, Cooperating and Learning supported by Advanced Knowledge Center in Immersive Technologies (AKCIT), with financial resources from the PPI IoT/Manufatura 4.0 / PPI HardwareBR of the MCTI grant number 057/2023, signed with EMBRAPII

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT (GPT-5.2) to assist with translation and language editing (grammar, spelling, and clarity). The author(s) reviewed and edited the output and take full responsibility for the final content.

References

- [1] B. Wang, S. Min, X. Deng, J. Shen, Y. Wu, L. Zettlemoyer, H. Sun, Towards understanding chain-of-thought prompting: An empirical study of what matters, 2023. URL: <https://arxiv.org/abs/2212.10001>. arXiv:2212.10001.
- [2] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, Z. Hu, Reasoning with language model is planning with world model, 2023. URL: <https://arxiv.org/abs/2305.14992>. arXiv:2305.14992.
- [3] Q. Zhu, R. Zhao, H. Yan, Y. He, Y. Chen, L. Gui, Soft reasoning: Navigating solution spaces in large language models through controlled embedding exploration, 2025. URL: <https://arxiv.org/abs/2505.24688>. arXiv:2505.24688.
- [4] P. I. Frazier, A tutorial on bayesian optimization, 2018. URL: <https://arxiv.org/abs/1807.02811>. arXiv:1807.02811.
- [5] D. Kahneman, Thinking, fast and slow, Penguin, London, 2012.
- [6] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2023. URL: <https://arxiv.org/abs/2201.11903>. arXiv:2201.11903.
- [7] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, 2023. URL: <https://arxiv.org/abs/2203.11171>. arXiv:2203.11171.
- [8] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, 2023. URL: <https://arxiv.org/abs/2210.03629>. arXiv:2210.03629.
- [9] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, K. Narasimhan, Tree of thoughts: Deliberate problem solving with large language models, 2023. URL: <https://arxiv.org/abs/2305.10601>. arXiv:2305.10601.
- [10] A. Grattafiori, A. Dubey, A. Jauhri, et al., The llama 3 herd of models, 2024. URL: <https://arxiv.org/abs/2407.21783>. arXiv:2407.21783.
- [11] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. URL: <https://arxiv.org/abs/2310.06825>. arXiv:2310.06825.
- [12] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Yang, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, X. Liu, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Guo, Z. Fan, Qwen2 technical report, 2024. URL: <https://arxiv.org/abs/2407.10671>. arXiv:2407.10671.
- [13] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, J. Schulman, Training verifiers to solve math word problems, 2021. URL: <https://arxiv.org/abs/2110.14168>. arXiv:2110.14168.
- [14] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, G. Neubig, Pal: Program-aided language models, arXiv preprint arXiv:2211.10435 (2022).
- [15] A. Patel, S. Bhattamishra, N. Goyal, Are nlp models really able to solve simple math word problems?, 2021. URL: <https://arxiv.org/abs/2103.07191>. arXiv:2103.07191.
- [16] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, J. Berant, Did aristotle use a laptop? a question

- answering benchmark with implicit reasoning strategies, 2021. URL: <https://arxiv.org/abs/2101.02235>. arXiv:2101.02235.
- [17] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, 2023. URL: <https://arxiv.org/abs/2205.11916>. arXiv:2205.11916.
- [18] W. Chen, Z. Zhang, G. Liu, R. Zheng, W. Shi, C. Dun, Z. Wu, X. Jin, L. Yan, Flaming-hot initiation with regular execution sampling for large language models, 2025. URL: <https://arxiv.org/abs/2410.21236>. arXiv:2410.21236.
- [19] X. Wang, D. Zhou, Chain-of-thought reasoning without prompting, 2024. URL: <https://arxiv.org/abs/2402.10200>. arXiv:2402.10200.

A. PREGU - Experiments

Experimental Setup and Reproducibility

All experiments were executed on an NVIDIA DGX-H100 system equipped with 8× H100 GPUs (80 GB each).

The models evaluated were **LLaMA-3-8B**, **Mistral-7B**, and **Qwen2-7B**, selected for their architectural diversity and open accessibility. All experiments followed the same inference configuration, including token truncation to the top- k ($k = 50$) logits for entropy estimation and a fixed entropy threshold of $\tau = 3.0$ bits. Uncertainty detection followed $t_{\min} = 5$ (minimum prefix), and we trigger at most one interruption per path (first token where $H_t \geq \tau$).

For the latent-space optimization phase, Bayesian Optimization was applied over a projected subspace of $d = 50$ dimensions, using the Expected Improvement (EI) acquisition function. Each refinement round sampled $k = 5$ latent perturbations per reasoning path, guided by the composite reward function in Eq. 1.

To account for stochasticity in both token sampling and latent perturbation, each experimental configuration was executed across multiple independent runs with distinct random seeds. All results were reported as mean and standard deviation to capture the natural variability of large language model inference. Benchmarks included GSM8K, GSM-Hard, SVAMP, and StrategyQA under the zero-shot setting, following the standard prompt structures described in the methodology.

For each benchmark, a random subset of 200 problems was uniformly sampled from the full dataset at the beginning of each run. This random selection ensured that different executions explored distinct subsets while maintaining comparable task difficulty distributions. The random seed controlling the sampling process was synchronized with the inference seed, guaranteeing consistency between data selection and generation randomness across runs.

Finally, representative examples from different benchmarks are included in the Appendix to illustrate the full execution trace of PREGU. Each example shows both stages of operation—entropy-based interruption and latent-space refinement—highlighting how uncertainty localization leads to structured reasoning improvements. These examples provide a transparent depiction of the reasoning process and facilitate independent verification of the experimental pipeline.

Note. The examples reported here correspond to raw model outputs. Therefore, they may include hallucinations or other generation artifacts (e.g., spurious claims or incoherent fragments), and are provided solely to illustrate the behavior of the proposed pipeline.

A.1. PREGU Execution Example — StrategyQA

Initial Question. *Mayor: head of municipal government such as a town or city. Are Mayors safe from harm from the federal government?*

Ground Truth: False

Phase 1: Partial Reasoning Generation

(Entropy pause marked as [STOP]. Rewards refer to the partial prefix quality.)

Path 0

1) Mayors are[STOP]

Path 1

1) Mayor is a position of municipal government. 2) The municipal government,[STOP]

Path 2 (Selected)

1) The mayor is the head of a municipal government. 2) The federal government is a higher level of government. 3) The constitution grants[STOP]

Path 3

The Mayor is the local leader of a city or town. A federal government[STOP]

Path 4

1) The Mayor is the leader of the local government and not part of the federal government.
2) No[STOP]

Phase 2: Complete Reasoning Chains

(Each path is refined in latent space and yields a complete answer.)

Path 0 – Complete Chain

... auge officials of municipal governments, which are part of the state government, not the federal government. 2) The federal government has no direct authority over municipal governments, but it can influence them through various means such as funding or laws. 3) However, the federal government does not have the power to harm mayors directly, as they are not federal officials.

Answer:

True. Mayors are safe from harm from the federal government.

Path 1 – Complete Chain

... ooften, has its own laws and regulations, but it is part of the larger national government. 3) The federal government has the power to enforce laws and regulations, and it can intervene in the affairs of municipal governments if necessary. 4) However, the federal government is not typically involved in the day-to-day operations of municipal governments, and mayors are usually free to carry out their duties without interference from the federal government. 5) Therefore, mayors are generally safe from harm from the federal government, but there are exceptions if the mayor is breaking federal laws or if there is a national emergency.

Answer:

Mayors are generally safe from harm from the federal government, but there are exceptions.

Answer: mayors are enerally safe from harm from the federal overnment but there are exceptions

Path 2 – Complete Chain (Selected)

... kh the power to the federal government to regulate the states and municipalities. 4) The federal government can use its power to investigate and prosecute individuals, including mayors, if they are suspected of violating federal laws. 5) However, the federal government is not allowed to arbitrarily harm or threaten mayors.

Answer:

False. Mayors are not safe from harm from the federal government if they are suspected of violating federal laws, but the federal government is not allowed to arbitrarily harm or threaten mayors.

Path 3 – Complete Chain

... iederates over the entire country, not just a single city or town. Therefore, the federal government has the power to intervene in the affairs of a city or town, including the removal or replacement of the Mayor. So, Mayors are not safe from harm from the federal government.

Answer: False.

Path 4 – Complete Chain

... dit: The Mayor is not a federal official, so they are not under the protection of the federal government. 3) The federal government has no direct authority over the Mayor unless there is a violation of federal law. 4) The Mayor is not immune from prosecution if they break federal law.

Answer:

False. The Mayor is not safe from harm from the federal government if they break federal law.

A.2. PREGU Execution Example – GSM8k

Initial Question. Ashley’s pizza delivery costs \$15. What is the total amount that Ashley should give the delivery man if she wants to give a tip that is equal to 1/5 of the amount she ordered?

Ground Truth: 18

Phase 1: Partial Reasoning Generation

(Entropy pause marked as [STOP].)

Path 0

1/5 means Ashley should give 1/5 of the total price.[STOP]

Path 1

1/5 of the ordered amount is the tip Ashley wants to give. So, to find the amount of the tip, we[STOP]

Path 2 (Selected)

Ashley's tip should be 1/5 of what she paid for a delivery, so the tip amount is 1/5 * \$15.[STOP]

Path 3

1/5 of Ashley's pizza delivery is her tip. So, to find the tip amount, we[STOP]

Path 4

1/5 (twentieth) of the[STOP]

Phase 2: Complete Reasoning Chains

(Each path is refined in latent space and yields a complete answer.)

Path 0 – Complete Chain

... aille's pizza delivery costs \$15, so we need to find 1/5 of \$15.

Answer: \$3

Path 1 – Complete Chain

... RPC:

1. Multiply the cost of the pizza by 1/5.
2. Add the tip to the cost of the pizza.

Answer:

\$16.50 ($15 + (15 * 1/5)$)

Path 2 – Complete Chain (Selected)

... Now, to find the total amount Ashley should give the delivery man, we add the tip amount to the cost of the delivery: $\$15 + (1/5 * \$15)$.

Answer:

\$18

Path 3 – Complete Chain

... Luxury Real Estate in Dubai will divide the cost of the pizza delivery by 5. Then, we will add the tip amount to the cost of the pizza delivery to find the total amount Ashley should give the delivery man.

Answer:

$\$15 + (\$15 / 5) = \$15 + \$3 = \$18$.

Path 4 – Complete Chain

... imanager's cost is the tip. So, we need to find 1/5 of \$15.

Answer: \$3

Selection Rule. Each partial reasoning sequence generated in *Stage 1* serves only as an exploratory prompt. The final selection among reasoning paths occurs after the complete refinements in the latent space (*Stage 2*), based on the final evaluation scores. The chosen path thus corresponds to the reasoning that achieved the highest overall reward according to the combined objective function.

Discussion. The examples provided demonstrate the operational dynamics of PREGU, showing how uncertainty detection and localized latent-space optimization interact to enhance reasoning reliability. Together, they offer a transparent view of the model's decision process and its transition from intuitive to analytical reasoning.