TIDES: Entropy-guided Decoding Phase Transition via Temporal Context Diffusion for Small Language Models

Anonymous ACL submission

Abstract

For autoregressive language models, inference-time methods have been widely explored to enhance model capability. Decoding adjusts token-level sampling locally, while reasoning modifies output structure without altering the decoding mechanism itself. However, neither provides an explicit mechanism for deciding when to stop exploring and commit to an answer, a limitation especially acute for small language models (SLMs), which cannot rely on scale to compensate for suboptimal inference. To address this, we propose **TIDES** (**T**raining-free **I**nference-time **D**iffusion via **E**ntropy-guided **S**core functions), a reasoning-aware decoding method that treats per-step token entropy as a commit signal to drive an explicit exploration-exploitation transition. The resulting generation trajectory participates in a noising-then-denoising diffusion spectrum, realized as temporal context diffusion across the generated sequence. Evaluated on diverse benchmarks with two SLMs, TIDES consistently improves over zero-shot CoT while remaining competitive with existing methods at approximately $1.8\times$ the cost of greedy decoding, and avoids over-thinking regression on knowledge-recall tasks.

1 Introduction

As language models have scaled in capacity, it has been observed that models can perform in-context learning (ICL) (Dong et al., 2024), adapting to new tasks at inference time without parameter updates. This finding marked a shift from training-centric to inference-centric improvement, motivating a growing body of inference-time methods that operate on frozen models. The inference-centric paradigm is especially consequential for small language models (SLMs): unlike large models, which can absorb suboptimal inference through scale, SLMs must extract maximum utility from every decode step.

On the decoding side, temperature reshapes the next-token distribution locally but cannot rearrange its modes, leaving the global context unchanged (Zhang et al., 2024). On the reasoning side, self-consistency (Wang et al., 2023) aggregates over multiple sampled paths but leaves the decoding mechanism itself unaltered.

Latent reasoning (Hao et al., 2024) challenges the premise of token-level reasoning, arguing that most tokens serve textual coherence rather than genuine reasoning. Instead, it performs autoregressive computation in the continuous hidden space, though operating within these black-box representations leaves the exploration-exploitation transition implicit.

Despite this progress, none of these methods provides an explicit exploration-exploitation transition during generation, resulting in two failure modes. (1) Under-exploration: the model prematurely commits to the first plausible reasoning path, foreclosing better solutions. (2) Over-exploration: the model expends unnecessary computation on queries whose answers are readily available, occasionally drifting from a correct brief response. In both cases, the model has no commit signal.

To this end, we propose **TIDES** (**T**raining-free **I**nference-time **D**iffusion via **E**ntropy-guided **S**core functions), a reasoning-aware decoding method that treats per-step token entropy as a commit signal to drive an explicit exploration-exploitation transition. Specifically, TIDES views the token sequence as a temporal axis: tokens in the exploration phase are produced under progressively increasing BOS noise, and those in the exploitation phase under decreasing noise, so the generation trajectory participates in a noising-then-denoising diffusion spectrum.

We evaluate TIDES on eight benchmarks spanning Math, STEM, and Knowledge Recall using Qwen2.5-1.5B and 3B (Yang et al., 2024). TIDES consistently improves over zero-shot CoT on tasks within the model's latent capability, while remain-

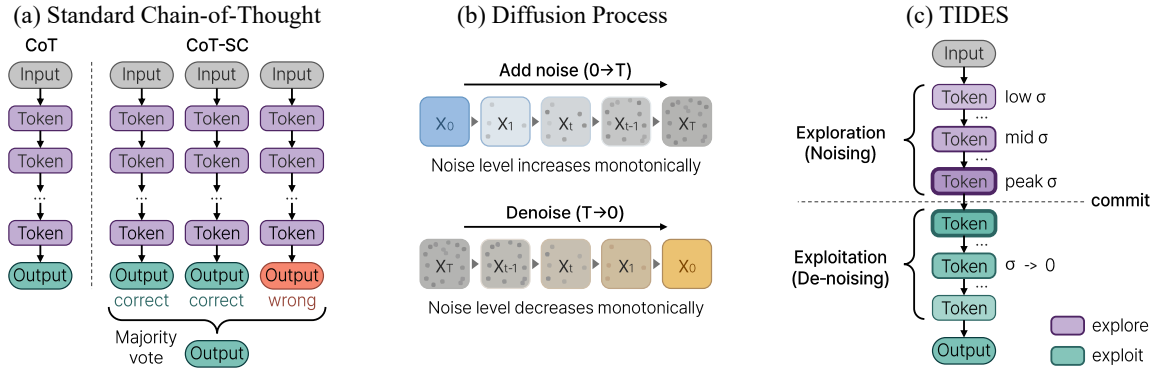

Figure 1: Conceptual overview of TIDES. (a) Standard CoT and CoT-SC generate token sequences under fixed decoding. (b) Diffusion applies monotonically increasing then decreasing noise. (c) TIDES induces an entropy-guided noising-denoising phase transition, forming a temporal context diffusion spectrum along the token sequence.

ing competitive with existing inference-time methods with minimal computational overhead and avoiding over-thinking that leads to incorrect answer.

2 Method

Let f_θ be a frozen autoregressive language model with embedding matrix E, and let $e_{\text{BOS}} := E[\text{BOS}] \in \mathbb{R}^d$ be the BOS token embedding. We run greedy autoregressive decoding while dynamically updating the position-0 slot of KV cache. Every generated token attends to the BOS position, so modifying it applies a global context shift to the entire continuation without touching any prompt token or model weight, achieving what temperature cannot: rearranging the distribution modes of all subsequently generated tokens simultaneously.

Energy and score estimator. At step t, we use the Gaussian-smoothed entropy $H_\sigma(e) := \mathbb{E}_{u\sim\mathcal{N}(0,I_d)}[H(e+\sigma u)]$ (Duchi et al., 2012; Nesterov and Spokoiny, 2017) as a smooth surrogate of the non-smooth raw entropy $H_t := -\sum_v p_t(v)\log p_t(v)$. Its gradient signal is estimated via an antithetic two-point zeroth-order (ZO) directional finite-difference proxy (Spall, 1992; Berahas et al., 2022),

$$\widehat{\nabla H_\sigma}(e) = \frac{H(e+\sigma u) - H(e-\sigma u)}{2\sigma}\, u, \quad u \sim \text{Unif}(S^{d-1}),$$

with a single scale $\varepsilon \equiv \sigma$ for both perturbation and smoothing radius. A trust-region clip $\|\widehat{\nabla H_\sigma}\| \le 1$ prevents large-magnitude artifacts.

Langevin update. The BOS embedding is updated via an SDE-inspired discrete update,

$$e_{t+1} \leftarrow e_t - \frac{\sigma^2}{2}\,\widehat{\nabla H_\sigma}(e_t) + \sigma\,\xi_t, \quad \xi_t \sim \mathcal{N}(0, I),$$

following the Euler–Maruyama form of overdamped Langevin diffusion (Roberts and Tweedie, 1996; Welling and Teh, 2011), where the drift follows the entropy proxy and the noise maintains exploration diversity.

Phase controller. The σ schedule is governed by an entropy-guided phase controller, centered on the peak baseline $H_{\text{peak,warmup}} := \max_{t<k_{\text{warm}}} H_t$, which is frozen after k_{warm}=10 tokens, inspired by warmup-based optimizers that delay large steps until estimates stabilise.

During the exploration phase, while $H_t \ge \frac{1}{2}H_{\text{peak,warmup}}$, σ_t ramps toward $\sigma_{\max}$ at a rate proportional to running entropy variability,

$$\sigma_{t+1} = \min\big(\sigma_t + 0.1 \cdot \text{std}(H_{1:t}),\ \sigma_{\max}\big),$$

and the BOS embedding is updated every k=5 steps. Once H_t drops below the threshold, the exploitation phase begins: σ decays geometrically, with Langevin updates continuing until σ vanishes and generation reverts to greedy.

The commit threshold $\frac{1}{2}H_{\text{peak,warmup}}$ follows from a geometric-mean argument rather than tuning: in branching-factor space $B{:=}\exp(H)$, it equals $\sqrt{B_{\text{peak,warmup}}}$, the geometric mean of the peak and the deterministic limit (B=1), the natural midpoint for an entropy-valued quantity. We set $\sigma_{\max} = \|e_{\text{BOS}}\|/\sqrt{d \cdot \text{SNR}}$, where SNR=0.25 is the method's single free hyperparameter.

Temporal context diffusion. Viewing the token sequence as a temporal axis, tokens in the forward phase are produced under progressively increasing BOS noise; tokens in the backward phase under decreasing noise. The sequence thus participates in a noising-then-denoising diffusion spectrum along the temporal axis. Detailed our main key concepts are provided in Appendix A.

LLM	Concept	Method	Math						STEM						Knowledge Recall			
			GSM8K		AMC23		AIME24		MMLU-STEM		TheoremQA		GPQA		TriviaQA		PopQA	
			Acc.	Avg Tok.	Acc.	Avg Tok.	Acc.	Avg Tok.	Acc.	Avg Tok.	Acc.	Avg Tok.	Acc.	Avg Tok.	Acc.	Avg Tok.	Acc.	Avg Tok.
Qwen2.5 Instruct (1.5B)	Baseline	ZS-CoT	70.20	241.5	30.00	744.9	3.33	1007.2	53.40	312.1	16.00	598.6	22.73	607.1	31.00	3.6	11.20	3.6
	Decoding	AdaTemp	71.20	243.7	27.50	818.2	0.00	1667.0	55.40	297.0	**20.60**	593.5	22.73	610.0	30.80	3.7	11.60	3.4
		CD (DoLa)	60.60	194.5	15.00	1051.3	3.33	2732.3	48.60	322.9	11.80	766.3	23.23	901.3	28.20	3.5	12.00	3.3
	Reasoning	FS-CoT (k=4)	58.80	111.9	15.00	420.1	0.00	645.4	55.40	57.7	16.60	160.1	**28.79**	528.6	**33.60**	3.0	**14.00**	3.1
		CoT-SC (n=3)	70.00	681.2	20.00	2243.2	**6.67**	2599.4	**57.20**	976.8	17.00	1790.5	25.76	1705.9	29.40	10.9	11.60	10.6
	Reasoning-aware Decoding	**TIDES**	**73.20**	244.8	**37.50**	790.3	0.00	1663.1	56.00	299.3	20.20	614.8	24.24	647.0	31.20	3.6	12.00	3.4
		TIDES (n=3)	**73.20**	733.9	**37.50**	2370.4	0.00	4908.3	56.40	900.0	19.20	1854.4	24.75	1957.9	31.20	10.9	12.00	10.2
Qwen2.5 Instruct (3B)	Baseline	ZS-CoT	81.80	224.9	35.00	772.1	**10.00**	1091.7	64.00	323.4	32.80	600.2	28.79	671.3	36.00	4.1	13.00	3.7
	Decoding	AdaTemp	83.60	218.1	42.50	850.3	6.67	1146.8	66.80	319.9	**36.40**	581.6	32.83	679.3	35.00	3.9	12.40	3.8
		CD (DoLa)	79.00	207.4	32.50	948.5	6.67	1628.4	61.40	336.8	28.20	604.6	25.25	787.7	31.60	3.7	9.80	3.5
	Reasoning	FS-CoT (k=4)	**84.80**	210.0	40.00	714.6	**10.00**	1264.1	60.20	133.5	28.00	370.5	31.31	751.0	**39.20**	3.2	**14.80**	3.1
		CoT-SC (n=3)	84.60	671.1	42.50	2384.5	**10.00**	3680.9	**67.60**	985.2	33.80	1740.5	31.82	1990.9	34.20	12.3	11.80	11.9
	Reasoning-aware Decoding	**TIDES**	84.40	219.8	**52.50**	911.1	6.67	1094.2	65.60	321.6	34.20	597.4	**33.33**	670.3	36.00	4.0	13.40	3.7
		TIDES (n=3)	84.20	659.1	47.50	2733.7	6.67	3254.3	66.20	964.0	34.00	1814.8	31.82	2021.5	36.00	12.0	13.40	11.1

Table 1: Overall evaluation of TIDES capability across different tasks. Each cell reports Accuracy (%) and average tokens generated. **Bold** indicates the best score and underline the second-best within each LLM group.

3 Experiment

3.1 Experimental Settings

Datasets. We evaluate on eight benchmarks across three groups. The **Math** group comprises GSM8K (Cobbe et al., 2021), AMC23 (MAA, 2023), and AIME24 (MAA, 2024). The **STEM** group comprises MMLU-STEM (Hendrycks et al., 2021), TheoremQA (Chen et al., 2023), and GPQA (Rein et al., 2024). The **Knowledge Recall** group comprises TriviaQA (Joshi et al., 2017) and PopQA (Mallen et al., 2023). For each benchmark, we evaluate on up to 500 test instances sampled with a fixed seed. Detailed dataset statistics and answer extraction protocols are provided in Appendix B.

Implementation Details. We evaluate Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct (Yang et al., 2024) with frozen weights. At inference, all models are run in fp16 precision with greedy decoding as the base. Detailed hyperparameter settings and generation budgets are provided in Appendix C.

Compared Methods. We compare against the following methods. **Zero-shot CoT (ZS-CoT)** (Kojima et al., 2023) serves as the baseline chain-of-thought method. **DoLa** (Chuang et al., 2024), an instantiation of Contrastive Decoding (Li et al., 2023), and **Adaptive Temperature (AdaTemp)** are decoding-side baselines that reshape the token distribution locally. **Few-shot CoT (FS-CoT)** (Wei et al., 2022) is a reasoning-side baseline that elicits reasoning chains via in-context exemplars. **Self-Consistency (CoT-SC)** (Wang et al., 2023) aggregates over multiple sampled reasoning paths via majority voting. **TIDES** denotes our full method, and **TIDES** $(n{=}3)$ runs three independent passes with majority voting, matching the sample count of CoT-SC $(n{=}3)$ for a controlled compute comparison. AdaTemp adapts the entropy-based temperature scaling of (Zhang et al., 2024) to operate within the same phase controller as TIDES, without BOS perturbation.

3.2 Main Results

As shown in Table 1, we report accuracy and average token generated across all eight benchmarks. We highlight three findings below.

Math. TIDES consistently improves over ZS-CoT on GSM8K and AMC23 across both model sizes, with the largest gains on AMC23 where latent capability exists but is not reliably elicited by greedy decoding. On AIME24, no method consistently improves over ZS-CoT, confirming that TIDES elicits latent capability rather than manufacturing missing capability.

STEM. TIDES remains competitive with the strongest baselines on MMLU-STEM and TheoremQA, and achieves the best accuracy on GPQA for the 3B model, while remaining competitive on the 1.5B model.

Knowledge Recall. TIDES avoids the overthinking regression observed in AdaTemp and CoT-SC, maintaining accuracy at ZS-CoT level while generating a comparable number of tokens.

4 Analysis

Benchmark-group Interpretation. The benchmark suite spans eight tasks across three groups, each testing a distinct hypothesis. The Math group (GSM8K, AMC23, AIME24) traces a difficulty ladder and the STEM group (MMLU-STEM, TheoremQA, GPQA) runs the same ladder outside arithmetic. GSM8K confirms that TIDES improves on problems within the model's latent capability

LLM	Concept	Method	GSM8K	AMC23	AIME24	MMLU	ThmQA	GPQA	TriviaQA	PopQA
Qwen2.5 Instruct (1.5B)	Baseline	ZS-CoT	2.0	6.7	13.1	2.3	3.2	6.2	0.0	0.0
	Decoding	AdaTemp	3.6	12.3	27.4	4.4	5.2	9.1	0.1	0.1
		CD (DoLa)	4.4	25.9	66.5	8.0	13.9	22.4	0.1	0.1
	Reasoning	FS-CoT (k=4)	1.1	5.2	10.7	1.1	2.7	7.1	0.0	0.0
		CoT-SC (n=3)	10.5	34.5	48.6	15.9	18.4	26.5	0.2	0.2
	Reasoning-aware Decoding	**TIDES**	3.7	11.7	24.3	4.5	5.1	9.9	0.1	0.1
		TIDES (n=3)	11.1	35.2	71.7	13.6	17.5	29.9	0.3	0.3
Qwen2.5 Instruct (3B)	Baseline	ZS-CoT	2.2	9.2	15.6	3.1	4.9	9.3	0.1	0.1
	Decoding	AdaTemp	4.1	18.4	24.1	6.5	7.1	13.0	0.1	0.1
		CD (DoLa)	6.3	27.9	47.6	9.8	13.0	22.1	0.2	0.2
	Reasoning	FS-CoT (k=4)	2.4	10.5	28.4	3.4	4.6	11.6	0.1	0.1
		CoT-SC (n=3)	14.5	43.3	81.1	20.0	21.9	41.3	0.3	0.3
	Reasoning-aware Decoding	**TIDES**	4.6	17.2	23.0	6.1	7.6	14.2	0.2	0.1
		TIDES (n=3)	13.7	51.3	68.1	18.2	20.8	42.7	0.5	0.4

Table 2: Average solving time (seconds per example) across tasks.

LLM	Benchmark	Parameter	Value	Acc. (%)	Time (s/ex)
Qwen2.5 Instruct (1.5B)	GSM8K	SNR	0.10	73.4	3.68
			0.25	73.2	3.70
			0.50	73.2	3.76
			1.00	73.6	3.69
			4.00	73.4	3.76
		Warm-up (k_{warm})	3	73.0	3.57
			5	73.4	3.62
			10	73.4	3.70
			15	73.0	3.78
			20	73.6	3.84
		BOS Update Period (k)	1	73.4	4.27
			3	73.8	4.0
			5	73.2	3.70
			10	73.0	3.46

Table 3: Hyperparameter sensitivity of TIDES on GSM8K (Qwen2.5-1.5B-Instruct). Shaded rows indicate default values. All three hyperparameters show broad insensitivity within practical ranges.

with no regression on easy instances, and AMC23 shows the effect persists at moderate difficulty. AIME24 and GPQA (1.5B) are capability-ceiling cases where TIDES cannot manufacture missing capability; conversely, TIDES achieves the best accuracy among all methods on GPQA (3B), confirming that entropy-guided exploration is effective when sufficient latent capability exists.

The Knowledge Recall group (TriviaQA, PopQA) tests not accuracy uplift but regression avoidance. AdaTemp and CoT-SC extend reasoning chains on short-answer tasks and occasionally push the model away from the correct brief answer, whereas TIDES avoids over-thinking regression on these tasks.

Decomposing Latency and Token Savings. TIDES adds a zeroth-order score query every k=5 autoregressive steps plus one BOS KV refresh, yielding approximately 1.8–2.0× greedy overhead in practice (Table 2). In terms of token usage, TIDES generates a comparable number of tokens to ZS-CoT while using approximately one-third the tokens of CoT-SC (n=3) across most benchmarks (Table 1), making it a favorable accuracy-efficiency tradeoff at a fraction of the latency and token cost. TIDES (n=3) further closes the gap on harder benchmarks at CoT-SC-comparable cost.

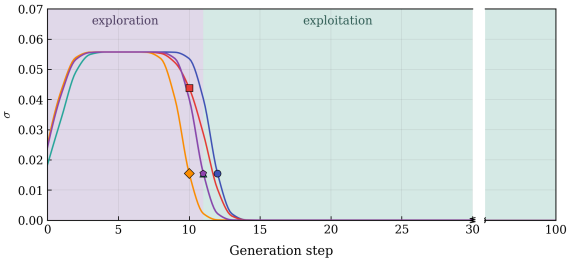

Figure 2: Phase transition trajectories for five GSM8K (Qwen2.5-1.5B) samples. Markers indicate the entropy-triggered transition from exploration to exploitation.

Hyperparameter Sensitivity. Table 3 reports a sensitivity analysis of **SNR**, **warm-up** (k_{warm}), and **BOS update period** (k) on GSM8K (Qwen2.5-1.5B). All three parameters show broad insensitivity within practical ranges: accuracy varies by at most 0.8 pp across the full SNR range, consistent with the single-scale framing that predicts model-invariant optimal SNR. Latency increases modestly with larger k_{warm} as the peak estimate stabilises later, and decreases with larger k at the cost of coarser context updates.

Phase Transition Trajectory. Figure 2 plots the σ schedule across five representative GSM8K samples. In all cases, σ ramps quickly to $\sigma_{\max}$ during the exploration phase, after which it decays geometrically to zero once per-step entropy drops below the half-peak threshold. The consistency of this pattern across samples confirms that the entropy-guided phase controller produces reliable exploration–exploitation transitions rather than arbitrary early or late commits. Trajectory visualizations across all benchmarks and both model scales are provided in Appendix D.

5 Conclusion

We proposed TIDES, a reasoning-aware decoding method that demonstrates explicit exploration-exploitation transition during inference. Evaluated on eight benchmarks with Qwen2.5-1.5B and 3B, TIDES consistently improves over zero-shot CoT on GSM8K and AMC23 where the model has latent but underutilized capacity. It remains competitive on STEM benchmarks and avoids over-thinking regression on knowledge-recall tasks, all at approximately 1.8× the cost of greedy decoding. These results show that TIDES can achieve explicit exploration-exploitation transition in small language models at inference time.

6 Limitations

Unfair few-shot comparison. TIDES is zero-shot; the few-shot CoT baseline uses in-context demonstrations unavailable to our method, making direct comparison unfair. Designing exploratory few-shot prompts remains future work.

Limited model scale. All experiments are conducted on SLMs up to 3B parameters. Whether the entropy-guided phase controller produces consistent exploration–exploitation transitions at larger scales remains unverified, as the entropy landscape and BOS embedding geometry may differ substantially in models with more parameters or trained on different data mixtures. Generalization to larger models remains unverified.

Capability ceiling. TIDES elicits latent capability by providing more reliable exploration during generation, but it cannot manufacture knowledge or reasoning capacity that is entirely absent from the base model. This is evidenced by near-zero accuracy on AIME24 for the 1.5B model, where no inference-time method achieves meaningful improvement. In such capability-ceiling regimes, the entropy signal may still trigger phase transitions, but the exploited distribution remains concentrated on incorrect reasoning paths, and additional inference compute yields no benefit.

Knowledge ceiling. TIDES does not augment the model's parametric knowledge and therefore cannot recover factual answers that fall outside the model's training distribution. On knowledge-recall benchmarks such as TriviaQA and PopQA, TIDES neither improves nor degrades accuracy relative to ZS-CoT, which we interpret positively: by committing early when entropy peaks are low, TIDES avoids the over-thinking regression observed in AdaTemp and CoT-SC, where extended generation chains occasionally suppress a correct brief answer in favour of a spurious elaboration.

σ schedule design space. The current σ schedule is inspired by warmup-based optimizers. Alternative schedules have not been fully explored. Systematic exploration of alternative schedules, including their interaction with the commit threshold and BOS update period, is left for future work.

References

Albert S. Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. 2022. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, 22:507–560.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023. TheoremQA: A theorem-driven question answering dataset. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. DoLa: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*.

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

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. 2012. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

MAA. 2023. AMC 2023. Hugging Face Datasets, `https://huggingface.co/datasets/math-ai/amc23`. Curated by math-ai.

MAA. 2024. AIME 2024. Hugging Face Datasets, https://huggingface.co/datasets/HuggingFaceH4/aime_2024. Curated by HuggingFaceH4.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Yurii Nesterov and Vladimir Spokoiny. 2017. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Gareth O. Roberts and Richard L. Tweedie. 1996. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

James C. Spall. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Max Welling and Yee Whye Teh. 2011. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 681–688, Bellevue, WA, USA. Omnipress.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Shimao Zhang, Yu Bao, and Shujian Huang. 2024. EDT: Improving large language models' generation by entropy-based dynamic temperature sampling. *arXiv preprint arXiv:2403.14541*.

A Key Concepts

We summarize the three key concepts underlying TIDES.

- **Entropy-guided Phase Transition**: A method in which per-step token entropy is used as a commit signal to explicitly mark the boundary between exploration and exploitation during generation.
- **Temporal Context Diffusion**: An approach in which noising and denoising processes are applied along the token sequence as a temporal axis, where BOS embedding perturbation serves as the global context shift mechanism that induces progressively increasing noise during exploration and decreasing noise during exploitation.
- **Reasoning-aware Decoding**: A concept for designing the decoding process to perform explicit exploration–exploitation transition during generation. In this work, we realize this concept as a training-free inference method via entropy-guided BOS perturbation.

B Benchmark Details

Table 4 summarizes the benchmark statistics used in our experiments. For benchmarks larger than 500, we evaluate on a deterministic random subset (seed = 42); smaller benchmarks are used in full.

C Implementation Details

Table 5 shows the hyperparameters used for TIDES.

D Trajectory Visualizations Across Benchmarks

Figures 3 and 4 extend the trajectory plot in the main paper to all benchmarks and both model scales. Across datasets, the entropy-guided controller exhibits the same exploration–exploitation pattern: σ increases during the initial exploration phase and decays after the entropy-triggered commit point.

Notably, the trajectory patterns provide a qualitative explanation for the benchmark-level performance differences observed in Table 1. On GSM8K and AMC23, σ ramps to a high peak and sustains exploration over more generation steps before committing, reflecting higher per-step entropy and greater uncertainty in the reasoning path; this extended exploration is associated with the largest

Dataset	Category	Task Type	#Test (full)	#Eval (used)	Max New Tokens	Metric
GSM8K	Math	Free Form	1,319	500	512	Exact match (numeric)
AMC23	Math	Free Form	40	40	2048	Exact match (numeric)
AIME24	Math	Free Form	30	30	4096	Exact match (numeric)
MMLU-STEM	STEM	MCQ	3,153	500	512	MC letter extraction
TheoremQA	STEM	Free Form	747	500	2048	Normalized exact match
GPQA	STEM	MCQ	198	198	2048	MC letter extraction
TriviaQA	Know.	Free Form	18,669	500	64	Normalized exact match
PopQA	Know.	Free Form	14,267	500	64	Normalized exact match

Table 4: Benchmark statistics used in our experiments. For TriviaQA and PopQA, 500 instances are randomly sampled with a fixed seed from the full evaluation set.

Hyperparameter	1.5B	3B
Base decoding	Greedy	
Inference precision	`fp16`	
Frozen weights	✓	
SNR	0.25	
Warmup length (k_{warm})	10	
BOS update period (k)	5	
$\sigma_{\max}$	$\|e_{\text{BOS}}\|/\sqrt{d} \cdot \text{SNR}$	
Commit threshold	$\frac{1}{2} H_{\text{peak,warmup}}$	
σ decay	Geometric	
Gradient clip ($\|\widehat{\nabla H_\sigma}\|$)	≤ 1	
CoT-SC / TIDES (n)	3	
Seed	42	

Table 5: TIDES hyperparameter configuration used in all experiments.

accuracy gains over ZS-CoT. In contrast, on TriviaQA and PopQA, the entropy peak is low and the commit point is reached within only a few steps, indicating that the model rapidly converges to a high-confidence answer. In these cases, TIDES commits early and generates a comparable number of tokens to ZS-CoT, explaining why accuracy is maintained at baseline level rather than regressed as observed in AdaTemp and CoT-SC.

E Few-shot Prompts

The few-shot samples used in our experiments are listed in Tables 6–12.

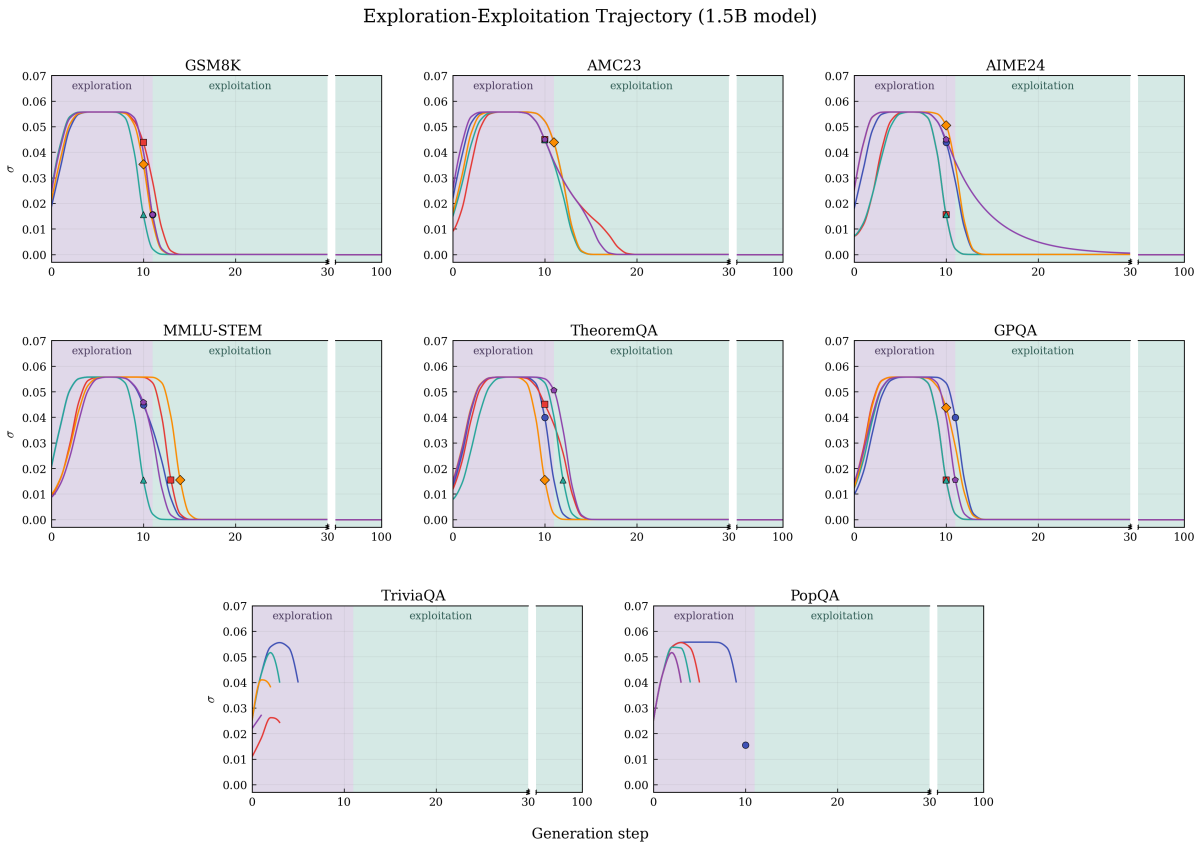

Figure 3: Phase transition trajectories across all benchmarks for Qwen2.5-1.5B-Instruct. Each panel shows the σ schedule and entropy-triggered transition points.

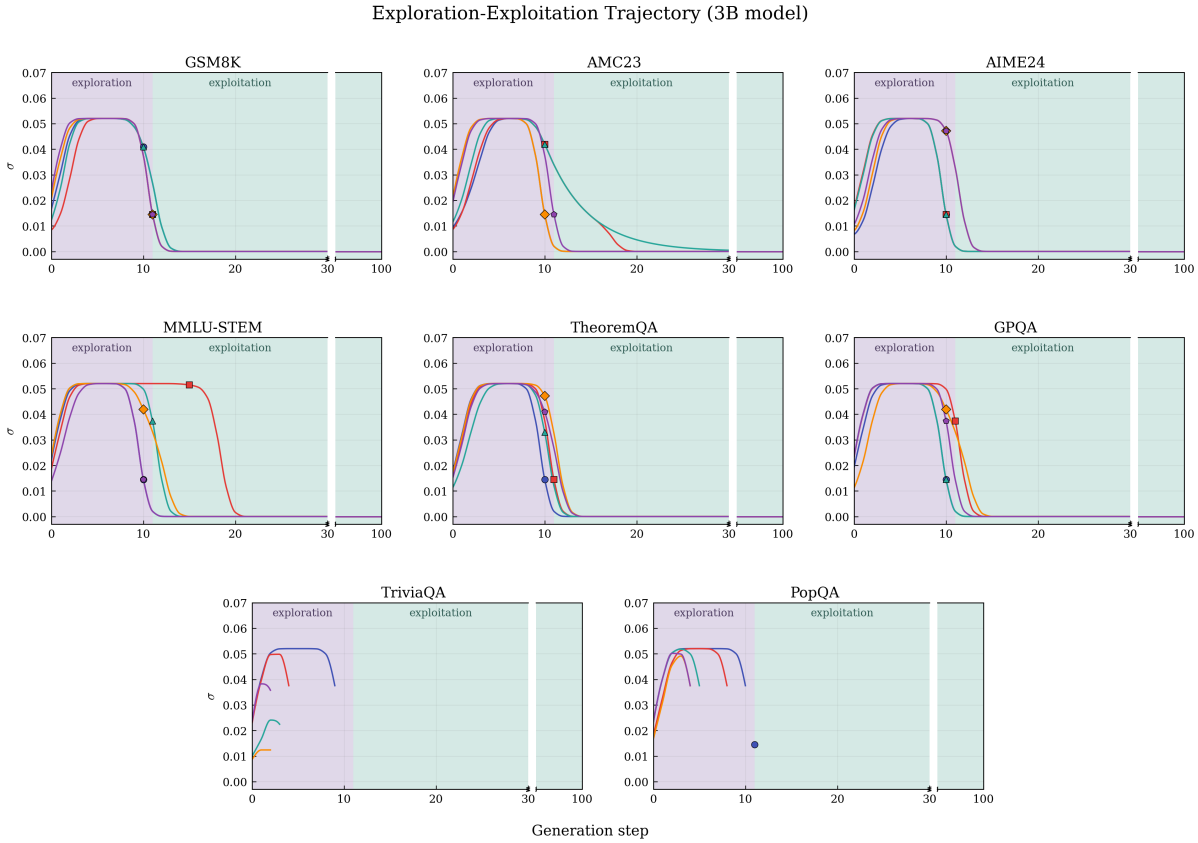

Figure 4: Phase transition trajectories across all benchmarks for Qwen2.5-3B-Instruct. Each panel shows the σ schedule and entropy-triggered transition points.

System.

Solve the math problem step by step. End your final answer with `#### <number>`.

Demonstrations (1–8 of 8).

Q1. There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A1. There are 15 trees originally. Then there were 21 trees after the workers planted some more. So they must have planted $21 - 15 = 6$ trees. The answer is $\boxed{6}$.

Q2. If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A2. There are originally 3 cars. 2 more cars arrive, so now there are $3 + 2 = 5$ cars. The answer is $\boxed{5}$.

Q3. Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A3. Originally, Leah had 32 chocolates and her sister had 42, so in total $32 + 42 = 74$. After eating 35, they have $74 - 35 = 39$ pieces left. The answer is $\boxed{39}$.

Q4. Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A4. Jason had 20 lollipops originally and now has 12. He gave Denny $20 - 12 = 8$ lollipops. The answer is $\boxed{8}$.

Q5. Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A5. He had 5 toys to start with. He got 2 from mom and 2 from dad, so $2 + 2 = 4$ new toys. In total he has $5 + 4 = 9$ toys. The answer is $\boxed{9}$.

Q6. There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A6. There are 4 days from Monday to Thursday. 5 computers were installed each day, so $5 \times 4 = 20$ new computers. Starting from 9, total $= 9 + 20 = 29$. The answer is $\boxed{29}$.

Q7. Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A7. Michael started with 58. After Tuesday he had $58 - 23 = 35$. After Wednesday he had $35 - 2 = 33$. The answer is $\boxed{33}$.

Q8. Olivia has \$23. She bought five bagels for \$3 each. How much money does she have left?
A8. Five bagels at \$3 each cost $5 \times 3 = \$15$. Starting from \$23, she has $23 - 15 = \$8$ left. The answer is $\boxed{8}$.

Test query.

{*question*}

Table 6: GSM8K — 8-shot

System.

Solve the math problem step by step. End your final answer with `#### <number>`.

Demonstrations (1–4 of 4).

Q1. How many ways are there to split the integers 1 through 14 into 7 pairs such that in each pair, the greater number is at least 2 times the lesser number?
A1. The integers 8–14 must each be the greater element, and 1–7 must be the lesser elements. Since 7 can only pair with 14, 7 pairs with 14. Then 6 has 2 choices, 5 has 3, 4 has 4, 3 has 3, 2 has 2, 1 has 1. Total $= 2 3 4 3 2 1 =$ `#### 144`.

Q2. Let $\mathcal{R}$ be the region in the complex plane consisting of all $z = z_1 + z_2$, where z_1 lies on the segment with endpoints 3 and $4i$, and $|z_2| \leq 1$. What integer is closest to the area of $\mathcal{R}$?
A2. The segment from 3 to $4i$ has length $\sqrt{3^2 + 4^2} = 5$. The region is its Minkowski sum with a unit disk: a 5×2 rectangle (area 10) plus two unit semicircles (area π). Total ≈ 13.14, closest integer `#### 13`.

Q3. What is the value of $(\log 5)^3 + (\log 20)^3 + (\log 8)(\log 0.25)$, where log denotes the base-ten logarithm?
A3. Let $a = \log 2$. Then $\log 5 = 1 - a$, $\log 20 = 1 + a$, $\log 8 = 3a$, $\log 0.25 = -2a$. $(1-a)^3 + (1+a)^3 = 2 + 6a^2$ and $(3a)(-2a) = -6a^2$. Total $= 2$ `#### 2`.

Q4. The roots of $10x^3 - 39x^2 + 29x - 6$ are the height, length, and width of a box. A new box is formed by lengthening each edge by 2. What is its volume?
A4. With roots a, b, c, $f(x) = 10(x - a)(x - b)(x - c)$. New volume $= (a + 2)(b + 2)(c + 2) = -f(-2)/10$. $f(-2) = -300$, so volume $= 300/10 = 30$. `#### 30`.

Test query.

{*question*}

Table 7: AIME24 / AMC23 — 4-shot

User-only message.

Preamble: `Here are some example questions from experts. An explanation is given before the final answer. Answer the final question yourself, giving your reasoning beforehand.`

Demonstrations (1–5 of 5; abbreviated).

Q1 (genetics). In a population, 1/400 individuals carry a recessive cancer allele b. Under Hardy–Weinberg, what fraction carry b but do not develop cancer?
A1. Let's think step by step: $q^2 = 1/400 \Rightarrow q = 0.05$, $p = 0.95$. Heterozygotes $= 2pq = 38/400$. **The correct answer is (D)**.

Q2 (chemistry titration). A 0.056 g Fe pellet is dissolved in 10 mL of 0.1 M HBr and titrated with 0.02 M $KMnO_4$. How many equivalence points?
A2. Let's think step by step: MnO_4^- oxidises Fe^{2+} first ($E^0 = 0.77$ V) then Br^- ($E^0 = 1.09$ V). Stoichiometry gives the first equivalence point at 10 mL and, after Fe^{2+} is exhausted, the second at 35 mL. **The correct answer is (A)**.

Q3 (physics, 3-D SHO). Energy and degeneracy of the third excited state of a 3-D isotropic harmonic oscillator?
A3. Let's think step by step: $E_n = (n + 3/2)\hbar\omega$ gives $E_3 = (9/2)\hbar\omega$; degeneracy $(n + 1)(n + 2)/2 = 10$. **The correct answer is (B)**.

Q4 (organic chemistry). "My compounds are on top of each other" — what does the chemist mean?
A4. Let's think step by step: this is TLC slang for similar R_f, i.e. similar polarities. **The correct answer is (A)**.

Q5 (cell biology). Which mitochondrial-protein mutation is *least* likely to be transmitted father→child?
A5. Let's think step by step: paternal mitochondria are not inherited; NADH dehydrogenase 2 is encoded by mtDNA (MT-ND2), the others by nuclear DNA. **The correct answer is (C)**.

Test query + final instruction.

Question: {*question*}
Choices: {*A–D*}
`Give step by step reasoning before you answer, and when you're ready to answer, please use the format "The correct answer is (insert answer here)":`

Table 8: GPQA — 5-shot

User template (applied to each demo and to the test query; {*subject*} is the humanized subject name, e.g. "college physics"):

```
The following is a multiple choice question about {subject}.
{question}
Choices:
(A) {a}
(B) {b}
(C) {c}
(D) {d}
Format your response as: "The correct answer is (X)" where X is one of A/B/C/D.
Let's think step by step.
```

Assistant target for each demo:

`The correct answer is (X).` (where X is the gold letter for that demo.)

Demo source.

Five demos per subject, drawn deterministically from `cais/mmlu`'s `dev` split in dataset order (*not* resampled; seed 42 controls only the test-pool shuffle). For each test example we look up its subject and prepend the five matching dev demos.

Example (subject = abstract algebra; first dev demo).

Q. Find all $c \in \mathbb{Z}_3$ such that $\mathbb{Z}_3[x]/(x^2 + c)$ is a field.
Choices: (A) 0 (B) 1 (C) 2 (D) 3
A. The correct answer is (B).

Table 9: MMLU-STEM — 5-shot (subject-matched)

System.

You are a STEM expert. Solve the problem step by step, applying the relevant theorem. Conclude with a line of the exact form "The answer is X", where X is the final answer only — a number, True or False, an option letter, or a list like `[1, 2, 3]`.

Demonstrations (1–5 of 5).

For each demo d:

User: `Problem:` {*d.question*}

Assistant: `The answer is` {*d.answer*}

Demo source.

Indices $n, \ldots, n+4$ of the shuffled (seed 42) text-only `TIGER-Lab/TheoremQA` test pool, where n is the eval-pool size. Identical demos are reused across all eval examples.

Test query.

`Problem:` {*question*}

Table 10: TheoremQA — 5-shot

System.

Answer the question with a short factual answer only — no explanation, no preamble. Just the answer.

Demonstrations (1–5 of 5).

For each demo d:

User: `Question:` {*d.question*}
`Answer:`
Assistant: {*d.first gold alias*}

Demo source.

`train` split of `trivia_qa/rc.nocontext`. We take the first 200 rows, shuffle them with seed 42, and keep the first five. Demos are fixed across all eval examples.

Test query.

`Question:` {*question*}
`Answer:`

Table 11: TriviaQA — 5-shot

System.

Answer the question with a short factual answer only — no explanation, no preamble. Just the answer.

Demonstrations (1–5 of 5).

For each demo d:

User: `Question:` {*d.question*}
`Answer:`
Assistant: {*d.first gold alias*}

Demo source.

We use the `test` split of `akariasai/PopQA`. After shuffling with seed 42, we take indices $n, \ldots, n+199$, where n is the eval-pool size, discard rows with empty `possible_answers`, shuffle the remaining candidates again with seed 42, and keep the first five. Demos are fixed across all eval examples.

Test query.

`Question:` {*question*}
`Answer:`

Table 12: PopQA — 5-shot