NORMCAST: EFFECTIVE LATENT-SPACE REASONING VIA SOFT EMBEDDING NORM CALIBRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent latent reasoning approaches enable large language models (LLMs) to reason in a continuous space and improve reasoning performance. As a representative method, Soft Thinking utilizes the probability-weighted average of candidate token embeddings as next input embedding. However, we identify that such averaging operation results in norm shrinkage at multi-candidate positions, where Soft Thinking departs from standard Chain-of-Thought (CoT) reasoning. Further analysis reveals that the resulting embeddings fall below the typical norm range of discrete embeddings, creating an input-scale mismatch that limits performance. Motivated by these observations, we propose **NormCaST** (Norm-Calibrated Soft Thinking), the first approach to analyze and optimize latent-space reasoning from the perspective of representation magnitude. Specifically, NormCaST calibrates each soft embedding to the probability-weighted average norm of its candidate token embeddings, restoring the typical input representation scale. This training-free calibration introduces no substantial computational overhead and serves as a plug-and-play enhancement for existing Soft Thinking variants. Extensive experiments across four models ranging from 1.7B to 32B parameters and multiple reasoning benchmarks demonstrate that NormCaST consistently outperforms its non-calibrated counterparts, improving average Pass@1 by up to 2.97 points while maintaining comparable generation length.

1 INTRODUCTION

Large language models (LLMs) achieve impressive reasoning performance through explicit Chain-of-Thought (CoT) (Wei et al., 2022; Feng et al., 2023; Yao et al., 2023; Goyal et al., 2024), which guides models to produce step-by-step intermediate reasoning in natural language. However, CoT reasoning is fundamentally constrained to discrete token sequences, inherently limiting the model's ability to represent abstract concepts and explore multiple reasoning paths simultaneously without committing to a single token at each generation step.

To overcome this bottleneck, recent work has shifted reasoning from discrete tokens into latent spaces, representing thoughts as continuous or distributional embeddings rather than natural-language sequences (Hao et al., 2024; Cheng & Van Durme, 2024; Shen et al., 2025; Tan et al., 2025; Xu et al., 2025a). As a representative training-free approach, Soft Thinking (Zhang et al., 2025) replaces discrete token sampling by feeding probability-weighted averages of token embeddings as subsequent inputs. By preserving distributional information, this formulation enables the model to reason beyond a single sampled token and theoretically enables implicit multi-path exploration in continuous space.

However, the promise of Soft Thinking has not fully materialized in practice. Wu et al. (2026) demonstrate that probability mass at most reasoning steps concentrates on the top-1 candidate, collapsing multi-path exploration to near-greedy decoding—a phenomenon they term the greedy pitfall. By introducing Gumbel-Softmax noise, this work partially restores multi-path exploration and yields considerable accuracy gains. Orthogonal improvements on discrete-soft switching, including SwiReasoning (Shi et al., 2026), SeLaR (Fu & Luo, 2026), and ThinkRouter (Xu et al., 2026), optimize when to invoke soft reasoning. However, an unsolved question remains: what limits Soft Thinking at the specific positions where it actually departs from standard CoT?

In this work, we identify a previously overlooked geometric phenomenon that limits the effectiveness of Soft Thinking: embedding norm shrinkage occurs at positions where multiple candi-

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

date token embeddings are combined and Soft Thinking departs from standard CoT. Since soft embeddings are convex combinations of candidate token embeddings, Jensen's inequality gives $\| \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] E[k] \|_2 \leq \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] \| E[k] \|_2$, with equality only when all positively weighted candidate embeddings share the same direction. At single-candidate positions, the soft embedding reduces to the only candidate token embedding and its norm remains unchanged. At multi-candidate positions, differences in candidate embedding directions generally make the inequality strict. Thus, the norm shrinkage is concentrated precisely where Soft Thinking forms genuinely continuous representations.

We further demonstrate that shrunk soft embeddings fall into a low-norm region rarely encountered in discrete tokens. On Qwen3 models performing English STEM reasoning, candidate token embeddings exhibit L2 norms that rarely fall below a model-specific threshold. Vocabulary entries below this threshold are disproportionately enriched for non-Latin or mixed scripts, symbols, and special or byte tokens, which are uncommon in this context. At multi-candidate positions, a substantial fraction of soft embeddings fall below the threshold, inducing a distributional mismatch with the input representations the model typically encounters during normal reasoning. Moreover, such misaligned embeddings could enter the unnormalized residual stream and propagate through layers, therefore affecting reasoning performance.

Motivated by these insights, we propose **NormCaST** (Norm-Calibrated Soft Thinking), a new training-free latent reasoning mechanism that rescales each soft embedding to match the probability-weighted average norm of its candidate tokens:

$$L_{\text{target}} = \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] \cdot \| E[k] \|_2, \qquad \tilde{e}_{t,\text{cal}} = \tilde{e}_t \cdot \frac{L_{\text{target}}}{\| \tilde{e}_t \|_2 + \epsilon}. \tag{1}$$

This single-line calibration restores the expected representation scale while preserving the direction and candidate-weighted composition of the original soft embedding. It introduces no learnable parameters, additional forward passes, or substantial computational overhead. Because it is applied after the soft embedding is formed, NormCaST leaves the underlying sampling, stopping, and switching mechanisms unchanged and can be integrated with existing Soft Thinking variants.

Our contributions in this work are:

1. We identify and characterize a previously overlooked geometric limitation of Soft Thinking: norm shrinkage selectively affects the critical multi-candidate positions where Soft Thinking departs from CoT and pushes some soft embeddings into out-of-distribution, low-norm regions that the model rarely enters when reasoning with discrete tokens.
2. We propose NormCaST, a training-free mechanism that restores each soft embedding to the probability-weighted average norm of its candidate token embeddings while preserving its direction. It is applied at every soft step while naturally leaving single-candidate embeddings unchanged. To the best of our knowledge, NormCaST is the first approach to analyze and optimize latent reasoning from the perspective of representation magnitude.
3. We conduct extensive validation of NormCaST on mathematics, science, and coding domains across multiple benchmarks, model families, and scales, demonstrating consistent accuracy gains over its uncalibrated counterparts while maintaining comparable generation length.

2 Preliminaries

2.1 Reasoning in Discrete Token Space

Let V be the model's vocabulary of size $|V|$, and let $E \in \mathbb{R}^{|V| \times d}$ denote the input embedding matrix, where $E[i] \in \mathbb{R}^d$ is the embedding of token $i \in V$. At each CoT reasoning step t, the language model produces logits before temperature scaling, conditioned on the input query $x_{1:Q}$ and the previously generated reasoning tokens $r_{1:t-1}$:

$$\ell_t = \text{LLM}\big(E[x_{1:Q}],\, E[r_{1:t-1}]\big). \tag{2}$$

The logits are converted into the raw and temperature-scaled distributions, after which the standard sampling pipeline applies top-k, top-p, and min-p filtering and renormalization:

$$p_t^{\text{raw}} = \text{softmax}(\ell_t), \qquad p_t^{\tau} = \text{softmax}(\ell_t / \tau), \qquad p_t = \text{Sample}(p_t^{\tau}). \tag{3}$$

Standard chain-of-thought (CoT) reasoning (Wei et al., 2022) samples a discrete token $r_t \sim p_t$ and feeds its embedding $E[r_t]$ as the next-step input, collapsing full distribution into a single token.

2.2 Reasoning in Continuous Space with Soft Embeddings

Soft Thinking (Zhang et al., 2025) replaces discrete token sampling with continuous interpolation in the embedding space. Instead of committing to a single token, the method computes a probability-weighted convex combination of token embeddings as the next input, defined as follows.

At each Soft Thinking step t, the model instead conditions on the preceding soft embeddings and likewise produces logits before temperature scaling:

$$\ell_t = \mathrm{LLM}\big(E[x_{1:Q}],\, \tilde{e}_{1:t-1}\big). \tag{4}$$

The same transformations in Eq. 3 yield p_t^{raw}, p_t^{τ}, and the filtered and renormalized distribution p_t. From this filtered distribution, Soft Thinking performs top-m candidate selection, where m specifies the maximum number of candidate tokens retained.

$$\mathcal{C}_t = \text{Top-M}(p_t, m). \tag{5}$$

Because filtering may assign zero mass to some entries among these m tokens, we define the effective candidate subset and its size as

$$\mathcal{C}_t^{+} = \{k \in \text{Top-M}(p_t, m) \mid p_t[k] > 0\}\,, \qquad n = |\mathcal{C}_t^{+}|. \tag{6}$$

The retained probabilities are renormalized over $\mathcal{C}_t$ and used to compute the probability-weighted average of the corresponding token embeddings. The sequence is updated as $\tilde{e}_{1:t} := \tilde{e}_{1:t-1} \| \tilde{e}_t$.

$$\tilde{p}_t[k] = \frac{p_t[k]}{\sum_{v \in \mathcal{C}_t} p_t[v]}, \qquad \tilde{e}_t = \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] \cdot E[k] \in \mathbb{R}^d. \tag{7}$$

The Cold Stop mechanism monitors the entropy of p_t and terminates soft reasoning when the entropy remains below a threshold δ for l consecutive steps, indicating sustained high confidence. After reasoning terminates, the model generates the final answer using standard discrete decoding.

Unlike discrete reasoning, which collapses the probability mass onto a single token and commits to one explicit reasoning path, Soft Thinking integrates multiple candidates into a continuous representation. When several tokens carry non-negligible probability, the soft embedding blends their semantic directions, enabling implicit exploration of multiple reasoning paths in parallel.

3 Why Soft Thinking Underperforms: Norm Shrinkage Selectively Hits Critical Positions

This section investigates the fundamental limitations of Soft Thinking. We observe that embedding norm shrinkage occurs selectively at the minority of positions where Soft Thinking departs from standard CoT and provides value over discrete decoding. At these positions, two or more effective candidate embeddings are convexly combined into a new embedding whose reduced norm pushes it into an out-of-distribution low-norm region that the model rarely visits during standard reasoning.

All diagnostic experiments are conducted on Qwen3 models using the recommended decoding configuration (temperature $\tau = 0.6$, top-$p = 0.95$, top-$k = 20$) and with Soft Thinking activated following its official implementation. Unless otherwise stated, all token-level statistics are collected from reasoning trajectories on the GPQA-Diamond benchmark (Rein et al., 2023).

3.1 Probability Concentration and the Minority of Critical Positions

Prior work (Wu et al., 2026) demonstrates that most soft steps in Soft Thinking are dominated by a single high-probability continuation, collapsing multi-path exploration to near-greedy decoding. We recapitulate this finding here to introduce a central concept: the minority of critical positions where Soft Thinking actually differs from discrete CoT.

Probability concentration is intrinsic. We measure mean top-1 mass at the three stages defined in Section 2. On GPQA-Diamond, p_t^{raw} reaches 0.918 for Qwen3-1.7B and 0.875 for Qwen3-8B; p_t^{τ} raises these values to 0.952 and 0.925, and $\tilde{p}_t$ to 0.957 and 0.940. Thus, concentration is intrinsic to the raw output, while subsequent stages only amplify it. Detailed results are provided in Appendix C.1.

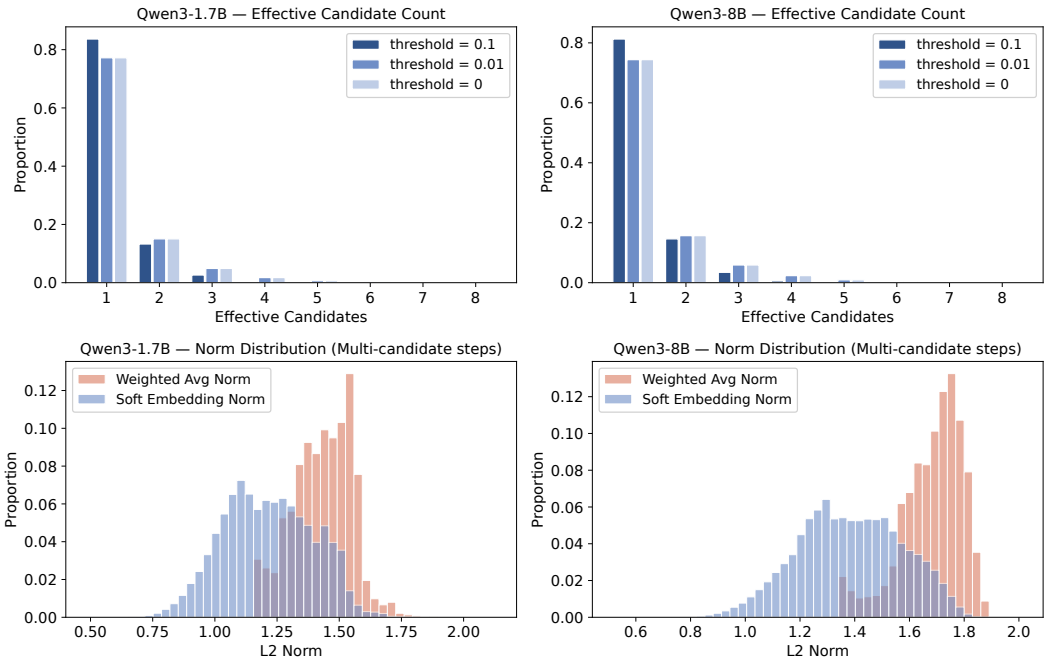

Figure 1: **Top row**: Effective candidate count per reasoning step on GPQA-Diamond, showing over 74% single-candidate dominance at all thresholds. **Bottom row**: Distribution of soft embedding L2 norms vs. weighted average norms of candidate embeddings, restricted to multi-candidate steps (≥ 2 candidates). A systematic leftward shift of soft embedding norms is evident in both models.

The minority matters. Figure 1 (top row) illustrates the distribution of effective candidate counts per reasoning step on GPQA-Diamond for Qwen3-1.7B and 8B. Following Eq. 6, an effective candidate is a token in $\mathcal{C}_t$ with positive filtered probability, and the effective candidate count is $n = |\mathcal{C}_t^+|$. We also report thresholds 0.1 and 0.01 as complementary views of how probability mass concentrates among the retained candidates, while all subsequent definitions of single- and multi-candidate positions use the canonical positive-probability threshold. Under this definition, over 74% of reasoning steps are dominated by a single candidate. When multiple candidates do exist, they are typically limited to 2 or 3, while instances of ≥ 4 candidates are notably rare. Analogous evaluations on GSM8K yield consistent distributions (see Appendix C.2).

Since Soft Thinking reduces exactly to standard CoT at single-candidate positions, its potential to outperform standard decoding relies entirely on the minority of multi-candidate positions. Through theoretical and empirical analyses, we reveal that at these critical positions, the norm shrinkage of soft embeddings emerges as a key factor limiting Soft Thinking's performance.

3.2 Theoretical and Empirical Evidence of Norm Shrinkage

Theoretical basis. Consider the soft embedding computation in Eq. 7, where the convex combination coefficients satisfy $\tilde{p}_t[k] \geq 0$ and $\sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] = 1$. Since the L2 norm is a convex function, by Jensen's inequality, we obtain:

$$\left\| \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] \cdot E[k] \right\|_2 \leq \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] \, \|E[k]\|_2. \tag{8}$$

Equality holds if and only if all candidate embeddings $E[k]$ with positive weight share the same direction. For single-candidate positions ($n = 1$), the soft embedding equals the top-1 token embedding and no norm reduction occurs. For multi-candidate positions ($n \geq 2$), the soft embedding norm is strictly smaller than the probability-weighted average norm of its constituent embeddings, since different tokens almost never share identical directions in high-dimensional embedding spaces.

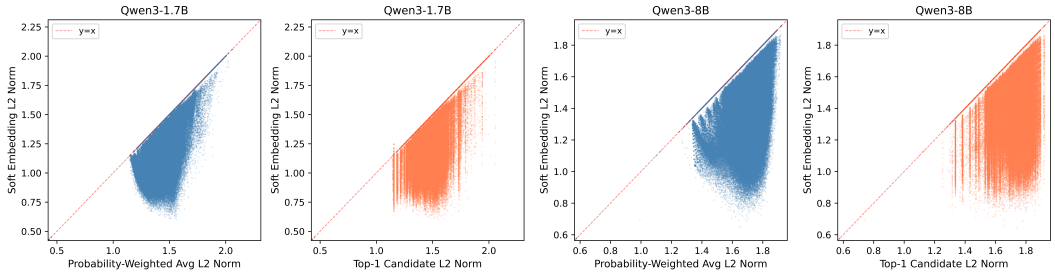

Figure 2: Per-step scatter plots on GPQA-Diamond. The first and third panels plot the soft embedding L2 norm against the probability-weighted average norm for Qwen3-1.7B and 8B, respectively; all points falling on or below the diagonal directly verify norm shrinkage. The second and fourth panels compare the soft embedding norm against the top-1 candidate embedding norm; nearly all points lie on or below the reference line, indicating $\|\tilde{e}_t\|_2 \leq \|E[k_{\text{top-1}}]\|_2$ almost everywhere.

Empirical evidence. The bottom row of Figure 1 provides direct evidence on GPQA-Diamond. Restricting to multi-candidate steps, we overlay the L2 norm distribution of soft embeddings with the probability-weighted average norm of the corresponding candidate embeddings. In both models, the soft embedding norm distribution exhibits an obvious leftward shift relative to the weighted average norm distribution. Notably, this distributional gap is less evident when aggregating across all reasoning steps (see Appendix C.3), as the shrinkage effect is masked by the $\sim$75–80% of single-candidate steps where the norm remains unchanged.

Figure 2 examines this effect at the individual step level through scatter plots. Comparing the soft embedding norm against the probability-weighted average norm, all points fall on or below the $y = x$ reference line, quantifying the exact norm deficit induced by the convex combination. Furthermore, while Eq. 8 only guarantees that $\|\tilde{e}_t\|_2 \leq \sum \tilde{p}_t[k]\|E[k]\|_2$, our empirical analysis reveals that $\|\tilde{e}_t\|_2$ is almost universally bounded by $\|E[k_{\text{top-1}}]\|_2$ as well. As shown in the scatter plots comparing soft embedding norm against the top-1 candidate embedding norm, nearly all points lie on or below the diagonal. In effect, interpolating multiple candidates consistently degrades the representation scale of the soft embedding compared with discrete tokens.

3.3 Why Norm Shrinkage Can Harm Reasoning

A norm threshold during Soft Thinking reasoning. During Soft Thinking reasoning on English STEM benchmarks, candidate token embeddings at each step exhibit L2 norms that rarely fall below a characteristic threshold: approximately 1.156 for Qwen3-1.7B and 1.336 for Qwen3-8B. Tokens with norms below this threshold are almost never selected, causing the probability-weighted average norm of candidates to inherit the same threshold. This phenomenon is visible in the bottom row of Figure 1 and across all four panels of Figure 2.

The low-norm region has a distinct vocabulary composition. Vocabulary entries below this threshold are disproportionately enriched for non-Latin scripts, other or mixed scripts, symbols, and special/byte tokens—categories uncommon in English STEM reasoning. Their composition differs markedly from the norm region occupied by common English reasoning candidates. Together with the fact that ordinary top-1 candidates rarely occupy this range, this difference indicates that a soft embedding pushed into the low-norm region has an out-of-distribution magnitude misaligned with the model's expected input distribution. A detailed analysis is provided in Appendix C.4.

Soft embedding norms fall below the threshold. As established in Section 3.2, soft embedding norms at multi-candidate positions are systematically suppressed by the convex combination. The bottom row of Figure 1 shows that a substantial fraction of soft embeddings fall below the characteristic norm threshold, into the region where the model virtually never selects tokens in this task context. These soft embeddings thus deviate from the norm range occupied by discrete tokens typically selected during English STEM reasoning.

Norm reduction propagates through the unnormalized residual stream. The input embedding in modern Transformer architectures enters the residual stream before any block-level normalization.

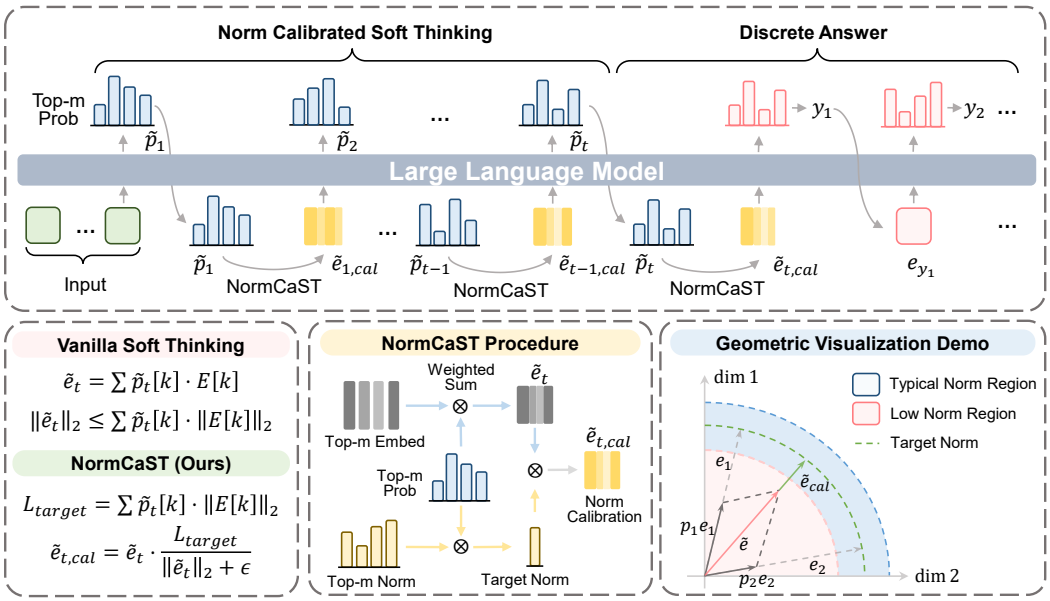

Figure 3: Overview of NormCaST. The method applies a simple rescaling to the soft embedding $\tilde{e}_t$, adjusting its L2 norm to match the probability-weighted average norm of the candidate token embeddings. This calibration pushes $\tilde{e}_t$ back into the typical norm range, restoring its semantic expressiveness and improving reasoning performance.

Although each Transformer block applies RMSNorm to its input, the residual pathway that carries the embedding into subsequent layers is unnormalized (Zhang & Sennrich, 2019; Xiong et al., 2020). A reduction in input embedding norm therefore persists in the residual pathway and can affect subsequent layers.

4 Method: NormCaST

Our analysis establishes that Soft Thinking exhibits systematic norm shrinkage selectively at critical positions. To correct this geometric mismatch, we propose NormCaST (Norm-Calibrated Soft Thinking), a simple and training-free rescaling operation. The method introduces no tunable hyperparameters and is applied at every Soft Thinking step. At single-candidate positions, the calibration factor reduces to 1 and the embedding naturally remains unchanged.

4.1 Calibration Procedure

At each step t, Soft Thinking constructs the candidate set $\mathcal{C}_t$, renormalizes the retained probabilities as $\tilde{p}_t$, and forms the soft embedding $\tilde{e}_t$ following Eq. 7. NormCaST preserves the direction of this embedding while restoring its magnitude to the probability-weighted average norm of the candidate embeddings. Specifically, it computes the target norm and calibrated embedding as

$$L_{\text{target}} = \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] \cdot \|E[k]\|_2, \qquad \tilde{e}_{t,\text{cal}} = \tilde{e}_t \cdot \frac{L_{\text{target}}}{\|\tilde{e}_t\|_2 + \epsilon} \tag{9}$$

where $\epsilon = 10^{-8}$ ensures numerical stability. The calibrated embedding $\tilde{e}_{t,\text{cal}}$ then replaces $\tilde{e}_t$ in the subsequent forward pass. Notably, when $n = 1$, we have $\tilde{e}_t = E[k]$ and $L_{\text{target}} = \|E[k]\|_2 = \|\tilde{e}_t\|_2$; thus, ignoring the negligible stabilizer ϵ, the calibration factor naturally reduces to 1, leaving the representation unchanged.

4.2 Design Choices

Why use the weighted average norm as target. Jensen's inequality derived in Section 3.2 establishes that $\|\tilde{e}_t\|_2 \leq \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] \cdot \|E[k]\|_2$ for any multi-candidate combination. The weighted average

Table 1: Main results across four models and five benchmarks. Green values in the Average Pass@1 columns denote the Pass@1 gain of each NormCaST variant relative to its non-calibrated counterpart (ST Vanilla or ST Gumbel). Blue and red values in the Average Generation Length columns denote the percentage change relative to CoT. Best results in each block are **bolded**.

	Pass@1 (%) ↑							Generation Length ↓	
	AIME 2024	AIME 2025	GSM8K	GPQA Diamond	MATH 500	Average		Average	
Qwen3-1.7B									
CoT (sampling)	44.44	37.78	90.73	**38.89**	91.80	60.73		10448.8	
ST Vanilla	41.11	33.33	90.09	36.03	90.27	58.17	↑0.88	**9850.9**	↓5.72%
NormCaST	38.89	37.78	90.70	37.37	90.53	59.05		9992.9	↓4.36%
ST Gumbel	43.33	34.44	**91.00**	37.71	**92.20**	59.74	↑1.58	10440.5	↓0.08%
NormCaST Gumbel	**47.78**	**38.89**	**91.00**	37.71	91.20	**61.32**		10161.0	↓2.75%
Qwen3-8B									
CoT (sampling)	70.00	62.22	95.83	58.42	96.47	76.59		9870.4	
ST Vanilla	**77.78**	60.00	95.86	55.39	95.87	76.98	↑0.66	9445.8	↓4.30%
NormCaST	75.56	61.11	95.65	59.26	96.60	77.64		9438.8	↓4.37%
ST Gumbel	72.22	61.11	95.86	59.43	96.80	77.08	↑2.95	9479.6	↓3.96%
NormCaST Gumbel	**77.78**	**68.89**	**95.98**	**60.44**	**97.07**	**80.03**		**9430.2**	↓4.46%
Qwen3-32B									
CoT (sampling)	82.22	**76.67**	**96.29**	64.14	97.73	**83.41**		8008.1	
ST Vanilla	76.67	61.11	96.18	64.81	97.40	79.23	↑2.97	7930.8	↓0.97%
NormCaST	82.22	67.78	95.93	**67.34**	97.73	82.20		**7551.0**	↓5.71%
ST Gumbel	82.22	72.22	96.03	65.49	**97.80**	82.75	↑0.36	8048.0	↑0.50%
NormCaST Gumbel	**84.44**	71.11	96.01	66.33	97.67	83.11		7863.4	↓1.81%
gpt-oss-20b									
CoT (sampling)	**82.22**	71.11	**95.05**	63.64	**92.93**	80.99		5546.9	
ST Vanilla	70.00	66.67	93.93	63.64	89.40	76.73	↑0.52	5211.9	↓6.04%
NormCaST	73.33	66.67	93.78	63.47	89.00	77.25		5370.5	↓3.18%
ST Gumbel	77.78	66.67	94.77	**65.32**	91.80	79.27	↑2.24	**5060.5**	↓8.77%
NormCaST Gumbel	**82.22**	**74.44**	94.77	64.98	91.13	**81.51**		5262.2	↓5.13%

norm L_{target} is therefore the natural choice for a calibration target: it represents the expected norm of a discrete token drawn from the candidate distribution and serves as a theoretically principled upper bound on the soft embedding norm. We compare this against two alternative targets: NormCaST Top-1, which rescales to the top-1 candidate's norm, and NormCaST Random, which rescales to a norm sampled uniformly from the candidate norm range. As reported in Section 5.3, both alternatives underperform the weighted target, which we attribute to their failure to preserve the full distributional norm information. The weighted variant is therefore adopted as our default method.

Integration with existing methods. NormCaST is orthogonal to other Soft Thinking-style latent reasoning methods. While existing approaches modify the candidate sampling strategy (e.g., Soft Thinking Gumbel (Wu et al., 2026)), the soft-discrete switching policy (e.g., SwiReasoning (Shi et al., 2026), SeLaR (Fu & Luo, 2026), ThinkRouter (Xu et al., 2026)), or the termination criterion, NormCaST operates solely on the resulting soft embeddings. Therefore, it can be combined with these methods in a plug-and-play fashion without extra modifications. We empirically validate this complementarity by combining NormCaST with two representative methods (Soft Thinking Gumbel and SwiReasoning), as detailed in Section 5 and Appendix D.2 respectively.

5 Experiments

5.1 Experimental Setup

Benchmarks and Models. We evaluate the proposed method on five benchmarks in different domains, including (i) Mathematics: **AIME 2024** (AMC, 2024), **AIME 2025** (AMC, 2025), **GSM8K** (Cobbe et al., 2021) and **MATH500** (Hendrycks et al., 2021); (ii) Science: **GPQA-Diamond** (Rein et al., 2023). For broader-domain evaluation, we additionally consider three code-generation benchmarks: **HumanEval** (Chen et al., 2021), **MBPP** (Austin et al., 2021), and **LiveCodeBench** (Jain et al., 2024).

Table 2: Ablation on the calibration target norm with Qwen3-8B, comparing our default probability-weighted variant (NormCaST Weighted) against the Top-1 and Random alternatives. Green and blue values denote the Pass@1 gain and the generation length change relative to the CoT (sampling) baseline, respectively. Best results in each block are **bolded**.

	Pass@1 (%) ↑							Generation Length ↓	
	AIME 2024	AIME 2025	GSM8K	GPQA Diamond	MATH 500	Average		Average	
Qwen3-8B									
CoT (sampling)	70.00	62.22	95.83	58.42	96.47	76.59		9870.4	
ST Vanilla	**77.78**	60.00	**95.86**	55.39	95.87	76.98	↑0.39	9445.8	↓4.30%
NormCaST Random	72.22	**64.44**	95.63	58.42	96.00	77.34	↑0.75	9348.4	↓5.29%
NormCaST Top-1	76.67	61.11	95.75	56.23	96.47	77.25	↑0.66	**9285.0**	↓5.93%
NormCaST Weighted	75.56	61.11	95.65	**59.26**	**96.60**	**77.64**	↑1.05	9438.8	↓4.37%

The details are provided in Appendix B.1. To assess generalizability, experiments are conducted on four models from different families, spanning 1.7B to 32B parameters: **Qwen3-1.7B**, **Qwen3-8B**, **Qwen3-32B** (Yang et al., 2025), and **gpt-oss-20b** (Agarwal et al., 2025).

Metrics. We report **Pass@1** accuracy to measure model performance, which is computed as the average correctness over 3 random seeds (0, 7, 42) with one sample per seed. We also report the **Generation Length**, which is the average number of tokens generated per problem including both soft thinking trajectories and discrete answer tokens, to measure token costs and reasoning efficiency.

Baselines and Our Methods. We compare our method with three baselines: (i) **CoT (sampling)**, which is the standard chain-of-thought reasoning with discrete token decoding and sampling strategies; (ii) **Soft Thinking (ST Vanilla)**, the original Soft Thinking method, which performs the reasoning process in continuous embedding space (Zhang et al., 2025); (iii) **Soft Thinking Gumbel (ST Gumbel)**, which proposes adding Gumbel-Softmax noise for enhanced exploration (Wu et al., 2026). All baselines are configured according to their official open-source code and recommended hyperparameters. Our proposed methods, **NormCaST** and **NormCaST Gumbel**, apply norm calibration to ST Vanilla and ST Gumbel, respectively.

Implementation Details. For a fair comparison, all methods use temperature $\tau = 0.6$ and the same sampling filters: top-$k = 20$, top-$p = 0.95$, and min-$p = 0$. For every Soft Thinking variant, we perform top-m candidate selection after filtering with a candidate limit of $m = 10$. The maximum number of generated tokens is set to 32,768. All experiments are implemented with the SGLang inference framework (Zheng et al., 2024) and run on eight NVIDIA A800 80GB GPUs.

5.2 Comparison to Baselines

Table 1 reports Pass@1 accuracy and generation length for all methods across four models and five benchmarks. As calibration operates orthogonally on top of any Soft Thinking variant, our central finding is that NormCaST improves accuracy over the variant it is applied to on most models and benchmarks, while keeping generation length comparable to or shorter than that variant.

Calibration consistently improves every Soft Thinking variant. Applied to Soft Thinking Vanilla, NormCaST raises the average Pass@1 on all four models, with gains of $+0.88$ on Qwen3-1.7B, $+0.66$ on Qwen3-8B, $+2.97$ on Qwen3-32B, and $+0.52$ on gpt-oss-20b. The same holds for the Gumbel variant: NormCaST Gumbel improves over Soft Thinking Gumbel on all four models, by $1.58, 2.95, 0.36$ and 2.24 points on Qwen3-1.7B, 8B, 32B and gpt-oss-20b respectively. Since Gumbel-Softmax perturbs the candidate distribution whereas calibration restores the embedding norm, the two mechanisms act on complementary axes and their gains accumulate rather than interfere.

Best overall accuracy. NormCaST Gumbel attains the highest average Pass@1 on three of the four models and surpasses the CoT baseline on Qwen3-1.7B, Qwen3-8B, and gpt-oss-20b. The improvement is most pronounced on Qwen3-8B, where NormCaST Gumbel reaches 80.03 against 76.59 for CoT and 77.08 for Soft Thinking Gumbel. This confirms that restoring the norm of soft embeddings recovers semantic expressiveness that stochastic exploration alone cannot provide.

Table 3: Results on three code-generation benchmarks with Qwen3-8B. Green values denote the average Pass@1 gain of each NormCaST variant over its non-calibrated counterpart, while blue values denote the average generation-length reduction relative to CoT. Best results are **bolded**.

	Pass@1 (%) ↑					Generation Length ↓	
	Human Eval	LiveCode Bench	MBPP	Average		Average	
Qwen3-8B							
CoT (sampling)	73.98	**56.39**	94.94	75.10		7347.5	
ST Vanilla	74.59	53.76	91.18	73.18	↑0.39	**6345.9**	↓13.63%
NormCaST	76.42	52.33	91.96	73.57		6461.9	↓12.05%
ST Gumbel	76.42	56.03	94.29	75.58	↑0.28	6970.1	↓5.14%
NormCaST Gumbel	**77.64**	54.60	**95.33**	**75.86**		6971.0	↓5.12%

Comparable average generation length. Calibration does not inflate the reasoning budget. Across all models, both NormCaST and NormCaST Gumbel produce shorter outputs than CoT and stay close to their non-calibrated counterparts. On Qwen3-32B, NormCaST shortens generation by 5.71% relative to CoT, far exceeding the 0.97% reduction of Soft Thinking Vanilla, and NormCaST Gumbel shortens it by 1.81% while Soft Thinking Gumbel instead lengthens it by 0.50%. These results show that norm calibration improves reasoning quality without sacrificing efficiency.

5.3 Ablation Study

To assess the effect of target-norm selection, Table 2 compares the uncalibrated ST Vanilla baseline with three rescaling strategies on Qwen3-8B. NormCaST Weighted uses the probability-weighted candidate norm defined in Eq. 9; NormCaST Top-1 uses the norm of the highest-probability candidate $\|E[k_{\text{top-1}}]\|_2$; and NormCaST Random samples a target uniformly from $[\min_{k\in\mathcal{C}_t^+}\|E[k]\|_2, \max_{k\in\mathcal{C}_t^+}\|E[k]\|_2]$. All three calibrated variants improve upon the average Pass@1 of 76.98 achieved by ST Vanilla, indicating that norm rescaling is beneficial across the evaluated target choices. The probability-weighted target achieves the highest average Pass@1 of 77.64, compared with 77.34 for Random and 77.25 for Top-1, and also exceeds 76.59 for CoT. These results support the probability-weighted candidate norm as the default calibration target.

5.4 Results on Broader Domains

To evaluate generalization beyond math and science reasoning, Table 3 reports results on HumanEval, MBPP, and LiveCodeBench with Qwen3-8B. NormCaST increases the average Pass@1 of Soft Thinking Vanilla from 73.18 to 73.57, yielding a gain of 0.39 points. Likewise, NormCaST Gumbel improves upon Soft Thinking Gumbel from 75.58 to 75.86, yielding a gain of 0.28 points and the highest average Pass@1 among the evaluated methods. Relative to CoT, NormCaST and NormCaST Gumbel reduce average generation length by 12.05% and 5.12%, respectively. The gains in average Pass@1 for both paired variants provide evidence that the benefit of norm calibration extends to code-generation tasks beyond the math and science benchmarks used in the main evaluation.

6 Conclusion

In this work, we identify systematic embedding norm shrinkage as a geometric limitation of Soft Thinking at the minority of critical, multi-candidate positions where it differs from discrete CoT. Such norm shrinkage pushes soft embeddings into a low-norm region inconsistent with the input-norm range the model typically encounters. To address this limitation, we introduce NormCaST, a training-free latent reasoning mechanism that preserves each soft embedding's direction while rescaling its norm to the probability-weighted average norm of its candidate tokens. Across four models and five STEM benchmarks, NormCaST improves every paired Soft Thinking variant, with gains of up to $+2.97$ average Pass@1 points. It also improves average accuracy on three code benchmarks and integrates effectively with SwiReasoning, demonstrating its applicability across reasoning domains and Soft Thinking variants.

AI USE STATEMENT

In this work, we used generative AI tools to refine hypotheses, implement methods and interpret results. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, design or provide feedback on research methodology or experiments, or support qualitative and thematic data analysis. Generating synthetic data sets, assisting with translation, and cleaning or reformatting data sets are not applicable to this work. Additionally, we used generative AI tools to edit the research paper to improve readability, suggest a structure for the research paper, and create or edit software code. We have reviewed all AI-assisted work. Specifically, we manually reviewed the paper structures suggested by LLMs for appropriateness and checked AI-edited text for accuracy, clarity, and fluency. LLM-generated code was verified and tested for correctness by two authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on enhancing the reasoning accuracy and efficiency of LLMs, which does not raise safety concerns. This work involves no collection of sensitive data. All evaluations are conducted using publicly available models and benchmarks under their original licenses.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we describe the formulation and calibration procedure of NormCaST in Section 4. Section 5.1 specifies the evaluated models and benchmarks, baselines, metrics, random seeds, inference framework, and hardware. Appendix B.1 provides benchmark sources and evaluation protocols, Appendix B.2 presents detailed pseudocode, and Appendix B.3 lists the decoding and stopping hyperparameters used in our experiments. All reported scores are computed using deterministic programmatic evaluators without an LLM judge.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

AMC. AIME 2024: American Invitational Mathematics Examination. Hugging Face Dataset, 2024. URL https://huggingface.co/datasets/HuggingFaceH4/aime_2024.

AMC. AIME 2025: American Invitational Mathematics Examination. Hugging Face Dataset, 2025. URL https://huggingface.co/datasets/yentinglin/aime_2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. In *AAAI Conference on Artificial Intelligence*, pp. 17682–17690, 2024. doi: 10.1609/AAAI.V38I16.29720. URL https://mlanthology.org/aaai/2024/besta2024aaai-graph/.

Hanzhu Chen, Lin Yang, Jie Wang, Junhao Yan, Zhe Wang, Xize Liang, and Jianye Hao. Latent-guided reasoning: Empowering small LLMs with large-model thinking. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 64406–64434, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6958e9d0f5a76d54ff97da8c45f4d52e-Paper-Conference.pdf.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *CoRR*, abs/2412.13171, 2024. doi: 10.48550/ARXIV.2412.13171. URL https://doi.org/10.48550/arXiv.2412.13171.

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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation, 2023. URL https://arxiv.org/abs/2311.01460.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/dfc310e81992d2e4cedc09ac47eff13e-Abstract-Conference.html.

Renyu Fu and Guibo Luo. SeLaR: Selective latent reasoning in large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 7073–7087. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.320. URL https://doi.org/10.18653/v1/2026.acl-long.320.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 41340–41391. Curran Associates, Inc., 2025. doi: 10.52202/

085713-1380. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3b01972cf31e6fa0fe29e4b8b5c2a0a1-Paper-Conference.pdf.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ph04CRkPdC.

Alperen Gozeten, Muhammed Ildiz, Xuechen Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. Continuous chain of thought enables parallel exploration and reasoning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 82881–82914, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/85ec26ef94c3acb4c195e905df1ff4f7-Paper-Conference.pdf.

Runquan Gui, Zhihai Wang, Jie Wang, Chi Ma, Huiling Zhen, Mingxuan Yuan, Jianye Hao, Defu Lian, Enhong Chen, and Feng Wu. Hypertree planning: Enhancing LLM reasoning via hierarchical thinking. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=45he3Ri6JP.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *CoRR*, abs/2412.06769, 2024. doi: 10.48550/ARXIV.2412.06769. URL https://doi.org/10.48550/arXiv.2412.06769.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213. Curran Associates, Inc., 2022. doi: 10.52202/068431-1613. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: compressing chain-of-thought into continuous space via self-distillation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 677–693. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.36. URL https://doi.org/10.18653/v1/2025.emnlp-main.36.

Dachuan Shi, Abedelkadir Asi, Keying Li, Xiangchi Yuan, Leyan Pan, Wenke Lee, and Wen Xiao. SwiReasoning: Switch-Thinking in Latent and Explicit for Pareto-Superior Reasoning LLMs. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 137060–137093,

2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ddb7bad46132a323aa9d039000941881-Paper-Conference.pdf.

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Jian Luan, and Ruihua Song. Think silently, think fast: Dynamic latent compression of LLM reasoning chains. *CoRR*, abs/2505.16552, 2025. doi: 10.48550/ARXIV.2505.16552. URL https://doi.org/10.48550/arXiv.2505.16552.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Zehao Wang, Lin Yang, Jie Wang, Kehan Wang, Hanzhu Chen, Bin Wang, Jianye Hao, Defu Lian, Bin Li, and Enhong Chen. Logictree: Improving complex reasoning of LLMs via instantiated multi-step synthetic logical data. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113992–114027. Curran Associates, Inc., 2025. doi: 10.52202/085713-3802. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a5b5342b9abcf83fe882169cfd48d5e7-Paper-Conference.pdf.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. SIM-CoT: Supervised implicit chain-of-thought. *CoRR*, abs/2509.20317, 2025. doi: 10.48550/ARXIV.2509.20317. URL https://doi.org/10.48550/arXiv.2509.20317.

Junhong Wu, Jinliang Lu, Zixuan Ren, Gangqiang Hu, Zhi Wu, Dai Dai, and Hua Wu. LLMs are single-threaded reasoners: Demystifying the working mechanism of soft thinking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ASLuOoP78o.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10524–10533. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/xiong20b.html.

Xin Xu, Tong Yu, Xiang Chen, Haoliang Wang, Julian McAuley, and Saayan Mitra. ThinkRouter: Efficient reasoning via routing thinking between latent and discrete spaces, 2026. URL https://arxiv.org/abs/2602.11683.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23336–23351. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.1137. URL https://aclanthology.org/2025.acl-long.1137/.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot++: Test-time scaling with soft chain-of-thought reasoning, 2025b. URL https://arxiv.org/abs/2505.11484.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang

Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, and Xin Wang. Soft thinking: Unlocking the reasoning potential of LLMs in continuous concept space. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/f7396d1c54d51416958d63e285377103-Abstract-Conference.html.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: efficient execution of structured language model programs. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

A Related Work

Explicit Reasoning. Explicit reasoning represents intermediate computation as discrete text. Chain-of-Thought (CoT) prompting elicits step-by-step solutions through demonstrations containing natural-language rationales (Wei et al., 2022). Subsequent work improves the reliability and generalization of this paradigm while retaining textual reasoning chains (Kojima et al., 2022; Wang et al., 2023; Zhou et al., 2023; Shao et al., 2024; Wang et al., 2025). Tree of Thoughts organizes intermediate thoughts into a search tree, enabling evaluation and backtracking across solutions (Yao et al., 2023). Graph of Thoughts and HyperTree Planning further extend the organization of explicit reasoning for complex tasks (Besta et al., 2024; Gui et al., 2025). However, even when exploring multiple paths, these approaches construct each path through discrete token choices: every decoding step commits to one token rather than passing the candidate distribution forward as a continuous input.

Training-based latent reasoning. Training-based methods use pretraining or post-training to learn continuous intermediate representations that support reasoning without verbalizing every step (Deng et al., 2023; Cheng & Van Durme, 2024; Geiping et al., 2025; Xu et al., 2025a;b; Wei et al., 2025; Tan et al., 2025; Chen et al., 2026; Gozeten et al., 2026). Coconut exemplifies this paradigm by feeding final hidden state back as the next input embedding and using curriculum training to replace explicit reasoning with continuous thoughts (Hao et al., 2024). CODI uses self-distillation from an explicit CoT branch to train a continuous reasoning branch within a shared model, compressing textual reasoning into recurrent hidden states (Shen et al., 2025). Across this literature, the shared objective is to preserve reasoning capability while reducing the dependence on lengthy textual chains. Although these methods can improve token efficiency, they require additional model or interface training, motivating training-free alternatives that apply directly to pretrained models.

Training-free latent reasoning. Training-free methods construct continuous inputs from token distributions without updating model parameters. Soft Thinking forms each intermediate input as the probability-weighted average of candidate token embeddings and feeds it back to the model, preserving information in a single continuous representation (Zhang et al., 2025). Stochastic Soft Thinking identifies that the soft embedding is dominated by the highest-probability token and introduces noise into the soft-token distribution to encourage exploration (Wu et al., 2026). Other work adaptively combines latent and discrete reasoning to improve reasoning quality and efficiency (Shi et al., 2026; Fu & Luo, 2026; Xu et al., 2026). These refinements primarily change sampling noise or the policy for switching between latent and discrete computation. By addressing norm shrinkage at the representation level, NormCaST is orthogonal to these inference strategies and can be incorporated into their continuous reasoning steps.

B Supplementary Details

B.1 Benchmark and Evaluation Details

We conduct evaluations on 8 reasoning benchmarks, including GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), AIME 2024 (AMC, 2024), AIME 2025 (AMC, 2025) for mathematical reasoning; GPQA-Diamond (Rein et al., 2023) for STEM reasoning; HumanEval (Chen et al., 2021), LiveCodeBench (Jain et al., 2024) and MBPP (Austin et al., 2021) for code generation.

- **GSM8K:** The official test split contains 1,319 grade-school mathematics word problems that evaluate multi-step arithmetic reasoning in natural language. https://huggingface.co/datasets/openai/gsm8k.
- **MATH500:** This curated evaluation set contains 500 high-school competition problems spanning algebra, geometry, number theory, and related mathematical topics. https://huggingface.co/datasets/HuggingFaceH4/MATH-500.
- **AIME 2024:** The benchmark contains 30 competition-level problems from the 2024 American Invitational Mathematics Examination, each requiring a concise integer answer. https://huggingface.co/datasets/HuggingFaceH4/aime_2024.
- **AIME 2025:** The evaluation set contains 30 AIME problems in the same competition-style numerical-answer format. https://huggingface.co/datasets/yentinglin/aime_2025.

- **GPQA-Diamond:** This expert-validated subset contains 198 multiple-choice questions across biology, physics, and chemistry. https://huggingface.co/datasets/hendrydong/gpqa_diamond_mc.
- **HumanEval:** The benchmark comprises 164 handwritten Python function-completion tasks evaluated by executing generated code against unit tests. https://huggingface.co/datasets/openai/openai_humaneval.
- **LiveCodeBench:** The evaluation version used here contains 279 competitive-programming problems, with generated programs checked against executable tests. https://huggingface.co/datasets/PrimeIntellect/LiveCodeBench-v5.
- **MBPP:** We use the 257-problem test split of the sanitized MBPP configuration, where each Python programming task includes a natural-language specification and unit tests. https://huggingface.co/datasets/google-research-datasets/mbpp.

The evaluation implementation follows the official Soft Thinking repository. We use Math-Verify for answer parsing and matching on the five mathematics and science benchmarks, with exact option matching for GPQA-Diamond. For the three coding benchmarks, we use their official executable-code evaluators. All reported scores are computed by deterministic programmatic evaluators and no LLM judge is used.

B.2 NormCaST Implementation

Algorithm 1 provides the detailed implementation of NormCaST. The standard Soft Thinking inference procedure is shown in black, while the norm-calibration operations are highlighted in blue: they compute the probability-weighted target norm and rescale the soft embedding before it is appended to the reasoning sequence.

Algorithm 1 Norm-Calibrated Soft Thinking (NormCaST)

Require: Query tokens $x_{1:Q}$, language model LLM, input embedding matrix E, maximum generation length $T_{\max}$, sampling parameters (τ, k, p, ρ), candidate limit m, Cold Stop parameters (δ, l), and ϵ
Ensure: Final answer y

Initialize the soft-embedding sequence $\tilde{e}_{1:0} \leftarrow [\,]$
for $t = 1, \ldots, T_{\max}$ **do**
  $\ell_t \leftarrow \mathrm{LLM}\big(E[x_{1:Q}], \tilde{e}_{1:t-1}\big)$ ▷ Forward pass
  $p_t^{\mathrm{raw}} \leftarrow \mathrm{softmax}(\ell_t)$ ▷ Raw next-token distribution
  $p_t^{\tau} \leftarrow \mathrm{softmax}(\ell_t/\tau)$ ▷ Temperature scaling
  $p_t \leftarrow \mathrm{Sample}(p_t^{\tau}; k, p, \rho)$ ▷ Filtering and renormalization
  if $\mathrm{ColdStop}(p_t; \delta, l)$ **or** $\arg\max_{v \in V} p_t[v] = r_{\mathrm{EOT}}$ **then**
    $\tilde{e}_{1:t} \leftarrow \tilde{e}_{1:t-1} \,\|\, E[r_{\mathrm{EOT}}]$
    break ▷ End soft thinking and switch modes
  end if
  $\mathcal{C}_t \leftarrow \text{Top-M}(p_t, m)$ ▷ Retain candidate tokens
  $\mathcal{C}_t^{+} \leftarrow \{v \in \mathcal{C}_t \mid p_t[v] > 0\}$; $n \leftarrow |\mathcal{C}_t^{+}|$ ▷ Effective candidates
  $\tilde{p}_t[k] \leftarrow p_t[k] / \sum_{v \in \mathcal{C}_t} p_t[v]$ for $k \in \mathcal{C}_t$
  $\tilde{e}_t \leftarrow \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] \cdot E[k]$ ▷ Construct the vanilla soft embedding
  $L_{\mathrm{target}} \leftarrow \sum_{k \in \mathcal{C}_t} \tilde{p}_t[k] \cdot \|E[k]\|_2$ ▷ Compute target norm
  $\tilde{e}_{t,\mathrm{cal}} \leftarrow \tilde{e}_t \cdot L_{\mathrm{target}} / (\|\tilde{e}_t\|_2 + \epsilon)$ ▷ Norm calibration; identity when $n = 1$
  $\tilde{e}_{1:t} \leftarrow \tilde{e}_{1:t-1} \,\|\, \tilde{e}_{t,\mathrm{cal}}$ ▷ Append the calibrated embedding
end for
$y \leftarrow \mathrm{DiscreteDecode}\big(\mathrm{LLM}; E[x_{1:Q}], \tilde{e}_{1:t}\big)$ ▷ Standard discrete answer decoding
return y

Here, Sample applies top-k, top-p, and min-p filtering and renormalizes the probabilities; k, p, and ρ are respectively the top-k cutoff, top-p cumulative-mass threshold, and min-p relative-probability threshold. We use ρ for min-p to distinguish it from the top-p threshold p. $\mathrm{ColdStop}(p_t; \delta, l)$ becomes true after l consecutive steps whose entropy is below δ; a naturally predicted r_{EOT} likewise ends the

soft phase, and in either case its discrete embedding is appended once to mark the mode transition. $\mathrm{DiscreteDecode}$ denotes ordinary autoregressive token sampling from the resulting context until EOS or the remaining length budget is exhausted.

Computational cost analysis. NormCaST reuses the candidate embeddings and probabilities already computed by Soft Thinking and requires no additional model forward pass. Beyond the original weighted sum, each step adds only the norms of at most m candidate embeddings, one weighted scalar reduction, the norm of $\tilde{e}_t$, and one scalar–vector rescaling. These operations require $O(m \cdot d)$ computation, matching the asymptotic cost of constructing the soft embedding itself. With $m = 10$ fixed in our experiments, this overhead is negligible relative to the Transformer forward pass. The method also introduces no trainable parameters, additional KV-cache entries, or extra decoding steps.

B.3 Hyperparameter Details

All datasets use temperature $\tau = 0.6$, top-$k = 20$, top-$p = 0.95$, min-$p = 0$, and a maximum generation length of 32,768. Every Soft Thinking variant sets the candidate limit to $m = 10$ after filtering. The resulting effective candidate count is $n = |\mathcal{C}_t^+|$ and can be smaller than m when filtering assigns zero probability to retained entries. The Gumbel-Softmax temperature is set to 0.5. Following the official Soft Thinking implementation, Cold Stop uses entropy threshold $\delta = 0.01$ and requires $l = 256$ consecutive confident steps. We apply the same Cold Stop settings to all four Soft Thinking variants in the main comparison, and no dataset-specific decoding parameter is tuned.

C Supplementary Motivation Analysis

C.1 Detailed Analysis of Probability Concentration

We compare three distributions at each reasoning step: the raw softmax $p_t^{\mathrm{raw}} = \mathrm{softmax}(\ell_t)$, the temperature-scaled softmax $p_t^{\tau} = \mathrm{softmax}(\ell_t/\tau)$ with $\tau = 0.6$, and the distribution $\tilde{p}_t$ obtained after applying the sampling filters, top-m candidate selection, and candidate-level renormalization. The last is obtained by applying the sampling filters to p_t^{τ}, retaining the $m = 10$ highest-probability entries, and renormalizing their probabilities.

Table 4: Mean top-1 probability at the three stages of the Soft Thinking sampling pipeline.

Dataset	Model	Raw softmax	Temperature-scaled	Filtered, top-m selected, and renormalized
GSM8K	Qwen3-1.7B	0.912	0.947	0.955
	Qwen3-8B	0.917	0.950	0.958
GPQA-Diamond	Qwen3-1.7B	0.918	0.952	0.957
	Qwen3-8B	0.875	0.925	0.940

Table 4 shows a monotonic increase for every model–dataset pair. Importantly, the raw distributions are already highly concentrated, with mean top-1 mass between 0.875 and 0.918. Temperature scaling increases this range to 0.925–0.952, and sampling filters followed by candidate truncation and renormalization further raise it to 0.940–0.958. Thus, the sampling pipeline amplifies rather than creates the model's intrinsic probability concentration.

Figures 4–7 show the three stages for both models on GPQA-Diamond and GSM8K. Each figure pairs the complete probability range with a magnified view of the high-concentration region.

C.2 More Detailed Analysis of the Minority of Critical Positions

Table 5 summarizes the mean effective candidate count and the proportion of single-candidate steps under thresholds $\gamma \in \{0.1, 0.01, 0\}$. The canonical definition uses $\gamma = 0$, and a position is multi-candidate when $n \geq 2$ under this threshold.

Under the canonical threshold $\gamma = 0$, the mean effective count ranges from 1.25 to 1.42, while 74.40–82.06% of reasoning steps contain exactly one effective candidate. Thus, only a minority of positions combine multiple candidate embeddings.

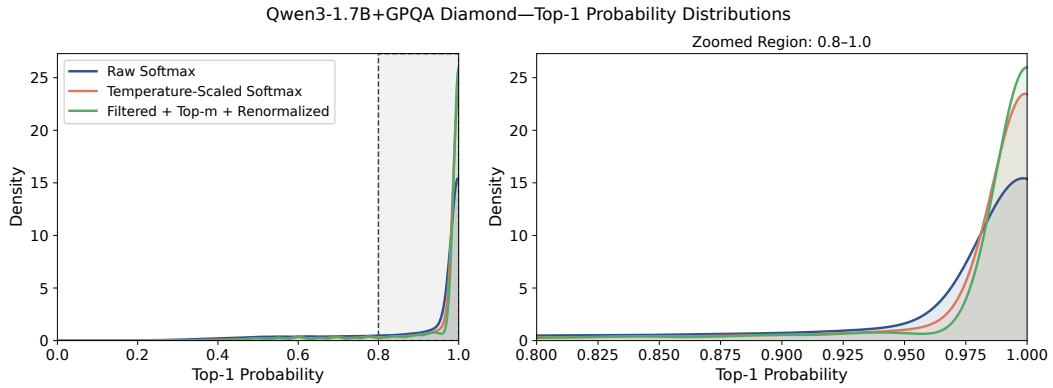

Figure 4: Top-1 probability distributions at the three Soft Thinking sampling stages for Qwen3-1.7B on GPQA-Diamond. The left panel shows the complete $[0, 1]$ range, with the dashed line marking the boundary of the interval magnified in the right panel ($[0.8, 1.0]$). From raw softmax to temperature scaling and then sampling filters plus truncation to at most m candidates with renormalization, probability mass moves progressively toward one, showing that post-processing amplifies concentration already present in the raw distribution.

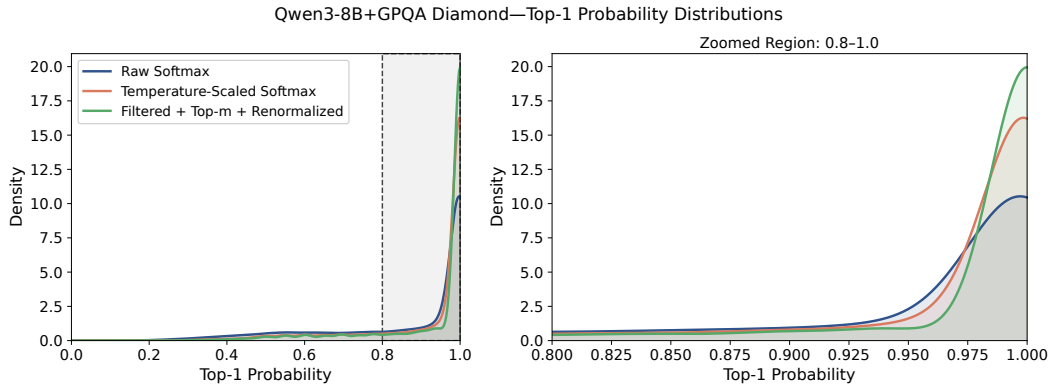

Figure 5: Three-stage top-1 probability distributions for Qwen3-8B on GPQA-Diamond, showing the same progressive concentration after temperature scaling, sampling filters, and candidate renormalization.

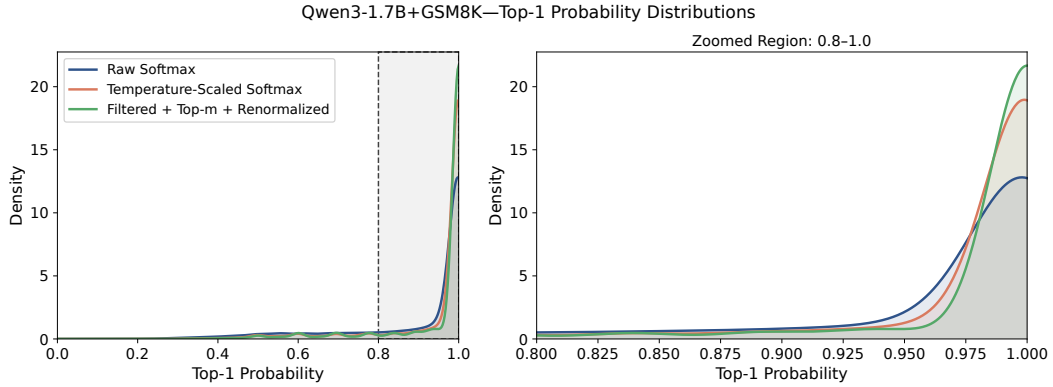

Figure 6: Three-stage top-1 probability distributions for Qwen3-1.7B on GSM8K.

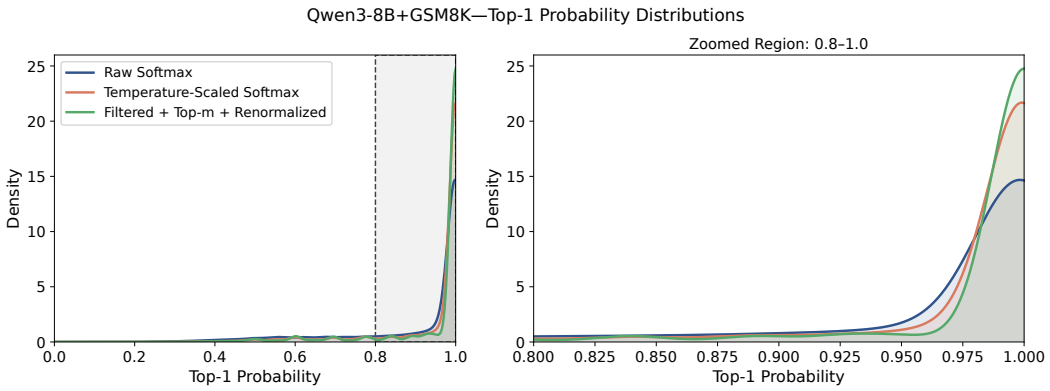

Figure 7: Three-stage top-1 probability distributions for Qwen3-8B on GSM8K.

Table 5: Average effective candidate count and percentage of single-candidate steps. Counts are computed from the filtered and renormalized distribution.

Dataset	Model	Threshold γ	Mean effective count	Single-candidate steps
GSM8K	Qwen3-1.7B	0.1	1.17	85.22%
		0.01	1.28	79.94%
		0	1.28	79.94%
	Qwen3-8B	0.1	1.16	86.60%
		0.01	1.25	82.06%
		0	1.25	82.06%
GPQA-Diamond	Qwen3-1.7B	0.1	1.20	83.65%
		0.01	1.35	77.21%
		0	1.35	77.21%
	Qwen3-8B	0.1	1.24	81.28%
		0.01	1.42	74.40%
		0	1.42	74.40%

Figure 8 presents the GSM8K counterpart to the top row of Figure 1 and shows the same single-candidate dominance.

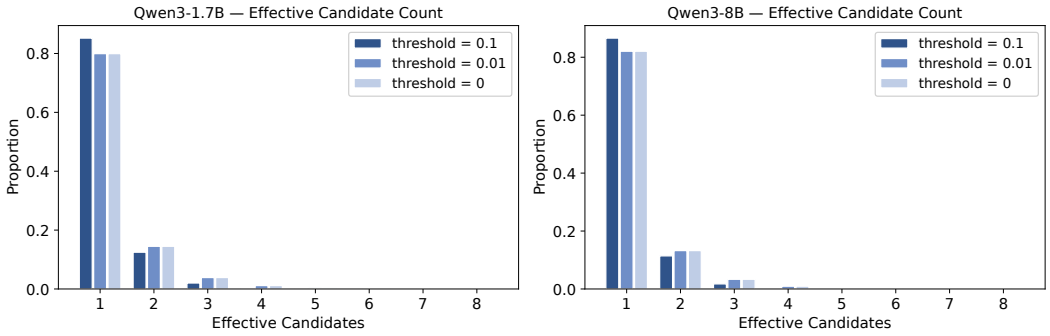

Figure 8: Effective candidate count per reasoning step on GSM8K, showing over 75% single-candidate dominance at all thresholds.

C.3 More Detailed Analysis of Norm Shrinkage

Multi-candidate steps. Figure 9 provides the GSM8K counterpart to the bottom row of Figure 1 and shows the same leftward shift of soft embedding norms relative to probability-weighted candidate norms on the multi-candidate subset.

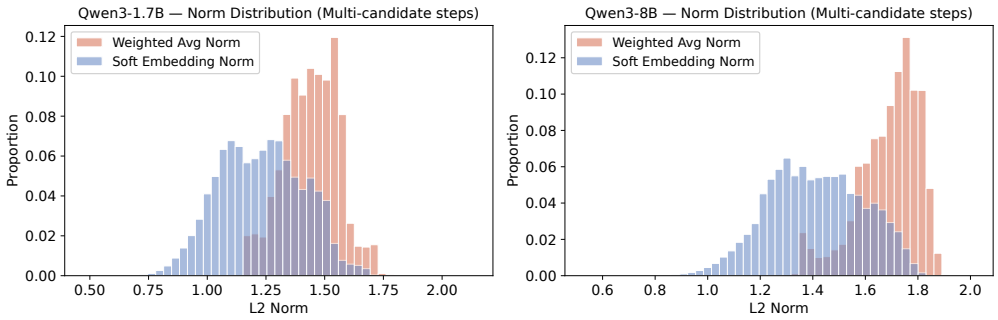

Figure 9: Soft embedding norms and probability-weighted candidate norms on the GSM8K multi-candidate subset ($\gamma = 0$).

All reasoning steps. Figure 10 provides the GSM8K counterpart to Figure 2; both scatter plots aggregate all reasoning steps and show the same per-step norm deficit against the weighted target and the top-1 candidate embedding. Figures 11 and 12 present the corresponding all-step norm distributions on GPQA-Diamond and GSM8K. Compared with the multi-candidate subsets, norm shrinkage is less evident after aggregating all reasoning steps because the minority of positions that shrink is diluted by the large number of single-candidate positions whose norms remain unchanged.

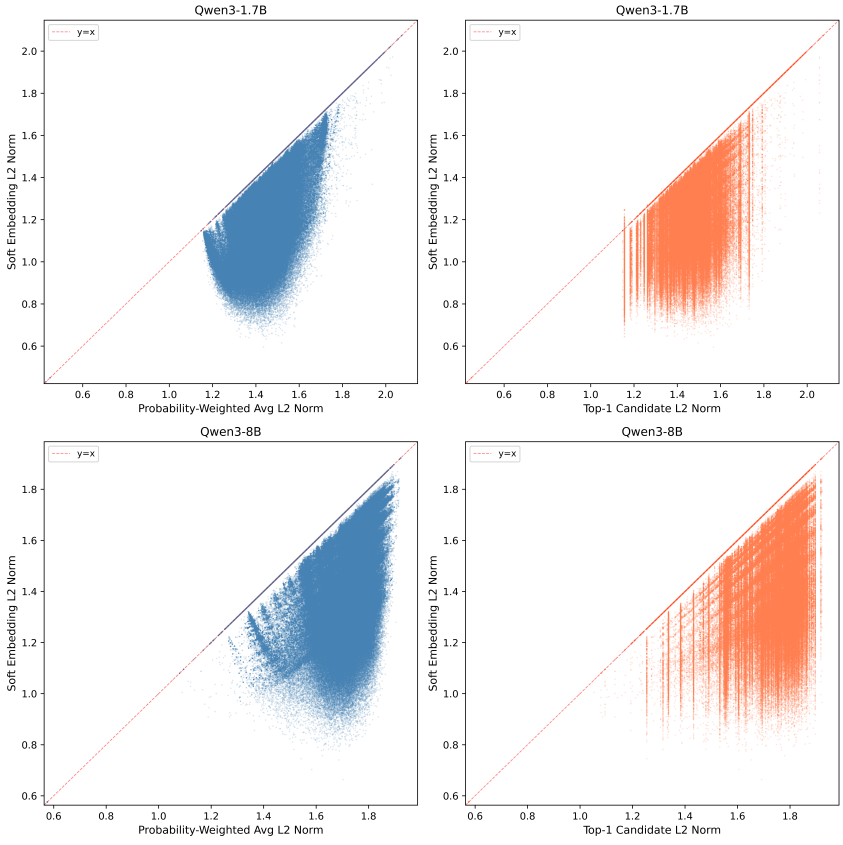

Figure 10: Per-step GSM8K comparison over all reasoning steps. The weighted-average comparisons obey the theoretical diagonal bound; the top-1 comparisons show the same systematic deficit relative to a representative discrete candidate.

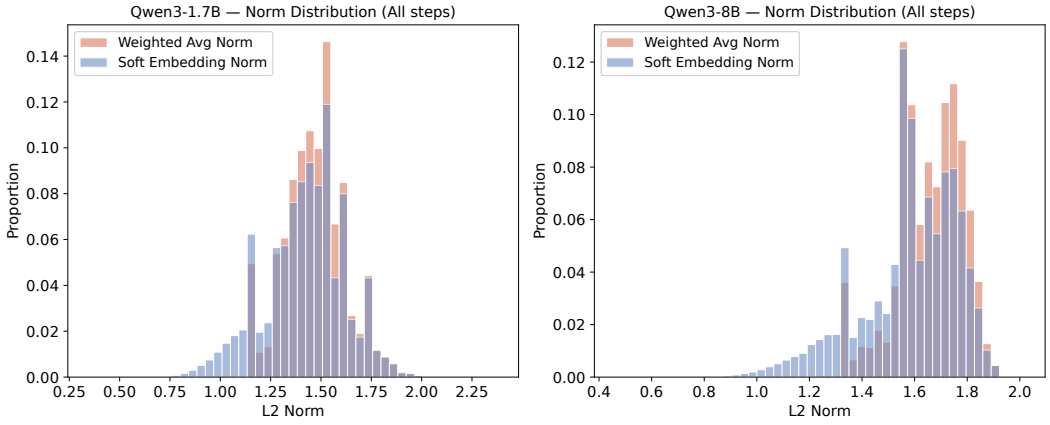

Figure 11: All-step soft embedding norms and probability-weighted candidate norms on GPQA-Diamond.

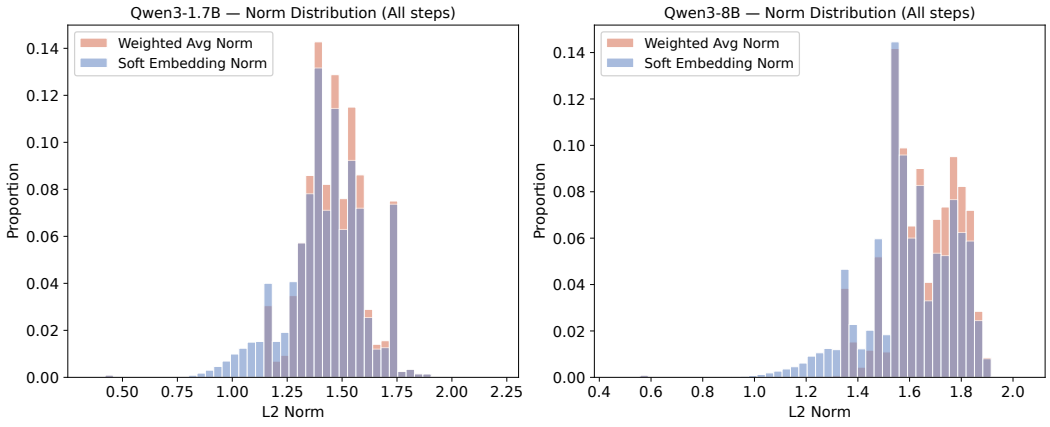

Figure 12: All-step soft embedding norms and probability-weighted candidate norms on GSM8K.

C.4 Detailed Analysis of Low-Norm Vocabulary Entries

We analyze all tokenizer-backed rows in the input embedding matrices of Qwen3-1.7B and Qwen3-8B. Each matrix contains 151,936 rows, of which 151,669 correspond to tokenizer entries; the remaining 267 padding rows are excluded.

The per-step scatter plots in Figure 2 show that the lowest 0.1% tail of all-step top-1 candidate norms forms a distinct cluster on the $y = x$ diagonal in the lower-left corner of each panel, clearly separated from the main candidate cluster. Inspection shows that steps in this tail correspond to three special or control tokens: `<think>` and `</think>`, which delimit the thinking phase, and `<|im_end|>`, which terminates a message or generated sequence. Because this extreme tail is dominated by highly structured, near-deterministic boundary events rather than ordinary reasoning continuations, we remove the lowest 0.1% of norms before calculating the threshold. For each model, we then take the minimum norm in the remaining set—equivalently, the 0.1th percentile of the pooled all-step top-1 candidate norms from GPQA-Diamond and GSM8K. This yields low-norm thresholds of 1.156 for Qwen3-1.7B and 1.336 for Qwen3-8B.

Each tokenizer entry is decoded independently and assigned to one of seven categories using the following rules:

- **Latin**: Latin letters or combining marks form the dominant script in the decoded token.
- **CJK**: Han characters, Japanese kana, Hangul, or Bopomofo form the dominant script.
- **Cyrillic**: Cyrillic letters or combining marks form the dominant script.
- **Arabic**: Arabic letters or combining marks form the dominant script.
- **Other or mixed scripts**: another script is dominant, or multiple scripts tie for the largest character count.
- **Digits, punctuation, or symbols**: the decoded token contains no Unicode letter or combining mark and is not assigned to the special category.
- **Special, byte, or control**: the entry is a special ID, byte-form token, control token, empty decode, or replacement-character decode; this rule takes precedence over the script rules.

Table 6: Composition of the full tokenizer-backed vocabulary and the model-specific low-norm subsets. Category columns report the percentage within each subset; non-Latin combines CJK, Cyrillic, Arabic, and other or mixed scripts. Enrichment is the non-Latin share relative to that of the full vocabulary.

(a) Subset statistics and enrichment

Subset	d	Threshold	Entries	Vocab. share	Enrichment
Full tokenizer vocabulary	–	–	151,669	100.00%	1.00×
Qwen3-1.7B low-norm subset	2,048	1.156	9,409	6.20%	2.51×
Qwen3-8B low-norm subset	4,096	1.336	46,758	30.83%	1.45×

(b) Category composition

Subset	Latin	Non-Latin	Digits / symbols	Special / byte
Full tokenizer vocabulary	61.63%	30.90%	4.18%	3.30%
Qwen3-1.7B low-norm subset	5.27%	77.65%	12.29%	4.79%
Qwen3-8B low-norm subset	41.28%	44.94%	7.05%	6.73%

Table 6 shows that the low-norm subsets have a markedly different composition from the complete tokenizer vocabulary, with non-Latin scripts enriched under both model-specific thresholds. Figures 13 and 14 provide the corresponding two-panel views across the full embedding-norm range: the upper stacked bars report the tokenizer-category composition within each norm bin, while the lower bars report the number of entries on a logarithmic scale. The dashed line marks the characteristic threshold for each model. Both figures show that the bins below this threshold contain substantially fewer Latin entries and larger shares of non-Latin scripts, digits or symbols, and special or byte tokens than the higher-norm region associated with common English reasoning candidates.

Together with the trajectory evidence that ordinary reasoning candidates rarely occupy this range, this compositional shift shows that the low-norm region is not representative of the model's usual input-embedding regime during English STEM reasoning. Soft embeddings pushed into this region therefore enter the residual stream at an out-of-distribution magnitude misaligned with the model's expected input distribution.

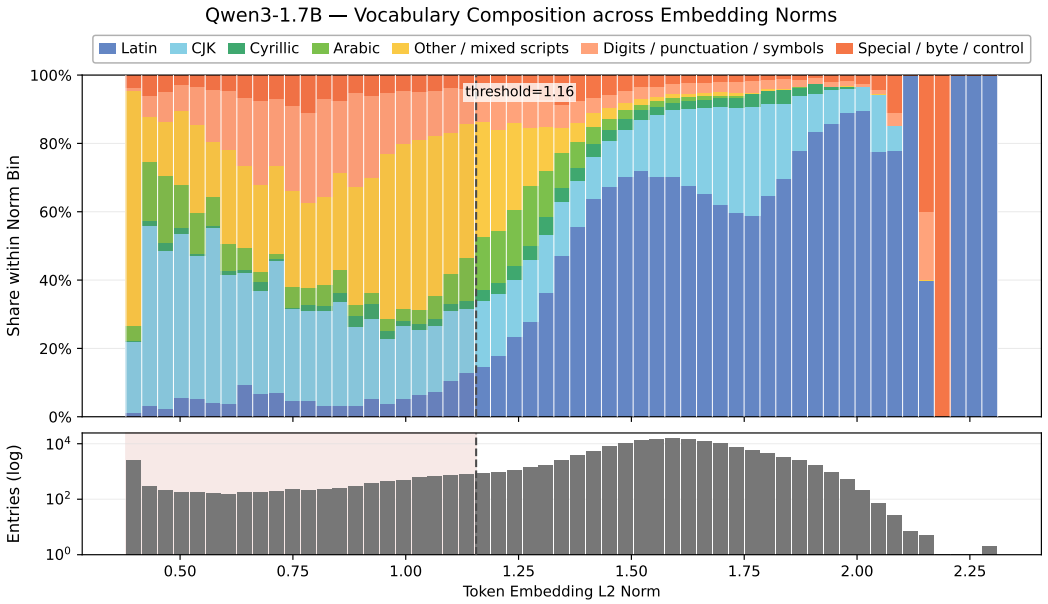

Figure 13: Vocabulary composition across input-embedding norms for Qwen3-1.7B. The upper panel reports category shares within each norm bin, the dashed line marks the trajectory-derived threshold, and the lower panel reports the number of entries on a logarithmic scale.

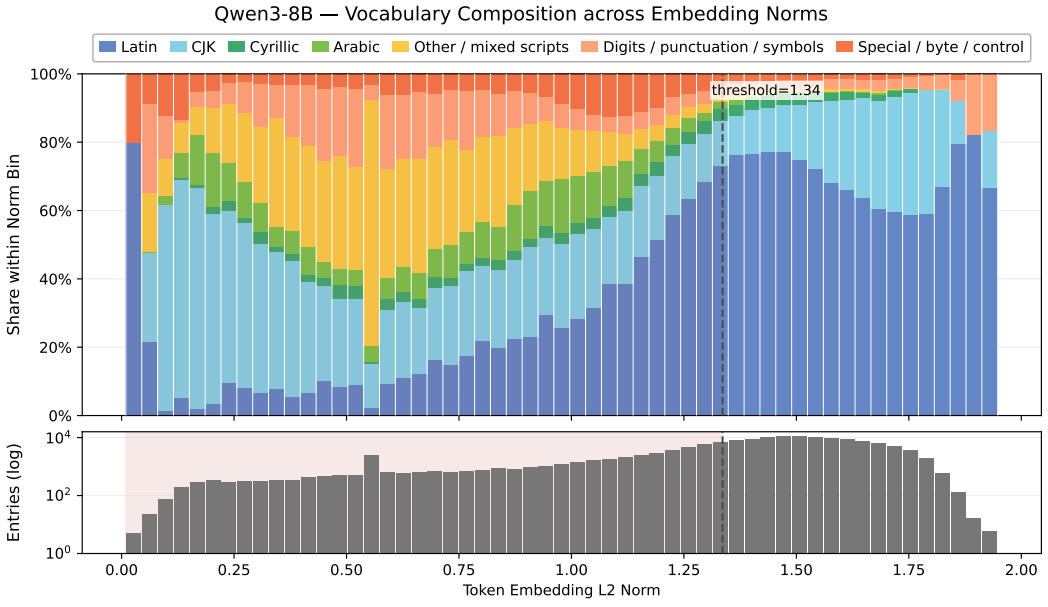

Figure 14: Vocabulary composition across input-embedding norms for Qwen3-8B, using the same two-panel layout, threshold definition, and classification protocol as Figure 13.

D SUPPLEMENTARY RESULTS

D.1 DETAILED GENERATION-LENGTH RESULTS

Table 7 expands the average generation-length column of Table 1 into per-benchmark values. Overall, calibration does not substantially increase reasoning length relative to the corresponding uncalibrated variants, and several calibrated configurations instead produce shorter trajectories.

Table 7: Detailed generation-length results across four models and five benchmarks. Length includes both Soft Thinking steps and discrete answer tokens; lower is better. Blue and red values in the Average columns denote the percentage change relative to CoT. Best results in each model block are **bolded**.

Generation Length ↓							
	AIME 2024	**AIME 2025**	**GSM8K**	**GPQA Diamond**	**MATH 500**	**Average**	
Qwen3-1.7B							
CoT (sampling)	17416.4	18676.6	2187.7	**8489.2**	5474.2	10448.8	
ST Vanilla	**16751.2**	15778.6	2188.8	9017.3	5518.7	**9850.9**	↓5.72%
NormCaST	17133.9	**15683.3**	2231.8	9216.5	5699.0	9992.9	↓4.36%
ST Gumbel	17846.5	17945.6	**2137.3**	8771.7	5501.5	10440.5	↓0.08%
NormCaST Gumbel	17482.1	16864.6	2144.5	8890.0	**5423.9**	10161.0	↓2.75%
Qwen3-8B							
CoT (sampling)	15664.3	17679.1	2261.2	8372.2	5375.1	9870.4	
ST Vanilla	14496.4	**16621.1**	**2223.4**	8484.4	5403.6	9445.8	↓4.30%
NormCaST	**14133.1**	17097.5	2223.9	8336.6	5403.0	9438.8	↓4.37%
ST Gumbel	14647.5	16886.0	2233.4	8277.5	5353.8	9479.6	↓3.96%
NormCaST Gumbel	14730.4	16658.8	2229.5	**8211.0**	**5321.1**	**9430.2**	↓4.46%
Qwen3-32B							
CoT (sampling)	11718.5	16140.9	1730.1	5808.6	4642.5	8008.1	
ST Vanilla	12442.4	15525.7	1643.1	5527.2	4515.6	7930.8	↓0.97%
NormCaST	**10763.9**	**15389.1**	**1638.9**	**5408.5**	4554.6	**7551.0**	↓5.71%
ST Gumbel	12676.9	15564.9	1677.9	5816.1	4504.1	8048.0	↑0.50%
NormCaST Gumbel	12084.8	15409.2	1676.6	5643.0	**4503.2**	7863.4	↓1.81%
gpt-oss-20b							
CoT (sampling)	8434.0	11886.7	475.3	4988.1	1950.4	5546.9	
ST Vanilla	8676.1	10776.8	503.3	**4498.5**	**1605.0**	5211.9	↓6.04%
NormCaST	8381.0	**10628.1**	523.3	5239.1	2080.9	5370.5	↓3.18%
ST Gumbel	7882.7	10676.5	460.0	4523.1	1760.2	**5060.5**	↓8.77%
NormCaST Gumbel	**7818.9**	11615.4	**456.7**	4651.0	1768.8	5262.2	↓5.13%

D.2 INTEGRATION WITH SWIREASONING

To demonstrate the plug-and-play nature of NormCaST, we combine it with the switching policy of SwiReasoning (Shi et al., 2026). All SwiReasoning-specific hyperparameters follow the settings reported in the original paper. Specifically, we use the asymmetric dwell windows $W_{E\to L} = 512$ and $W_{L\to E} = 0$ and fix the exit-signal coefficient to $\beta_0 = 0.7$. The entrance-signal coefficient α_0 follows the original model–dataset-specific configuration reported in Table 8. Following the original unlimited-budget setting, switch-count control is disabled ($C_{\max} = \infty$, corresponding to `max_switch_count=None`). The shared global generation cap remains 32,768.

Norm calibration is applied at every SwiReasoning latent step immediately after the probability-weighted candidate embedding is formed and before transition-signal mixing. The explicit-decoding procedure, control-token definitions, entropy-based routing rule, routing hyperparameters, and random seeds are held fixed between SwiReasoning Vanilla (SwiR Vanilla) and NormCaST SwiReasoning (NormCaST SwiR). Table 9 shows that this isolated change raises average Pass@1 by 1.75 points on Qwen3-1.7B and 2.27 points on Qwen3-8B. It also shortens the corresponding SwiReasoning trajectories, although both unlimited-switch variants remain longer than CoT on average. These

Table 8: SwiReasoning entrance coefficient α_0 used in the integration experiments. Following the original SwiReasoning configuration, α_0 varies by model and dataset, whereas the remaining SwiReasoning-specific parameters are shared across all entries.

Model	GSM8K	MATH500	AIME 2024	AIME 2025	GPQA-Diamond
Qwen3-1.7B	0.6	0.5	0.5	0.3	1.0
Qwen3-8B	0.5	1.0	0.9	0.9	1.0

Table 9: Supplementary results comparing a representative switch policy method, SwiReasoning, with its NormCaST-enhanced variant, NormCaST SwiR. Green values in the Average Pass@1 column denote the absolute Pass@1 gain of each NormCaST variant relative to its non-calibrated counterpart (e.g., NormCaST SwiR vs. SwiR Vanilla). In the Average Generation Length column, blue and red values indicate the percentage decrease and increase, respectively, compared to the standard CoT (sampling) baseline.

	Pass@1 (%) ↑							**Generation Length ↓**	
	AIME 2024	**AIME 2025**	**GSM8K**	**GPQA Diamond**	**MATH 500**	**Average**		**Average**	
Qwen3-1.7B									
CoT (sampling)	**44.44**	37.78	**90.73**	**38.89**	**91.80**	**60.73**		10448.8	
ST Vanilla	41.11	33.33	90.09	36.03	90.27	58.17	↑0.88	**9850.9**	↓5.72%
NormCaST	38.89	37.78	90.70	37.37	90.53	59.05		9992.9	↓4.36%
SwiR Vanilla	**44.44**	31.11	90.19	33.16	89.67	57.71	↑1.75	11930.8	↑14.18%
NormCaST SwiR	43.33	**41.11**	90.25	32.15	90.47	59.46		11403.2	↑9.13%
Qwen3-8B									
CoT (sampling)	70.00	62.22	95.83	58.42	96.47	76.59		9870.4	
ST Vanilla	**77.78**	60.00	**95.86**	55.39	95.87	76.98	↑0.66	9445.8	↓4.3%
NormCaST	75.56	61.11	95.65	**59.26**	**96.60**	**77.64**		**9438.8**	↓4.37%
SwiR Vanilla	64.44	64.44	95.30	54.71	95.60	74.90	↑2.27	11086.8	↑12.32%
NormCaST SwiR	68.89	**67.78**	95.30	58.75	95.13	77.17		10682.0	↑8.22%

paired results support the claim that norm calibration is complementary to latent–explicit routing rather than a replacement for it.