From Noise to Diversity: Random Embedding Injection in LLM Reasoning

Heejun Kim [*1] **Seungpil Lee** [*1] **Jewon Yeom** [2] **Jaewon Sok** [2] **Seonghyeon Park** [2] **Jeongjae Park** [2] **Taesup Kim** [2] **Sundong Kim** [1]

Abstract

Recent *soft prompt* research has tried to improve reasoning by inserting trained vectors into LLM inputs, yet whether the gain comes from the *learned content* or from the *act of injection itself* has not been carefully separated. We study Random Soft Prompts (RSPs), which drop the training step entirely and append a freshly drawn sequence of random embedding vectors to the input. Each RSP vector is sampled from an isotropic Gaussian matching the pretrained embedding-table statistics; the sequence carries no learned content, and yet reaches accuracy comparable to optimized *soft prompts* on math reasoning benchmarks in several settings. The mechanism unfolds in two stages: because attention has to absorb a never-seen-before random position, the distribution over the first few generated tokens flattens and reasoning trajectories *branch*, and as generation continues this influence dilutes naturally so the response commits to a single completion. We show that during inference RSPs lift early-stage token diversity and, combined with temperature sampling, widen Pass@N. Our contributions are: (i) RSP isolates the simplest form of *soft prompt* — training-free, freshly resampled — providing a unified lens for the structural effect of injection that variants otherwise differing in training and form all share; (ii) a theoretical and empirical validation of the underlying mechanism: attention-mediated branch opening, isotropic coverage across all directions, and the automatic attenuation of RSP attention as the KV cache grows; and (iii) an extension from inference to DAPO training, demonstrating practical gains.

[*]Equal contribution [1]Gwangju Institute of Science and Technology [2]Seoul National University. Correspondence to: Taesup Kim <taesup.kim@snu.ac.kr>, Sundong Kim <sundong@gist.ac.kr>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

As LLMs scale to billions of parameters, full fine-tuning is prohibitive, motivating parameter-efficient methods (Hu et al., 2022; Houlsby et al., 2019). Among these, *soft prompts* insert learnable continuous embeddings into the input and steer behavior through attention (Li & Liang, 2021; Lester et al., 2021); the paradigm is adopted for mathematical reasoning (Kang et al., 2026; Xu et al., 2025; Ye et al., 2026; Hao et al., 2025; Zhang et al., 2026). These methods diverge in injection position, training objective, and prompt form, and prior work attributes their gains to the *learned content* of the optimized vectors. The act of injecting embeddings into the input, common to all variants, has remained outside the analysis. Although theory shows trained prefixes only bias attention in a fixed direction (Petrov et al., 2024), the source of the effect itself remains unsettled.

To separate *learned content* from the *act of injection*, we use Random Soft Prompts (RSPs) as a control — continuous vectors drawn from an isotropic Gaussian matching the embedding-table statistics, with no learning. The control turns out to be non-neutral. Applying a chat template to a base model not fine-tuned for it degrades reasoning, yet appending RSPs recovers up to $+29$ pp on Qwen2.5-Math-1.5B; if simple noise were merely shaking the model, accuracy should have worsened instead. The injection itself, independent of any learned content, alters model behavior.

We analyze training-free RSPs on LLM reasoning. With each rollout receiving an *independent* draw, a single RSP injection *branches* hidden states into diverse trajectories early on, which then *stabilize* as decoding proceeds. Empirically, RSPs share the early-decoding entropy signature of optimized methods (LTPO, TTSV, SoftCoT) despite carrying no learned content; Pass@N accumulates only under *independent* resampling; and the same input-side diversity composes with DAPO (Yu et al., 2025) training. Our contributions are: (i) RSP isolates the simplest form of *soft prompt* — training-free and freshly resampled per rollout — providing a unified lens for the structural effect of injection across variants that otherwise differ in training objective and prompt form, while still reaching accuracy comparable to optimized methods in several settings; (ii) a theoretical and

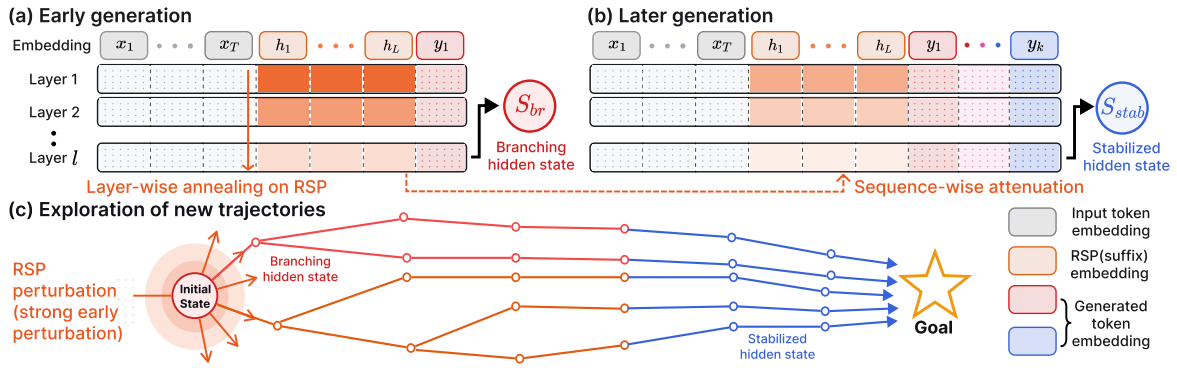


Figure 1. Conceptual overview of RSP-induced trajectory diversity. The hidden states shown are the final-layer outputs that drive the next-token distribution. (a) Early decoding: RSP perturbation reaches a *branching state* (red; formalized in §3.1), whose top-K differs from the no-RSP baseline and induces a diverse output distribution. (b) Later decoding: as the KV cache grows, the bound on RSP attention mass narrows (Theorem 3.2); branching events become rare and the hidden state (blue) commits to the branch already opened. (c) Across rollouts, the trajectories opened by early RSP perturbation accumulate into independent paths.

empirical account of the mechanism: attention-mediated branch opening, isotropic coverage across all directions, and the automatic attenuation of RSP attention as the KV cache grows; and (iii) DAPO extension showing independent RSP resampling composes with rollout-diversity training.

2. Background

2.1. Soft Prompt Methods for LLM Reasoning

Prior *soft prompt* methods share a common structure: they inject continuous vectors that do not correspond to discrete vocabulary tokens into the model's representation stream and *optimize* them under task-specific objectives. As parameter-efficient alternatives to full fine-tuning, Prefix-Tuning (Li & Liang, 2021), Prompt Tuning (Lester et al., 2021), and P-tuning (Liu et al., 2024) reached near-fine-tuning performance using learnable continuous vectors, after which LLM-reasoning variants followed. TTSV (Kang et al., 2026) optimizes *prefix* embeddings at test time via trajectory entropy minimization; SoftCoT (Xu et al., 2025) replaces explicit chain-of-thought tokens with soft tokens generated by an assistant model; LTPO (Ye et al., 2026) optimizes the *soft prompt* per problem to maximize its own confidence on that problem; COCONUT (Hao et al., 2025) feeds its hidden states back as input embeddings; MemGen (Zhang et al., 2026) uses embeddings as experience memory; and Pause tokens (Goyal et al., 2024) insert learnable tokens to provide additional computation steps.

These methods differ in position, training stage, and training target, yet share a common assumption: that the *learned representations* are the core driver of the effect. This paradigm faces several limitations. **(i)** Training is domain/benchmark-specific, requiring re-optimization for new settings and sensitive to seed/hyperparameter choices. **(ii)** Several have been observed to degrade on out-of-distribution challenging tasks (Ye et al., 2026). **(iii)** Evidence for the effect is largely empirical, with limited theoretical explanation beyond the training objective. Furthermore, soft prompting and prefix-tuning are theoretically restricted to biasing attention outputs in a fixed direction, without changing the relative attention pattern among content tokens (Petrov et al., 2024). Together, these findings cast doubt on whether the learned prompt vectors function as intended: they are suitable for *eliciting* or *combining* skills in the pretrained model, but they struggle to learn new capabilities.

Moreover, prior works have not separated the contribution of *injection itself* from that of *optimization*. In light of limitations (i)–(iii), this paper focuses on *injection itself*: we introduce *Random Soft Prompts* (RSPs) — vectors drawn from an isotropic Gaussian fitted to the pretrained embedding statistics, with the training stage entirely removed — and show that they produce output-distribution shifts and accuracy comparable to those of learned soft prompts, thereby disentangling the two contributions. Section 3 analyzes the mechanism theoretically; Section 4 validates it empirically.

2.2. Noise Injection in Neural Networks

Neural networks have incorporated noise at several stages and locations. During training, dropout (Srivastava et al., 2014) randomly deactivates *hidden activations*, and NEFTune (Jain et al., 2024) *adds* Gaussian noise to *input embeddings* during instruction fine-tuning to improve instruction following. At inference time, randomized smoothing (Cohen et al., 2019) injects Gaussian noise into *raw inputs* to obtain certified robustness, and SmoothLLM (Robey et al., 2025) perturbs prompts at the *discrete character* level to defend against jailbreaking. In RL, NoisyNet (Fortunato et al., 2018) *adds* learnable, trained noise to *network weights* to aid exploration. Rather than *adding* noise to embeddings — which alters existing token representations — RSP *ap-*

Table 1. Comparison with prior *soft prompt* methods (TTSV, SoftCoT, LTPO) under unified evaluation. Baseline and RSP values are from Table 2. **Bold** denotes the best result and underline denotes the second best for each model–benchmark pair.

Model	Benchmark	Baseline	RSP	TTSV	SoftCoT	LTPO
Qwen2.5-Math-7B-Instruct	MATH-500	83.20	**84.20**	83.80	82.80	83.00
	GSM8K	95.45	**95.68**	95.30	95.00	94.84
	AIME24	13.33	16.67	**23.30**	16.67	13.33
Qwen2.5-Math-1.5B-Instruct	MATH-500	73.00	75.20	75.20	**76.00**	72.40
	GSM8K	85.29	**85.75**	85.70	84.46	84.53
	AIME24	10.00	**20.00**	6.70	6.67	10.00

pends it as new positions, and differs in three respects: **(i)** unlike NoisyNet, it is entirely training-free, with no model fine-tuning and no learned noise parameters; **(ii)** it appends within a single forward pass, eliminating the multi-pass aggregation that robustness methods require; **(iii)** appending leaves the original tokens untouched, confining the perturbation to a controllable subset of attention keys whose influence is bounded by Theorem 3.2.

3. Random Soft Prompts and Why They Work

3.1. Random Soft Prompt

Random embedding vectors appended to LLM inputs — carrying no learned content — lift reasoning accuracy on math benchmarks and reach the range of optimized *soft prompt* methods (Table 1, §4.2). This section lays the theoretical groundwork for why a training-free distribution suffices, by tracking the *attention mass* w placed on RSP tokens. Early in decoding w is large enough to open a different decoding branch (§3.2); as the KV cache grows, the upper bound on w narrows automatically and the model commits to the branch already opened (§3.3) — an implicit explore-then-exploit. Independent resampling per rollout opens a fresh branch each time (proofs in Appendix E).

To isolate the act of injection from learned content while keeping the magnitude comparable to real tokens and not privileging any direction in embedding space, we draw RSP from an isotropic Gaussian fitted to the entrywise statistics of the pretrained embedding table. Let $\mathbf{W}_E \in \mathbb{R}^{V\times d}$ denote the pretrained token-embedding matrix, with V the vocabulary size and d the hidden dimension. Write its entrywise statistics as $\mu_E := \frac{1}{V\cdot d}\sum_{v,k}[\mathbf{W}_E]_{v,k}$ (the mean over all $V\cdot d$ entries) and $\sigma_E := \mathrm{std}(\mathbf{W}_E)$. An RSP of length L is a sequence of continuous vectors $\mathbf{H} = [h_1, \ldots, h_L] \in \mathbb{R}^{L\times d}$ drawn independently from

$$h_j \sim \mathcal{N}(\mu_E \mathbf{1}_d,\ \sigma_E^2 \mathbf{I}_d), \quad j = 1, \ldots, L. \tag{1}$$

The centered form $\bar{h}_j := h_j - \mu_E \mathbf{1}_d \sim \mathcal{N}(0, \sigma_E^2 \mathbf{I}_d)$ is a zero-mean isotropic Gaussian. Letting $\mathbf{X} \in \mathbb{R}^{T\times d}$ denote the input token-embedding sequence (the T rows of $\mathbf{W}_E$ corresponding to the prompt tokens), the main text uses the *suffix* form $[\mathbf{X}; \mathbf{H}] \in \mathbb{R}^{(T+L)\times d}$ as the default; other positions (*prefix* $[\mathbf{H}; \mathbf{X}]$, *infix* ($\mathbf{H}$ inserted within $\mathbf{X}$) (Kang et al., 2026; Xu et al., 2025; Ye et al., 2026; Zhang et al., 2026)) are reported as ablations in §4.2 and Appendix A. RSP is not a learnable parameter, so it receives no gradient and is freshly drawn for each rollout. We define decoding step t to be a *branching event* and the last-layer hidden state $\mathbf{s}^{(t)}$ to be a *branching state* when the LM-head top-K differs from that of the no-RSP baseline $\bar{\mathbf{s}}^{(t)}$ (Figure 1, red).[1]

3.2. Exploration: RSP strengthens early-stage exploration

Inserting a random prompt concentrates attention on it, and that concentration amplifies exploration. The effect of a single injection on the hidden state raises two questions — *how much* and *where* — decided respectively by the scalar w inside one attention head and by the distribution of $\mathbf{H}$. The isotropic Gaussian addresses both at once. Both questions trace back to the branching condition of §3.1: for a single RSP draw to shift top-K, its perturbation of the head output must be large enough to propagate through the layers (*how much*) and along directions that affect logit ordering (*where*).

Consider one attention head in self-attention layer ℓ. At query position i, the query attends $n \geq 1$ unmasked real tokens (n depends on i via the causal mask) together with $L \geq 1$ RSP tokens, all attention logits finite. Write the attention weights as $\alpha_{ij}^{(\ell)}$ (softmax-normalized so that $\sum_{j=1}^{n}\alpha_{ij}^{(\ell)} + \sum_{j=1}^{L}\alpha_{i,n+j}^{(\ell)} = 1$) and the value vectors on the real and RSP sides as $\mathbf{v}_j^{(\ell)}, \mathbf{v}_{r_j}^{(\ell)}$. Defining the total attention mass on random tokens $w_{r,i}^{(\ell)} := \sum_{j=1}^{L}\alpha_{i,n+j}^{(\ell)}$, the real-token mass $1 - w_{r,i}^{(\ell)}$ is positive and the head output $\mathbf{o}_i^{(\ell)}$ splits *exactly* into a renormalized real-token term $\tilde{\mathbf{o}}_i^{(\ell)}$ and an RSP-induced contribution $\boldsymbol{\eta}_i^{(\ell)}$:

$$\begin{aligned} \mathbf{o}_i^{(\ell)} &= (1 - w_{r,i}^{(\ell)})\,\tilde{\mathbf{o}}_i^{(\ell)} + \boldsymbol{\eta}_i^{(\ell)}, \\ \tilde{\mathbf{o}}_i^{(\ell)} &:= \sum_{j=1}^{n} \frac{\alpha_{ij}^{(\ell)}}{1 - w_{r,i}^{(\ell)}}\,\mathbf{v}_j^{(\ell)}, \quad \boldsymbol{\eta}_i^{(\ell)} := \sum_{j=1}^{L}\alpha_{i,n+j}^{(\ell)}\mathbf{v}_{r_j}^{(\ell)}. \end{aligned} \tag{2}$$

[1] Italic h_j: RSP input space; bold $\mathbf{s}^{(t)}$: hidden state. K denotes the cardinality in top-K analyses; nucleus sampling uses a separate threshold p. t for decoding step, ℓ for self-attention layer.

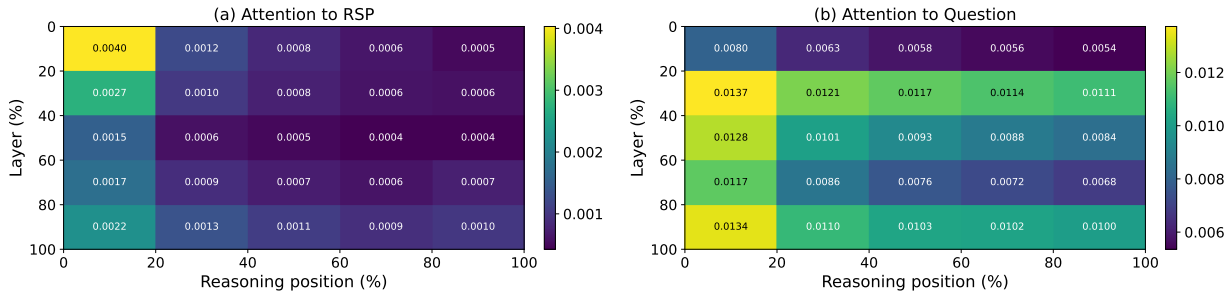


Figure 2. Per-token attention mass on Qwen2.5-Math-7B (*suffix*, 500 MATH-500 problems, independent RSP per problem). (a) RSP-token attention decreases along the reasoning-position axis, matching Theorem 3.2's KV cache-growth bound; the additional layer-axis decrease is an empirical phenomenon (deeper layers sharpen the real-vs-random gap) outside Theorem 3.2's scope. (b) Question-token attention stays comparatively uniform. Reported quantity: per-token mass of Appendix F, equal to $w_{r,i}^{(\ell)}/L$ averaged over heads and samples.

Derivation of Eq. (2). Write the head output as $\sum_{j=1}^{n} \alpha_{ij}^{(\ell)} \mathbf{v}_j^{(\ell)} + \sum_{j=1}^{L} \alpha_{i,n+j}^{(\ell)} \mathbf{v}_{r_j}^{(\ell)}$, then factor $1 - w_{r,i}^{(\ell)} = \sum_{j=1}^{n} \alpha_{ij}^{(\ell)} > 0$ out of the real-token sum, where $\sum_{j=1}^{n} \frac{\alpha_{ij}^{(\ell)}}{1-w_{r,i}^{(\ell)}} = 1$, so $\tilde{\mathbf{o}}_i^{(\ell)}$ is a valid weighted average over real-token value vectors. No approximation beyond the softmax-normalization identity is used. □

This single quantity $w_{r,i}^{(\ell)}$ controls both the attenuation ratio $1 - w_{r,i}^{(\ell)}$ on the real signal and the upper bound on the random contribution $\|\boldsymbol{\eta}_i^{(\ell)}\| \leq w_{r,i}^{(\ell)} \max_j \|\mathbf{v}_{r_j}^{(\ell)}\|$. The *magnitude* of RSP-induced variation thus reduces to one number in $[0, 1]$. Early in decoding the KV cache is short and this value is non-negligible, so a single RSP draw can perturb next-token decisions.

The scalar w controls *how much*; *where* is decided by the distribution of $\mathbf{H}$. For the local logit argument, isolate one RSP position and let $\bar{h} \in \mathbb{R}^d$ denote the i.i.d. marginal of any centered vector $\bar{h}_j$ in Eq. (1), with the other centered RSP positions fixed at zero. Let $z_a(\bar{h})$ be the under-RSP output logit at vocab token a and write the vocab-logit gap $\Delta_{ab}(\bar{h}) := z_a(\bar{h}) - z_b(\bar{h})$ ($\bar{h} = 0$ is the centered mean, not the no-RSP state; Appendix D). The transformer's logit map is non-linear in $\bar{h}$, but a first-order Taylor expansion around $\bar{h} = 0$ gives the local surrogate $\Delta_{ab}(\bar{h}) \approx \Delta_{ab}(0) + b_{ab}^\top \bar{h}$, with $b_{ab} := \nabla_{\bar{h}}(z_a - z_b)|_{\bar{h}=0} \in \mathbb{R}^d$ the gradient direction along which vocab tokens a, b swap rank (not unit-norm in general). Because RSP is training-free, we cannot know in advance which b_{ab} opens a useful branch, so the distribution must avoid systematically under-perturbing any direction. A deterministic injection pins to one direction; vocabulary sampling inherits the embedding table's anisotropy. The remaining design — maximize the worst-case directional variance at fixed budget — is uniquely solved by isotropy.

Proposition 3.1 (Maximin directional coverage)**.** *Let $\mathcal{D}_\rho$ denote the family of zero-mean distributions on $\mathbb{R}^d$ whose covariance satisfies $\mathrm{tr}(\Sigma_D) \leq \rho^2$. For each $D \in \mathcal{D}_\rho$, $\min_{\|u\|=1} \mathrm{Var}_{h\sim D}(u^\top h) = \lambda_{\min}(\Sigma_D)$, and $\sup_{D\in\mathcal{D}_\rho} \lambda_{\min}(\Sigma_D) = \rho^2/d$, attained iff $\Sigma_D = (\rho^2/d)\mathbf{I}_d$.*

This is a *design criterion* for the prompt law, not a guarantee of correctness; correctness enters through the task-side $p_{\min}(x)$ assumption of §3.3 and independent resampling.[2]

Gaussian RSP attains the equality with $\Sigma = \sigma_E^2 \mathbf{I}_d$ ($\rho^2 = d\sigma_E^2$) and adds *full support* on $\mathbb{R}^d$ (Appendix D). Isotropy excludes no direction; full support gives positive probability to the open branching-event region (§3.1) whenever it is non-empty (Appendix E.3.3). Monotone temperature preserves ranking and cannot reach this region, so RSP's *rank-changing* branching follows from the two properties together, splitting the hidden state into multiple branching states early in reasoning.

3.3. Annealing: KV cache growth dilutes RSP attention

The early branching stabilizes as decoding proceeds: the upper bound on $w_{r,i}^{(\ell)}$ narrows along the sequence axis with autoregressive cache growth. Each step adds one real-token term while RSP terms stay fixed; at frozen attention-logit gap, this asymmetry yields the following bound.

Theorem 3.2 (Attention-mass decay under KV cache growth)**.** *Consider a single attention head with per-head key dimension d_{head}, where query $\mathbf{q}_i$ at position i attends $n \geq 1$ real keys $\mathbf{k}_j$ and $L \geq 1$ RSP keys $\mathbf{k}_{r_{j'}}$ under finite logits. Define the pre-scale* attention-logit *gap $\Delta_i := \min_{j\in[n]} \mathbf{q}_i^\top \mathbf{k}_j - \max_{j'\in[L]} \mathbf{q}_i^\top \mathbf{k}_{r_{j'}}$ (distinct from the vocab-logit gap Δ_{ab} of §3.2). In scaled dot-product attention,*

$$w_{r,i} \leq \frac{L}{L + n \exp(\Delta_i/\sqrt{d_{head}})},$$

[2]The maximin is stated in input space; isotropy is not claimed to survive transformer layers. Any logit direction pulls back via the model Jacobian to an input-space b, and isotropy guarantees only that no such pulled-back direction (in the role of u above) has zero projected variance.

Table 2. Accuracy (%) across model configurations and benchmarks. Values in parentheses indicate the difference from the baseline. Full per-position breakdown is in Appendix A.

	MATH-500			GSM8K			AIME24		
Model	Baseline	RSP(*prefix*)	RSP(*suffix*)	Baseline	RSP(*prefix*)	RSP(*suffix*)	Baseline	RSP(*prefix*)	RSP(*suffix*)
Instruct Models									
LLaMA-3.1-8B-Inst	52.20	45.00 (−7.2)	49.40 (−2.8)	85.75	76.35 (−9.4)	86.73 (+1.0)	6.67	3.33 (−3.3)	10.00 (+3.3)
Qwen2.5-Math-7B-Inst	83.20	81.40 (−1.8)	83.40 (+0.2)	95.45	94.92 (−0.5)	95.68 (+0.2)	13.33	13.33 (+0.0)	16.67 (+3.3)
Qwen2.5-Math-1.5B-Inst	73.00	74.80 (+1.8)	74.20 (+1.2)	85.29	85.29 (+0.0)	84.69 (−0.6)	10.00	13.33 (+3.3)	20.00 (+10.0)
Base Models + ChatML (Format Mismatch)									
Qwen2.5-Math-7B	52.20	59.20 (+7.0)	70.40 (+18.2)	58.30	56.41 (−1.9)	75.97 (+17.7)	23.33	3.33 (−20.0)	23.33 (+0.0)
Qwen2.5-Math-1.5B	34.40	63.40 (+29.0)	51.00 (+16.6)	37.45	67.02 (+29.6)	50.64 (+13.2)	13.33	20.00 (+6.7)	13.33 (+0.0)

and the right-hand side is strictly decreasing in n and tends to 0 at fixed L and Δ_i.

Proof. Let $s_j := \mathbf{q}_i^\top \mathbf{k}_j/\sqrt{d_{\text{head}}}$, $s_{r_{j'}} := \mathbf{q}_i^\top \mathbf{k}_{r_{j'}}/\sqrt{d_{\text{head}}}$, $s_{r,\max} := \max_{j'} s_{r_{j'}}$, and $R := \sum_{j'=1}^{L} \exp(s_{r_{j'}})$. The gap definition gives $s_j \geq \Delta_i/\sqrt{d_{\text{head}}} + s_{r,\max}$ for every real j, and $\exp(s_{r,\max}) \geq R/L$ holds because the max dominates the average; combining the two and summing over the n real keys yields $\sum_{j=1}^{n} \exp(s_j) \geq (n/L)\exp(\Delta_i/\sqrt{d_{\text{head}}})\, R$. Substituting into $w_{r,i} = R/(\sum_j \exp(s_j) + R)$ gives

$$w_{r,i} \leq \frac{R}{(n/L)\exp(\Delta_i/\sqrt{d_{\text{head}}})\, R + R} = \frac{L}{L + n\,\exp(\Delta_i/\sqrt{d_{\text{head}}})}.$$

The derivative of the right-hand side in n is $-L\exp(\Delta_i/\sqrt{d_{\text{head}}})/(L + n\exp(\Delta_i/\sqrt{d_{\text{head}}}))^2 < 0$, so the bound is strictly decreasing and tends to 0 as $n \to \infty$. □

Applied at each layer ℓ to $w_{r,i}^{(\ell)}$ with gap $\Delta_i^{(\ell)}$, L caps the maximum influence while n grows each step. The theorem guarantees only sequence-axis attenuation at fixed (layer, query, gap); the gap may sharpen at deeper layers, as suggested empirically by Figure 2, but that effect is outside the theorem's scope. The accurate reading is "the attainable upper bound at a fixed gap decreases monotonically." In practice, Δ_i itself tends to widen along generation: real keys gain alignment with the query through accumulated semantic context, while RSP keys — isotropic random — carry no such learned alignment, tightening the same envelope further. Because Eq. (2) ties the random contribution to the same scalar, branching-event reachability inherits this envelope — as the cache grows, events become rare and the response commits to the branch already opened, an implicit explore-then-exploit. Figure 2 visualizes this pattern: RSP-token attention mass falls along both sequence and layer axes, while question-token attention stays comparatively stable — isolating the decay to the random injection rather than a generic attention property.

Lifting single-shot branching to task accuracy across N rollouts requires (M1) a positive lower bound $p_{\min}(x) > 0$ on the per-rollout probability that one execution reaches a correct trajectory and (M2) the N executions are independent. Under both, $\Pr(\text{Pass@}N \text{ on } x) \geq 1-(1-p_{\min}(x))^N$ (Appendix E.3.4). §3.2's isotropy and full support make local branch opening reachable when the open region is nonempty, but they do not by themselves guarantee the task-side correctness condition (M1); *independent resampling* supplies (M2). Sharing one RSP breaks (M2) and removes the accumulation — a signature §4 verifies empirically.

4. Empirical Validation

The mechanism of §3 runs from early *branching* through KV cache-growth narrowing to Pass@N accumulation under independent resampling. We verify this flow across three scenes. First, §4.2 tests whether RSPs preserve baseline accuracy and even recover it in some settings. Next, §4.3 examines whether the early diversification and later stabilization appear in entropy and attention signals. §4.4 asks whether the Pass@N gain depends on *independence* of the RSP draw. Finally, §4.5 extends the same perturbation from inference to RL training, where rollout diversity is critical.

4.1. Experimental Setup

We evaluate on three mathematical reasoning benchmarks, MATH-500 (Lightman et al., 2024), GSM8K (Cobbe et al., 2021), and AIME24, using LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024) and the Qwen2.5-Math model family (Yang et al., 2024) across instruct models and base models with mismatched chat templates. Tables 2 and 1 use greedy decoding to isolate the effect of RSPs from sampling stochasticity; the diversity and Pass@N experiments in §4.4 use the sampling settings specified there. RSPs follow the definition of §3.1, using the default *suffix* position with a freshly drawn $\mathbf{H}$ per rollout. Since Theorem 3.2 identifies the RSP length L as the design parameter capping the maximum perturbation influence, we ablate $L \in \{10, 15, 20\}$

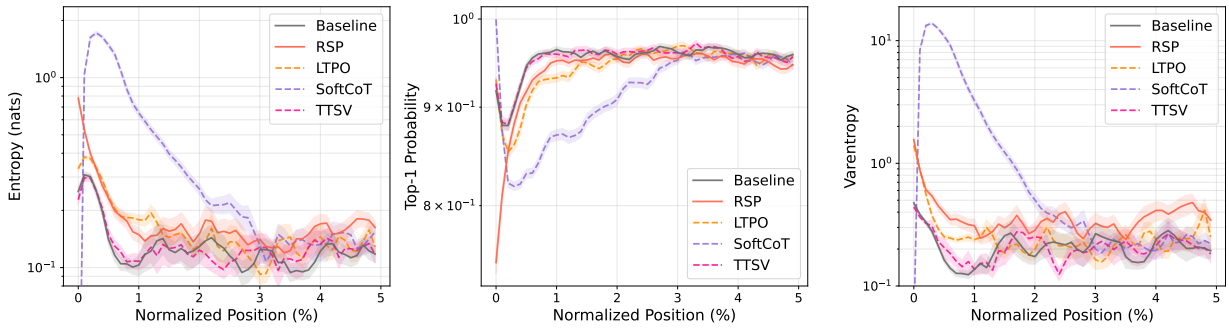


Figure 3. Mean entropy, top-1 probability, and varentropy during the first 5% of generation steps (Qwen2.5-Math-1.5B-Instruct, MATH-500). Shading indicates ± 1 SEM. Solid lines: RSP variants and the baseline. Dashed lines: prior *soft prompt* methods (LTPO, SoftCoT, TTSV).

per model to identify an appropriate perturbation strength. Tables 2 and 1 report results at the per-model selected length. The mechanism analyses in §4.3–§4.4 use a fixed setting (L=20, *suffix*), which insulates them from this selection effect. Full experimental details and per-position results are in Appendix A.

4.2. How Does Untrained RSP Compare to Baselines & Optimized Soft Prompts?

Compared with three optimized methods (TTSV, SoftCoT, LTPO) under a unified pipeline (Appendix A.2), RSP lands in the same accuracy band without any training, optimization, or task-specific tuning (Table 1, presented in §3). It wins outright on several model–benchmark cells and shows meaningful gains even on the challenging AIME24 setting. What we want to highlight is the converse: a fixed, training-free distribution reaching this range is the paper's central evidence that part of the *soft-prompt* effect comes less from learned content than from the random directional perturbation that injection induces — maximin coverage (§3.2) at a magnitude bounded by the KV cache (§3.3).

Beyond the direct comparison with optimized methods, we further evaluate RSP across two injection positions (*prefix*, *suffix*) and several model configurations, unpacking the per-configuration patterns of Table 2. On instruct models RSP stays within a few percentage points of the baseline. The most striking pattern is format-mismatch recovery: when a ChatML template is applied to a base model not trained for that format, *Suffix* recovers $+18.2/+17.7$ pp on Qwen2.5-Math-7B (MATH-500/GSM8K) and *prefix* recovers up to $+29.0/+29.6$ pp on Qwen2.5-Math-1.5B — if simple noise were responsible, accuracy should have dropped further.

RSP's *direct accuracy gains* concentrate in misaligned-input recovery and stay within $\pm$a few percentage points on well-aligned instruct models. The effects this paper centers on are the token-, trajectory-, and outcome-level diversity in §4.4, with accuracy recovery as a secondary outcome.

4.3. How Does RSP Affect the Output Distribution?

We next examine how RSP injection reshapes the output distribution. The setting is Qwen2.5-Math-1.5B-Instruct on MATH-500, and we measure per-token entropy, top-1 probability, and varentropy across generation steps. Figure 3 compares these metrics for the first 5% of generation steps across RSP variants and prior *soft prompt* methods (LTPO, SoftCoT, TTSV); full curves are in Appendix A.4.

Latent prompt methods including RSP share the same signature: *a rise in early-generation entropy followed by convergence to the baseline later in generation*. A natural mechanistic explanation is the out-of-vocabulary (OOV) effect — when continuous embeddings outside the vocabulary enter the input, the model has to attend to a never-before-seen key position on top of its familiar token distribution, redistributing probability mass across multiple candidates within the early next-token distribution. All three RSP injection positions show early-stage entropy↑, top-1 probability↓, and varentropy↑, a joint signature consistent with mass redistribution across multiple high-probability candidates, a precondition for the trajectory branching of §3.2. The change is confined to the early stage and all three metrics return to baseline in the middle and final portions, consistent with the *early influence with automatic KV cache-growth decay* pattern predicted in §3.3.

Prior methods trained under different objectives (LTPO, SoftCoT, TTSV) share this signature. Even TTSV, whose reward explicitly *lowers* mean reasoning-trajectory entropy, shows early-stage entropy comparable to the baseline. The entropy value itself is not a quality score — higher entropy is not always better; rather, the signature reveals a *shared mechanism* in which OOV embedding injection produces the same fingerprint independently of the optimization objective, supporting the hypothesis that the key driver of latent reasoning is the *act of injection*, not the learned content.

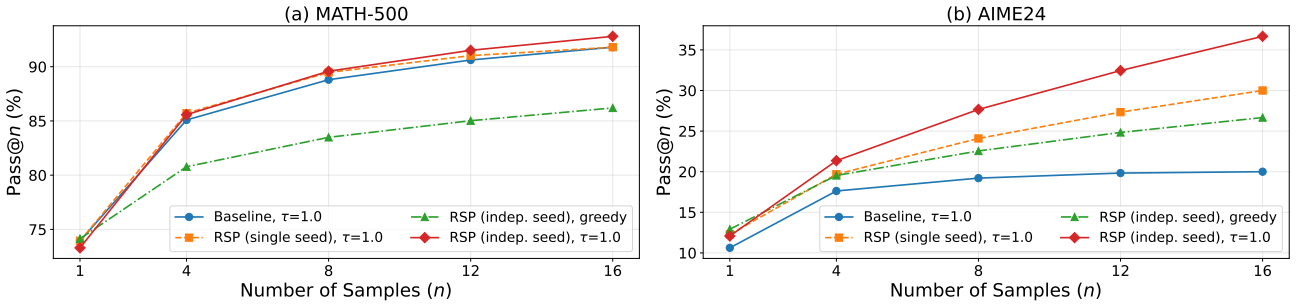


Figure 4. Pass@N scaling on (a) MATH-500 and (b) AIME24 with Qwen2.5-Math-1.5B-Instruct, 16 samples per problem. *Baseline*: temperature sampling only. *RSP (single seed)*: single RSP shared across samples combined with temperature. *RSP (indep. seed)*: a different RSP per sample, with or without temperature.

4.4. Does RSP Induce Trajectory Diversity?

Section 4.3 empirically established that RSP reshapes the early-stage output distribution. We now examine whether this shift translates into reasoning diversity at three complementary levels of analysis: token rankings, semantic content of trajectories, and task-level outcomes. The three analyses converge on a directional picture: RSP's first-token perturbation propagates into semantically more diverse reasoning trajectories, which in turn yields greater output diversity at the task level. All three analyses are conducted under a shared setting: Qwen2.5-Math-1.5B-Instruct, MATH-500, *suffix* injection, and $L = 20$.

Token-level: distribution beyond temperature. We quantify how much RSP shifts the first-token distribution — the critical forking point for reasoning trajectories (Wang et al., 2025). Using the baseline at τ=1.0 as reference, we compare baselines at τ=2.0, 3.0 (16 samples per problem) against RSP at τ=1.0 (averaged over 16 seeds) on three metrics: Spearman rank correlation ρ to detect rank reorganization, the probability mass placed outside the baseline's top-10 to measure support expansion, and Jensen–Shannon divergence to quantify the overall magnitude of distributional change.

Table 4. First-token distribution metrics measured against the baseline at τ=1.0, comparing baselines at higher temperatures (τ=2.0, 3.0) with RSP at τ=1.0. RSP metrics are averaged over 16 random seeds; Acc reports mean ± std across 16 samples per problem.

Metric	τ=2.0	τ=3.0	RSP
Spearman ρ	1.000	1.000	**0.709**
Mass outside top-10	5.18%	29.11%	**21.86%**
JS divergence	0.060	0.218	**0.131**
Acc (%)	20.77 ± 1.47	1.42 ± 0.35	**73.30 ± 1.07**

Table 4 shows the contrast: RSP's distributional shift falls between τ=2.0 and τ=3.0 in magnitude, but the mechanism differs — temperature is a monotone transform that preserves ranking ($z_a > z_b \iff p_a > p_b$ at any τ), while RSP partially preserves but reranks ($\rho = 0.709$). The Pass@1 row exposes the cost: matching RSP's magnitude with temperature toward τ=3.0 collapses accuracy to 1.42%, whereas RSP preserves 73.30%. Thus, RSP produces a fundamentally different perturbation than temperature scaling. Moreover, RSP's accuracy variability across 16 seeds at τ=1.0 is ±1.07, comparable to the ±1.47 that temperature sampling produces across 16 samples at τ=2.0 (detailed definitions in Appendix B).

Trajectory-level: semantic diversity. Token-level reranking raises a second question: do first-token perturbations diverge into distinct trajectories, or converge to similar ones? We sample 64 problems from MATH-500 and generate 300 independent trajectories per problem under Baseline and RSP at $\tau = 1.0$. Each trajectory is encoded by BGE-M3 (Chen et al., 2024) into a dense semantic vector, and we compute three metrics: *pairwise cosine distance* (overall semantic spread), *inter-cluster distance* between centroids of DBSCAN (Ester et al., 1996), a density-based clustering algorithm (separation between distinct trajectory groups), and *intra-cluster distance* (coherence within groups).

Table 5. Semantic diversity metrics (64 problems, 300 trajectories per condition).

Metric	Baseline	RSP
Pairwise cos. dist.	0.0612	**0.0879**
Inter-cluster dist.	0.0183	**0.0982**
Intra-cluster dist.	0.0526	0.0528

Table 5 reveals three patterns: pairwise distance rises by 1.4×, inter-cluster distance jumps by 5.4×, and intra-cluster distance is essentially unchanged — the trajectory set becomes more diverse and splits into more separated groups while preserving within-group coherence. RSP also yields larger pairwise distance than baseline on 56 of 64 problems.

Table 3. Per-benchmark accuracy (%) at each method's peak step on Qwen2.5-Math-7B. **Bold** denotes the better of the two RL methods on each benchmark. Avg is the unweighted five-benchmark mean.

Method	GSM8K	MATH-500	College	Minerva	AIME24	Avg
Base	57.2	52.6	18.3	13.6	8.6	30.06
DAPO	86.3	78.2	41.4	37.5	22.6	53.20
DAPO + RSP	**87.7**	**78.6**	**42.2**	**39.7**	**23.6**	**54.36**

Outcome-level: Pass@N scaling. Does this token- and trajectory-level diversity translate into task-level outcome diversity, measured by Pass@N (Chen et al., 2021) (the probability that at least one of N sampled solutions is correct)? We generate 16 samples per problem on MATH-500 and AIME24 and compare four conditions: *(i) Baseline* at $\tau = 1.0$; *(ii) RSP (single seed)* at $\tau = 1.0$, with one RSP shared across all samples; *(iii) RSP (indep. seed)* with greedy decoding, a fresh RSP per sample; and *(iv) RSP (indep. seed)* at $\tau = 1.0$, a fresh RSP per sample.

Condition *(iv)* achieves the highest Pass@N on both benchmarks, reaching 92.80% on MATH-500 and 36.67% on AIME24 at $N = 16$, while *(ii)* shows no improvement over the baseline — the diversity gain depends on varying the embeddings across samples rather than fixing a single perturbation. Pass@1 stays comparable to the baseline on MATH-500, and on the harder AIME24 *(iv)* surpasses it, suggesting that independent RSPs combined with temperature sampling expand trajectory diversity without sacrificing accuracy. Qualitative analysis is in Appendix C.

4.5. Application: DAPO Training with RSP

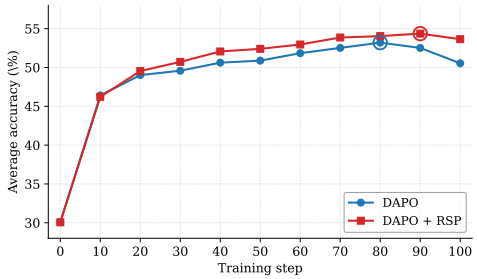


Figure 5. Five-benchmark average accuracy on Qwen2.5-Math-7B. DAPO + RSP reaches a higher peak (step 90) and stays stable through step 100.

Beyond inference (§4), does the same effect transfer to training? DAPO (Yu et al., 2025) is a recent GRPO (Shao et al., 2024) variant that preserves rollout diversity at the loss level through asymmetric clipping. We use it as a testbed to verify whether input-side branch opening composes with loss-side preservation to yield further gains. In DAPO+RSP, we incorporate the input-side perturbation of §4.4 into the RL rollout stage: each rollout is paired with an independently sampled suffix RSP. The full objective and implementation details are in Appendix G.

We implement DAPO on top of SimpleRL-Zoo (Zeng et al., 2025) and train Qwen2.5-Math-7B on a MATH Level-3–5 subset (~8.5K prompts): each step uses $G = 8$ rollouts per prompt at τ=1.0, batch 1,024, 4 PPO mini-batches of 256, for 100 steps. *DAPO* is compared against *DAPO + RSP*, where *suffix* RSP (L=20) is applied during rollouts only and evaluation uses no RSP. We evaluate on five math benchmarks (GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2024), College Math, Minerva Math (Lewkowycz et al., 2022), AIME24) and report the unweighted mean; AIME24 uses Avg@32 at τ=1.0, the others greedy.

Figure 5 and Table 3 show two patterns: a late-stage advantage and a delayed early reward growth. DAPO + RSP reaches 54.36% at step 90 (vs DAPO's 53.20% peak, $+1.16$ pp) and widens the gap to $+3.10$ pp at step 100 with DAPO's post-peak decline delayed; earlier the curves cross around step 20. The patterns follow from the methods acting at separate stages of the rollout cycle: DAPO preserves diversity at the loss level over already-generated rollouts while RSP perturbs the hidden states that produce them (§3.2), exposing the policy to broader learning signals. Empirically, this is the exploration–exploitation dynamics of RL (Sutton & Barto, 2018) induced from the input side.

5. Conclusion

We studied Random Soft Prompts (RSPs), training-free isotropic Gaussian vectors whose per-layer contribution is bounded by an attention-mass scalar that shrinks with KV cache growth and whose isotropic resampling reaches top-K entry regions inaccessible to rank-preserving temperature (§3). Empirically, RSP shares the early-decoding entropy signature of optimized *soft prompts*, requires independent resampling for Pass@N gains, and composes with DAPO training (§4–§4.5), suggesting that part of what trained *soft prompts* deliver is a *structural by-product of injection itself*.

Several directions invite future work. Open questions include whether the mechanism extends beyond RoPE-based decoders and math reasoning, whether RSP admits a continuous control dial like temperature sampling (e.g., via the RSP norm or its mixing ratio with real tokens), and whether the same diversity gains transfer to tasks with multiple valid answers such as code generation or open-ended reasoning. A theoretical account of the layer-axis decay outside Theorem 3.2's envelope would broaden the framework.

Acknowledgements

This work was supported by IITP (RS-2024-00445087, 50%; RS-2024-00436680, 40%, No. 2019-0-01842; 10%) funded by the Ministry of Science and ICT, Korea. Computing resource is supported by AICA, GIST SCENT-AI, and Advanced GPU Utilization Support Program (No. 02-26-01-0181).

References

Belrose, N., Ostrovsky, I., McKinney, L., Furman, Z., Smith, L., Halawi, D., Biderman, S., and Steinhardt, J. Eliciting Latent Predictions from Transformers with the Tuned Lens. *arXiv:2303.08112*, 2023.

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., and Liu, Z. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of ACL*, 2024.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating Large Language Models Trained on Code. *arXiv:2107.03374*, 2021.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training Verifiers to Solve Math Word Problems. *arXiv:2110.14168*, 2021.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified Adversarial Robustness via Randomized Smoothing. In *ICML*, 2019.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD*, 1996.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. Noisy Networks for Exploration. In *ICLR*, 2018.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer Feed-Forward Layers Are Key-Value Memories. In *EMNLP*, 2021.

Goyal, S., Ji, Z., Rawat, A. S., Menon, A. K., Kumar, S., and Nagarajan, V. Think before You Speak: Training Language Models with Pause Tokens. In *ICLR*, 2024.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z.,

Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., De Paola, B., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Zhang, G., Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Wang, Y., Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The Llama 3 Herd of Models. *arXiv:2407.21783*, 2024.

Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., and Tian, Y. Training Large Language Models to Reason in a Continuous Latent Space. In *ICLR*, 2025.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-Efficient Transfer Learning for NLP. In *ICML*, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022.

Jain, N., Chiang, P.-y., Wen, Y., Kirchenbauer, J., Chu, H.-M., Somepalli, G., Bartoldson, B. R., Kailkhura, B., Schwarzschild, A., Saha, A., Goldblum, M., Geiping, J., and Goldstein, T. NEFTune: Noisy Embeddings Improve Instruction Finetuning. In *ICLR*, 2024.

Kang, X., Shi, D., and Chen, L. Model Whisper: Steering Vectors Unlock Large Language Models' Potential in Test-time. In *AAAI*, 2026.

Lester, B., Al-Rfou, R., and Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. In *EMNLP*, 2021.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. Solving Quantitative Reasoning Problems with Language Models. In *NeurIPS*, 2022.

Li, X. L. and Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *ACL-IJCNLP*, 2021.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's Verify Step by Step. In *ICLR*, 2024.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. GPT Understands, Too. *AI Open*, 5:208–215, 2024.

Madaan, A. and Yazdanbakhsh, A. Text and Patterns: For Effective Chain of Thought, It Takes Two to Tango. *arXiv:2209.07686*, 2022.

Petrov, A., Torr, P. H. S., and Bibi, A. When Do Prompting and Prefix-Tuning Work? A Theory of Capabilities and Limitations. In *ICLR*, 2024.

Robey, A., Wong, E., Hassani, H., and Pappas, G. J. SmoothLLM: Defending Large Language Models Against Jailbreaking Attacks. *Transactions on Machine Learning Research*, 2025.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., and Guo, D. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv:2402.03300*, 2024.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. HybridFlow: A Flexible and Efficient RLHF Framework. In *EuroSys*, 2025.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

Wang, S., Yu, L., Gao, C., Zheng, C., Liu, S., Lu, R., Dang, K., Chen, X., Yang, J., Zhang, Z., Liu, Y., Yang, A., Zhao, A., Yue, Y., Song, S., Yu, B., Huang, G., and Lin, J. Beyond the 80/20 Rule: High-Entropy Minority Tokens Drive Effective Reinforcement Learning for LLM Reasoning. In *NeurIPS*, 2025.

Xu, Y., Guo, X., Zeng, Z., and Miao, C. SoftCoT: Soft Chain-of-Thought for Efficient Reasoning with LLMs. In *ACL*, 2025.

Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., Lu, K., Xue, M., Lin, R., Liu, T., Ren, X., and Zhang, Z. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. *arXiv:2409.12122*, 2024.

Ye, W., Liang, Y., and Shan, L. Thinking on the Fly: Test-Time Reasoning Enhancement via Latent Thought Policy Optimization. In *ICLR*, 2026.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, J., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M., Zhang, R., Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C., Yu, H., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W.-Y., Zhang, Y.-Q., Yan, L., Wu, Y., and Wang, M. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. In *NeurIPS*, 2025.

Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., and He, J. SimpleRL-Zoo: Investigating and Taming Zero Reinforcement Learning for Open Base Models in the Wild. In *COLM*, 2025.

Zhang, G., Fu, M., and Yan, S. MemGen: Weaving Generative Latent Memory for Self-Evolving Agents. In *ICLR*, 2026.

A. Experimental Setup and Full Accuracy Breakdown

All experiments are conducted on a single NVIDIA A6000 40GB GPU with greedy decoding, a maximum generation length of 3,072 tokens for MATH-500 and GSM8K, and 4,096 tokens for AIME24. Batch size is fixed per model (16 for LLaMA-3.1-8B-Instruct and Qwen2.5-Math-7B variants, 32 for Qwen2.5-Math-1.5B variants). The RSP token count L is selected per (model, benchmark) pair from $\{10, 15, 20\}$ (Table 7); the per-position breakdown across *prefix* ($[\mathbf{H}; \mathbf{X}]$), *infix* ($\mathbf{H}$ inserted within $\mathbf{X}$), and *suffix* ($[\mathbf{X}; \mathbf{H}]$) is in Table 6. The RSP injection harness used for these experiments is included as anonymized supplementary material, with run commands documented in the accompanying README.

Table 6. Accuracy (%) across model configurations, injection positions, and benchmarks. Values in parentheses indicate the difference from the baseline. **Bold** denotes the best result and underline denotes the second best for each model–benchmark pair.

Model	Mode	MATH-500	GSM8K	AIME24
Instruct Models				
LLaMA-3.1-8B-Instruct	Baseline	**52.20**	85.75	6.67
	Prefix	45.00 (−7.2)	76.35 (−9.4)	3.33 (−3.3)
	Infix	49.40 (−2.8)	85.97 (+0.2)	**10.00** (+3.3)
	Suffix	49.40 (−2.8)	**86.73** (+1.0)	**10.00** (+3.3)
Qwen2.5-Math-7B-Instruct	Baseline	83.20	95.45	13.33
	Prefix	81.40 (−1.8)	94.92 (−0.5)	13.33 (+0.0)
	Infix	**84.20** (+1.0)	95.60 (+0.2)	**16.67** (+3.3)
	Suffix	83.40 (+0.2)	**95.68** (+0.2)	**16.67** (+3.3)
Qwen2.5-Math-1.5B-Instruct	Baseline	73.00	85.29	10.00
	Prefix	74.80 (+1.8)	85.29 (+0.0)	13.33 (+3.3)
	Infix	**75.20** (+2.2)	**85.75** (+0.5)	6.67 (−3.3)
	Suffix	74.20 (+1.2)	84.69 (−0.6)	**20.00** (+10.0)
Base Models (Raw Text)				
Qwen2.5-Math-7B	Baseline	**72.20**	84.15	6.67
	Prefix	71.60 (−0.6)	**84.31** (+0.2)	**16.67** (+10.0)
	Infix	63.20 (−9.0)	64.29 (−19.9)	**16.67** (+10.0)
	Suffix	55.60 (−16.6)	54.13 (−30.0)	13.33 (+6.7)
Qwen2.5-Math-1.5B	Baseline	61.40	**77.86**	**16.67**
	Prefix	**64.40** (+3.0)	74.30 (−3.6)	10.00 (−6.7)
	Infix	63.20 (+1.8)	73.92 (−3.9)	10.00 (−6.7)
	Suffix	54.60 (−6.8)	58.61 (−19.3)	3.33 (−13.3)
Base Models + ChatML (Format Mismatch)				
Qwen2.5-Math-7B	Baseline	52.20	58.30	**23.33**
	Prefix	59.20 (+7.0)	56.41 (−1.9)	3.33 (−20.0)
	Infix	62.00 (+9.8)	69.07 (+10.8)	**23.33** (+0.0)
	Suffix	**70.40** (+18.2)	**75.97** (+17.7)	**23.33** (+0.0)
Qwen2.5-Math-1.5B	Baseline	34.40	37.45	13.33
	Prefix	**63.40** (+29.0)	**67.02** (+29.6)	**20.00** (+6.7)
	Infix	39.20 (+4.8)	50.42 (+13.0)	6.67 (−6.7)
	Suffix	51.00 (+16.6)	50.64 (+13.2)	13.33 (+0.0)

Table 7. Number of RSP tokens (L) used for each model and benchmark.

Model	MATH-500	GSM8K	AIME24
LLaMA-3.1-8B-Instruct	15	20	15
Qwen2.5-Math-7B-Instruct	20	20	20
Qwen2.5-Math-1.5B-Instruct	20	10	20
Qwen2.5-Math-7B	10	10	20
Qwen2.5-Math-1.5B	10	10	20
Qwen2.5-Math-1.5B (ChatML)	20	20	10
Qwen2.5-Math-7B (ChatML)	20	20	20

A.1. RSP Injection Positions and Prompt Templates

Below we show the prompt structure for each injection position across all model types. Blue text indicates RSP embeddings, and purple text indicates special tokens.

A.1.1. PREFIX

RSP embeddings are concatenated before the prompt embeddings. No text modification is applied.

ChatML (Qwen Instruct / Base + ChatML).

```
[Random Embeddings (L tokens)]
<|im_start|>system
Please reason step by step, and put your final answer within \boxed{}.<|im_end|>
<|im_start|>user
{question}<|im_end|>
<|im_start|>assistant
```

LLaMA Chat Template.

```
[Random Embeddings (L tokens)]
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
Cutting Knowledge Date:  December 2023
Today Date:  26 Jul 2024
Please reason step by step, and put your final answer within \boxed{}.<|eot_id|>
<|start_header_id|>user<|end_header_id|>
{question}<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

Raw Text (Qwen Base).

```
[Random Embeddings (L tokens)]
{question}
Please reason step by step, and put your final answer within \boxed{}.
```

A.1.2. INFIX

A description text and special tokens are inserted into the prompt. The special token positions are then replaced with RSP embeddings at the embedding level.

ChatML (Qwen Instruct / Base + ChatML).

```
<|im_start|>system
Please reason step by step, and put your final answer within \boxed{}.<|im_end|>
<|im_start|>user
{question}
There are {L} special tokens that contain compressed latent reasoning information
that might be useful for your reasoning.  If these tokens are useful for your case,
you can use them as reference.  If these tokens are not useful for your case, you
can ignore them and focus back to solving the problem.
Here are the {L} special tokens:  <special_token_1>...<special_token_L>
<|im_end|>
<|im_start|>assistant
```

LLaMA Chat Template.

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
Cutting Knowledge Date:  December 2023
```

```
Today Date:  26 Jul 2024
Please reason step by step, and put your final answer within \boxed{}.<|eot_id|>
<|start_header_id|>user<|end_header_id|>
{question}
There are {L} special tokens that contain compressed latent reasoning information
that might be useful for your reasoning.  If these tokens are useful for your case,
you can use them as reference.  If these tokens are not useful for your case, you
can ignore them and focus back to solving the problem.
Here are the {L} special tokens:
<reserved_special_token_0>...
<reserved_special_token_L>
<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

Raw Text (Qwen Base).

```
{question}
There are {L} special tokens that contain compressed latent reasoning information
that might be useful for your reasoning.  If these tokens are useful for your case,
you can use them as reference.  If these tokens are not useful for your case, you
can ignore them and focus back to solving the problem.
Here are the {L} special tokens:  <special_token_1>...<special_token_L>
Please reason step by step, and put your final answer within \boxed{}.
```

A.1.3. SUFFIX

For chat-templated models, RSP embeddings are inserted immediately before the end-of-turn token. For raw text models, RSP embeddings are concatenated at the end of the prompt.

ChatML (Qwen Instruct / Base + ChatML).

```
<|im_start|>system
Please reason step by step, and put your final answer within \boxed{}.<|im_end|>
<|im_start|>user
{question}[Random Embeddings (L tokens)]<|im_end|>
<|im_start|>assistant
```

LLaMA Chat Template.

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
Cutting Knowledge Date:  December 2023
Today Date:  26 Jul 2024
Please reason step by step, and put your final answer within \boxed{}.<|eot_id|>
<|start_header_id|>user<|end_header_id|>
{question}[Random Embeddings (L tokens)]<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

Raw Text (Qwen Base).

```
{question}
Please reason step by step, and put your final answer within \boxed{}.[Random
Embeddings (L tokens)]
```

A.2. Answer Extraction and Grading

The final answer is extracted from the model output through a fallback chain. Each step is attempted in order; if extraction fails, the next step is tried.

Answer Extraction Fallback Chain

1. `"final answer is $...$. I hope"` pattern (Minerva format)
2. `\boxed{...}`: last non-empty box is used; empty `\boxed{}` are skipped
3. Text following `"the answer is"`
4. Text following `"final answer is"`
5. Last number in the output (regex fallback)

Extracted answers are normalized by removing `$`, `\left`, `\right`, and unit strings. Grading is performed via symbolic equivalence checking: exact string match, numerical comparison (`math.isclose`, `rel_tol=1e-4`), and SymPy symbolic simplification.

A.3. Reproduced Prior Work Hyperparameters

All prior methods are reproduced on Qwen2.5-Math-1.5B-Instruct and Qwen2.5-Math-7B-Instruct, evaluated with greedy decoding (temperature $= 0$) and our unified answer extraction pipeline.

TTSV. *Prefix* length 20, trained for 20 epochs with AdamW optimizer, batch size 2, gradient accumulation 8 steps, and linear learning rate schedule. Learning rate is 1e-3 for Qwen-1.5B and 1e-5 for Qwen-7B.

LTPO. Table 8 reports the per-benchmark hyperparameters.

Table 8. LTPO hyperparameters.

Parameter	GSM8K	MATH-500	AIME24
tokens	8	8	8
steps	20	20	20
top_k	10	10	10
sigma	20	20	5
sigma_decay	0.95	0.95	0.95
lr	1e-2	5e-3	5e-2

SoftCoT. Projection module trained for 10 epochs. Thought tokens: 32 during training, 4 during evaluation. Batch size 4 for GSM8K, 1 for MATH with gradient accumulation 4. The small (assistant) model is Qwen2.5-Math-1.5B-Instruct in both configurations, while the large model is Qwen2.5-Math-1.5B-Instruct (learning rate 2e-5) or Qwen2.5-Math-7B-Instruct (learning rate 5e-6). For MATH-500 and AIME24 evaluation, we use the projection module trained on GSM8K, as it yields higher accuracy than the one trained on MATH.

A.4. Full Entropy Curves

Figure 6 shows the full normalized (0–100%) generation trajectories for entropy, top-1 probability, and varentropy, and Table 9 reports the corresponding scalar statistics including overall means, first-5% means, and the average number of generated tokens per problem. All methods converge to similar levels after the initial generation stage, confirming that the distributional changes induced by RSP are localized to the early reasoning phase.

Table 9. Per-step statistics for Qwen2.5-Math-1.5B-Instruct on MATH-500 across the full generation and the first 5% of generation steps, together with the average number of generated tokens per problem. Top-1 Prob (overall) is reported as a weighted average over the first-10%/middle/last-10% segment means with weights $0.1/0.8/0.1$. Extension of Figure 3 (main text) and Figure 6.

Metric	Baseline	*Prefix*	*Infix*	*Suffix*	LTPO	SoftCoT	TTSV
Tokens (mean)	575.7	558.4	549.9	573.3	592.3	594.4	577.4
Entropy (first 5%)	0.1332	0.1366	0.1402	0.1941	0.1662	0.4218	0.1359
Entropy (overall)	0.0871	0.0881	0.0899	0.1049	0.0909	0.1059	0.0864
Top-1 Prob (first 5%)	0.9535	0.9523	0.9525	0.9373	0.9437	0.9156	0.9534
Top-1 Prob (overall)	0.9684	0.9681	0.9678	0.9647	0.9683	0.9651	0.9685
Varentropy (first 5%)	0.2181	0.2207	0.2463	0.4010	0.2953	2.2234	0.2174
Varentropy (overall)	0.1353	0.1375	0.1422	0.2038	0.1452	0.2404	0.1361

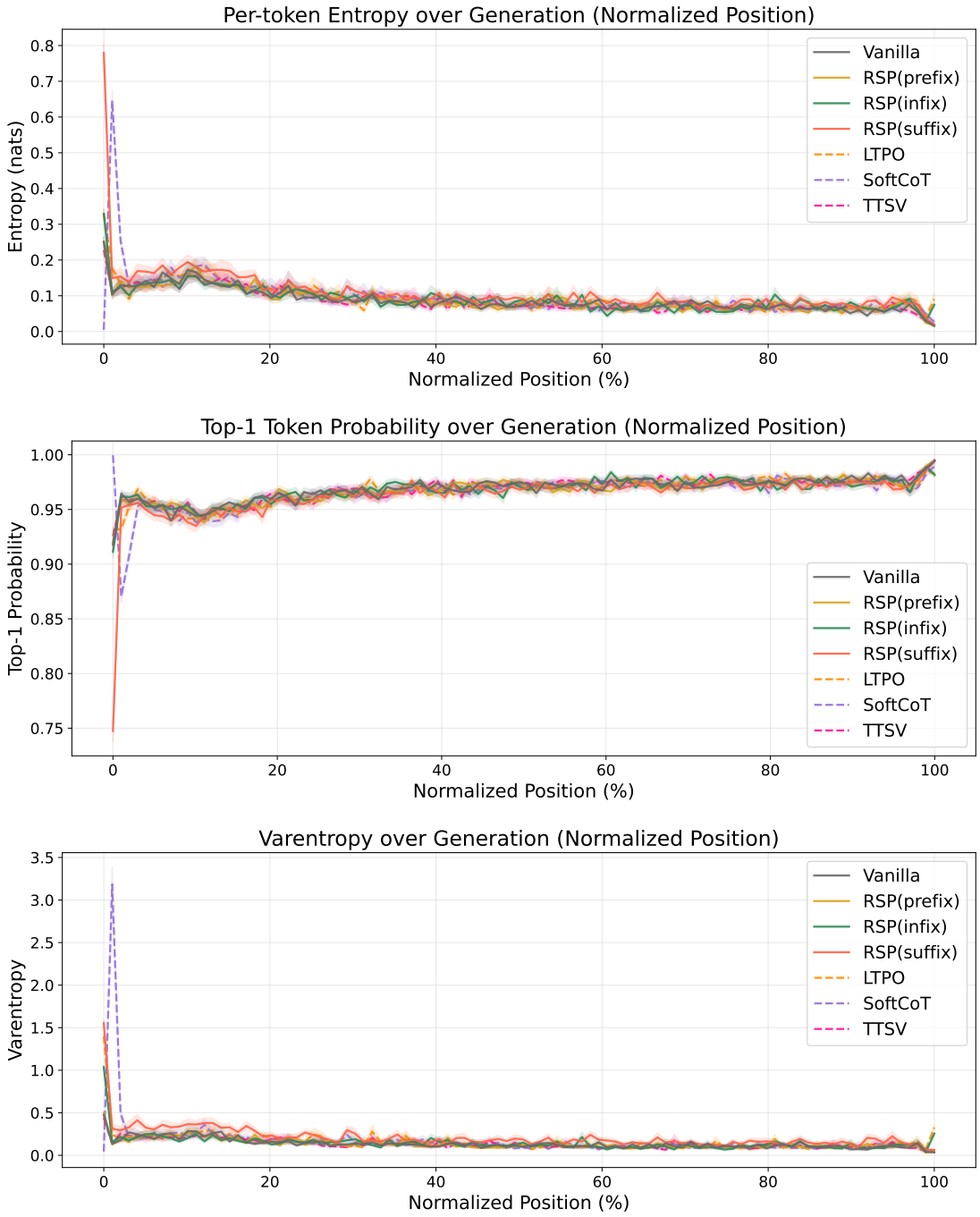


Figure 6. Mean entropy (top), top-1 probability (middle), and varentropy (bottom) over the full normalized generation trajectory (0–100%). Averaged over MATH-500, shading indicates ± 1 SEM. Solid lines: RSP variants and the baseline. Dashed lines: prior *soft prompt* methods. All methods converge after the initial stage.

B. Token-Level Distribution Metrics

This appendix provides formal definitions for the metrics used in Section 4.4. Numerical results are reported in Table 4 of the main text. We extract first-token logits via a single forward pass (no autoregressive generation) and store the top-100 token ids and logit values per problem. Temperature conditions reuse the baseline logits scaled by $1/\tau$, requiring no separate inference. RSP draws 16 independent seeds per problem (MATH-500), and reported metrics are the mean across the 16 seeds. Pass@1 accuracy is computed over 16 samples per problem, with the baseline at τ=1.0 reaching $73.92 \pm 0.78\%$.

Let P_{base} denote the baseline next-token distribution at τ=1.0 and P_{target} the candidate distribution being compared (e.g., baselines at τ=2.0, 3.0 or RSP at τ=1.0). All metrics are computed analytically from saved logits at the first generation step.

Spearman rank correlation. Spearman's ρ is the Pearson correlation coefficient between the token ranks of two distributions:

$$\rho \;=\; \mathrm{Pearson}(\mathrm{rank}(P_{\text{target}}),\ \mathrm{rank}(P_{\text{base}}))\,. \tag{3}$$

Because temperature scaling $P^{(\tau)} \propto P^{1/\tau}$ is a strictly monotone transform, the rank order of tokens is preserved exactly, so $\rho = 1$ for any $\tau > 0$ as a mathematical identity. Any value $\rho < 1$ therefore indicates rank reorganization beyond what temperature scaling can produce.

Mass outside top-K. Let $\mathrm{TopK}(P_{\text{base}})$ be the set of K tokens with the highest probability under P_{base}. The probability mass that the candidate distribution places *outside* this set is

$$\mathrm{MassOutside}_K \;=\; 1 - \sum_{t \in \mathrm{TopK}(P_{\text{base}})} P_{\text{target}}(t). \tag{4}$$

This directly measures support expansion: how much probability the candidate assigns to tokens that are not in the baseline's preferred set. Reported values use $K = 10$.

Jensen–Shannon divergence. The symmetrized, bounded version of Kullback–Leibler divergence:

$$\mathrm{JS}(P, Q) \;=\; \tfrac{1}{2}\,\mathrm{KL}(P \,\|\, M) + \tfrac{1}{2}\,\mathrm{KL}(Q \,\|\, M), \qquad M = \tfrac{1}{2}(P + Q). \tag{5}$$

We use base-2 logarithms so that $\mathrm{JS} \in [0, 1]$. Tokens absent from the stored top-100 are assigned a sentinel logit (-10^9) before softmax, which is negligible for our setting.

C. Qualitative Analysis: How RSP Reshapes Reasoning Trajectories

This appendix complements the outcome-level Pass@N results of §4.4 with a qualitative case study on AIME24, using the same N=16 sampling budget. We compare two of the four conditions defined in §4.4: *Baseline* (condition (i), τ=1.0) and *RSP (indep. seed)* (condition (iv), τ=1.0, *suffix*, L=20). Setting: Qwen2.5-Math-1.5B-Instruct, 30 AIME24 problems, 16 samples per problem per condition.

C.1. Aggregate Pattern Frequencies

Table 10 tabulates how RSP shifts coverage relative to Baseline across the 30 AIME24 problems.

Table 10. Coverage patterns across 30 AIME24 problems, N=16 samples per condition.

Pattern	Frequency	Problems
RSP-only correct (Baseline=0/16, RSP≥1/16)	5/30 (16.7%)	#6, #8, #14, #22, #25
Baseline-only correct (Baseline≥1/16, RSP=0/16)	0/30 (0%)	—
Both correct	6/30 (20.0%)	—
Both wrong	19/30 (63.3%)	—

Within the N=16 budget, RSP produces strict gains on five problems and never causes a strict regression. The two case studies below illustrate two faces of the same phenomenon: divergent solution framing (#6, §C.2) and divergent structural assumption (#22, §C.3). In both cases, RSP and Baseline diverge already in the opening sentences of the trajectory, and the divergence determines whether the trajectory reaches the correct answer.

C.2. Case Study 1: AIME24 Problem 6 (Constrained Optimization)

Problem.

Let $\mathcal{B}$ be the set of rectangular boxes with surface area 54 and volume 23. Let r be the radius of the smallest sphere that can contain each of the rectangular boxes that are elements of $\mathcal{B}$. The value of r^2 can be written as $\frac{p}{q}$, where p and q are relatively prime positive integers. Find $p + q$.

Ground truth: $\boxed{721}$, with the optimal box at $a = b = 2$, $c = 23/4$ giving $r^2 = 657/64$.

Aggregate result. Baseline produces 0/16 correct; RSP produces 1/16 correct. Baseline's 16 wrong final answers span 10 distinct values, with the modal answer 31 appearing in 4/16 samples; the correct value 721 does not appear. RSP's 16 samples produce 11 distinct values and include 721 exactly once.

Failure mode (Baseline). Baseline's representative trajectory (final answer 49) opens with an *algebraic identity* frame and never solves for the actual box dimensions.

Baseline trajectory (final answer 49) — excerpts

Opening: To solve the problem, we start by defining the dimensions of a rectangular box as a, b, and c. The surface area and volume of the box are given by $2(ab + bc + ca) = 54$ and $abc = 23$. We need to find the radius r of the smallest sphere that can contain the box, given by $r = \frac{1}{2}\sqrt{a^2 + b^2 + c^2}$. First, we need to find the expression for $a^2 + b^2 + c^2$ in terms of the given equations.
Identity manipulation: Squaring the surface area equation yields $(ab + bc + ca)^2 = 2916$, which expands to $a^2b^2 + b^2c^2 + c^2a^2 + 2abc(a + b + c) = 2916$. Substituting $2abc = 46$ and $(a + b + c)^2 = x + 54$ where $x := a^2 + b^2 + c^2$, then attempting to express $a^4 + b^4 + c^4$ in terms of x, the manipulation collapses to the tautology $x^2 = x^2 - 2y + 2y$.
Termination by guess: "This equation is always true, so we need another way to find x. ... After solving, we find $x = 45$." Thus $r^2 = 45/4$, and $p + q = 45 + 4 = \boxed{49}$.

Diagnosis. The trajectory frames the task as *find the closed form of* $a^2 + b^2 + c^2$ *from the given constraints* and never attempts to solve for the actual dimensions. Two consecutive identity manipulations collapse to a tautology that does not yield x, after which the trajectory commits to $x = 45$ without justification ("After solving, we find"). This pattern — identity manipulation, tautology, terminal guess — recurs across Baseline's 16 samples; the constraint set under-determines $a^2 + b^2 + c^2$ in closed form, and the algebraic-identity framing has no path to the correct value.

Success mode (RSP). The correct RSP trajectory (final answer 721) opens with an *optimization* frame and a symmetry assumption that reduces the system to a single cubic.

RSP trajectory (final answer 721) — excerpts

Opening: To solve the problem, we need to find the dimensions of the rectangular box that *minimizes* the radius of the smallest sphere that can contain it, given the surface area and volume constraints. The radius is $r = \frac{1}{2}\sqrt{a^2 + b^2 + c^2}$. To minimize r, we minimize $a^2 + b^2 + c^2$.
Symmetry reduction: A more straightforward approach is to use symmetry: $a^2 + b^2 + c^2$ is minimized when a, b, c are as close to each other as possible. Assume $a = b$ and solve for c: $2a^2 + 2ac = 27$ and $a^2c = 23$. From $a^2c = 23$, $c = 23/a^2$. Substituting: $2a^2 + 46/a = 27$, multiplying through by a gives $2a^3 - 27a + 46 = 0$.
Rational root and termination: By the Rational Root Theorem, $a = 2$ is a root. Then $c = 23/4$, $a^2+b^2+c^2 = 4+4+529/16 = 657/16$, $r^2 = 657/64$, and $p + q = \boxed{721}$.

Diagnosis. Three properties of the opening drive the success: **(i)** the task is framed as *minimizing* r (and thus $a^2 + b^2 + c^2$) rather than as a closed-form expression hunt; **(ii)** the path is committed to solving for a, b, c instead of for $a^2 + b^2 + c^2$ alone; **(iii)** a symmetry assumption $a = b$ reduces the system to one cubic in one unknown. The cubic $2a^3 - 27a + 46 = 0$ has $a = 2$ as a rational root, leading to the explicit configuration $(2, 2, 23/4)$ and the correct $r^2 = 657/64$.

Interpretation. Baseline's failure is rooted in the *opening sentence*: framing the task as "find the expression for $a^2+b^2+c^2$ from the given equations" commits the trajectory to a path that does not have a closed-form solution under the given constraints. RSP's perturbation produces, in 1 of 16 attempts, a trajectory whose opening frame is *constrained optimization* with a symmetry-based variable reduction. The two trajectories share the same model weights, the same problem constraints, and the same sampling temperature; the input-side RSP perturbation is the only intervention, and its visible effect is the different framing the model commits to in the first few sentences.

C.3. Case Study 2: AIME24 Problem 22 (Constrained List Construction)

Problem.

A list of positive integers has the following properties:

- The sum of the items in the list is 30.
- The unique mode of the list is 9.
- The median of the list is a positive integer that does not appear in the list itself.

Find the sum of the squares of all the items in the list.

Ground truth: $\boxed{236}$, corresponding to $\{5, 7, 9, 9\}$ with median $\frac{7+9}{2} = 8$.

Aggregate result. Baseline produces 0/16 correct; RSP produces 3/16 correct. Baseline's 16 samples produce 16 distinct final answers (no two agree); the correct value 236 does not appear. RSP's 16 samples concentrate slightly more, with 236 (3 times) and 248 (3 times) tied as modal answers and the remaining mass spread across 10 distinct values.

Structural-assumption split. The constraint "the median is a positive integer that does not appear in the list" *requires* an even-length list: for an odd-length list the median is by definition an element of the list. Across Baseline's 16 samples, 6 commit to an odd-length list (typically 5 elements) at the opening and only 3 consider an even-length list. Across RSP's 16 samples, 8 commit to odd and 6 consider even. The shift in the even-length consideration rate (3/16 → 6/16) is the structural correlate of RSP's accuracy gain.

Failure mode (Baseline). Baseline's representative trajectory (final answer 224) commits to an odd-length list *at the opening* and never reconsiders.

Baseline trajectory (final answer 224) — excerpts

Opening commitment: "Let's start by assuming the list has an odd number of elements. If the list has $2k + 1$ elements, the median is the $(k + 1)$-th element. Let's try with the smallest possible odd number…"
Construction: 5 elements, $a_1 + a_2 + 9 + a_4 + 9 = 30$ so $a_1 + a_2 + a_4 = 12$. Try $a_1 = 2$, $a_2 = 3$, $a_4 = 7$: list $\{2, 3, 7, 9, 9\}$, median 7, "which does not appear in the list, so this is a valid candidate."
Termination: Sum of squares $= 4 + 9 + 49 + 81 + 81 = \boxed{224}$.

Diagnosis. Two errors compound: (i) the trajectory *defaults to odd length* at the opening without justification, and the constraint "median not in the list" forces even length, which the trajectory never derives; (ii) the median value 7 is verified as a positive integer but *not* verified as absent from the list (7 is in fact an element). The same structural assumption recurs across most of Baseline's 16 samples; the elements vary, but the odd-length commitment and the incomplete constraint check are constant.

Success mode (RSP). The correct RSP trajectory (final answer 236) considers $n = 4$ from the outset and treats the median formula by parity.

RSP trajectory (final answer 236) — excerpts

Opening (Step 2: median by parity). "If the list has an odd number of elements, the median is the middle element. If the list has an even number of elements, the median is the average of the two middle elements. Since the median is a positive integer that does not appear in the list, it must be an integer and not equal to any of the other numbers in the list."
Step 3: starting at $n = 4$. "$n \geq 4$ (at least two 9's). Case 1: $n = 4$. Sum: $9 + 9 + x + y = 30$, so $x + y = 12$."
Construction: Try $x = 5$, $y = 7$: list $\{5, 7, 9, 9\}$, median $\frac{7+9}{2} = 8$ (integer, not in list). All conditions satisfied; sum of squares $= 25 + 49 + 81 + 81 = \boxed{236}$.

Diagnosis. Three properties distinguish this trajectory: **(i)** *even length is considered first* ($n = 4$ enumerated as the initial case); **(ii)** the median is computed as an arithmetic mean for even length, producing $8 \notin \{5, 7, 9, 9\}$; **(iii)** the constraint check is performed against the candidate *value* (8), not against the structural position. The other two correct RSP samples

reach 236 through partially incorrect intermediate reasoning, suggesting that some RSP samples that begin with the same odd-length assumption as Baseline can still arrive at the correct numerical answer.

Interpretation. Baseline's failure is rooted in the *first structural assumption* after the conditions are listed: the trajectory commits to an odd-length list within its first few hundred tokens and never reconsiders. RSP's perturbation increases the fraction of samples that consider $n = 4$ from 3/16 to 6/16, and one of these reaches the correct constraint check. The split between odd-length and even-length openings is the empirical fingerprint of the template-selection shift; on a small fraction of samples, the alternative template happens to be the structurally consistent one.

C.4. Summary of Case Studies

Both case studies exhibit the same shape. Baseline's 16 samples cluster around a single failure template (algebraic-identity framing for #6; odd-length-list commitment for #22) and never break out of it within the sample budget. RSP does not change the model's underlying knowledge: most RSP samples fall into the same templates as Baseline. What RSP changes is *which template the model commits to in the opening sentences*. On a small fraction of trajectories, the alternative template happens to admit a solvable subproblem (the optimization framing for #6, the even-length list for #22), and the trajectory reaches the correct answer. The accuracy lift (0/16 $\rightarrow$ 1/16 on #6, 0/16 $\rightarrow$ 3/16 on #22) is the outcome-level signature of this opening-sentence divergence, consistent with the input-side branch-opening mechanism of §3.2: the perturbation is largest while the KV cache is short and the trajectory has not yet committed to a structural template.

D. Prompt-Law Properties Used by the Theory

This appendix isolates the only properties of the RSP law used in the main text: centeredness, direction-agnostic covariance under a variance budget, and positive probability on every nonempty open set. We avoid stronger semantic claims because Section 3.2 needs only these geometric and measure-theoretic facts.

D.1. Centered Perturbations Do Not Impose Systematic First-Order Drift

Let z_a^{Base} denote the no-RSP output logit at vocab token a and $z_a^{\text{RSP}}(\bar{h})$ the corresponding logit when the centered RSP perturbation takes value $\bar{h}$. The (true) vocab-logit gap under RSP is $\Delta_{ab}(\bar{h}) := z_a^{\text{RSP}}(\bar{h}) - z_b^{\text{RSP}}(\bar{h})$, distinct from the no-RSP baseline gap $\Delta_{ab}^{\text{Base}} := z_a^{\text{Base}} - z_b^{\text{Base}}$. Note that $\bar{h} = 0$ corresponds to the centered RSP at its entrywise mean rather than to the no-RSP forward pass, so in general $\Delta_{ab}(0) \neq \Delta_{ab}^{\text{Base}}$. The transformer's logit map is non-linear in $\bar{h}$, but a first-order Taylor expansion around $\bar{h} = 0$ yields the local surrogate

$$\Delta_{ab}^{\text{lin}}(\bar{h}) \;:=\; \Delta_{ab} + b_{ab}^{\top}\bar{h},$$

with $\Delta_{ab} := \Delta_{ab}(0)$ and $b_{ab} := \nabla_{\bar{h}}(z_a^{\text{RSP}} - z_b^{\text{RSP}})|_{\bar{h}=0}$. The proposition below is stated for the surrogate Δ_{ab}^{lin}. For the true non-linear gap $\Delta_{ab}(\bar{h})$, the first-order term has zero mean under centered perturbations, but the Taylor remainder $r_{ab}(\bar{h}) = O(\|\bar{h}\|^2)$ may contribute a second-order mean shift; we make no claim about its sign or magnitude.

Proposition D.1 (No systematic first-order drift in the surrogate)**.** *If* $\mathbb{E}[\bar{h}] = 0$*, then*

$$\mathbb{E}\big[\Delta_{ab}^{\text{lin}}(\bar{h})\big] = \Delta_{ab}$$

for every token pair (a, b)*. If a deterministic centered prompt* $\bar{h}_0$ *is reused across runs, then*

$$\Delta_{ab}^{\text{lin}}(\bar{h}_0) - \Delta_{ab} = b_{ab}^{\top}\bar{h}_0$$

is a fixed directional bias.

Proof. The centered case is immediate from linearity of expectation:

$$\mathbb{E}\big[\Delta_{ab}^{\text{lin}}(\bar{h})\big] = \Delta_{ab} + b_{ab}^{\top}\mathbb{E}[\bar{h}] = \Delta_{ab}.$$

The deterministic case follows by substitution. □

This proposition rules out a run-averaged first-order bias. It does not say that individual prompt draws leave the logits unchanged.

D.2. Proof of Proposition 3.1

For a unit vector u, the first-order perturbation energy available along u is $\mathrm{Var}_{h\sim D}(u^\top h) = u^\top \Sigma_D u$. The worst-case directional energy is therefore the minimum Rayleigh quotient of Σ_D.

Proof. For any symmetric covariance matrix Σ_D,

$$\min_{\|u\|=1} u^\top \Sigma_D u = \lambda_{\min}(\Sigma_D).$$

Let the eigenvalues of Σ_D be $\lambda_1, \ldots, \lambda_d$. Then

$$\lambda_{\min}(\Sigma_D) \le \frac{1}{d}\sum_{m=1}^{d} \lambda_m = \frac{\mathrm{tr}(\Sigma_D)}{d} \le \frac{\rho^2}{d}.$$

Equality holds if and only if all eigenvalues are equal, i.e., $\Sigma_D = (\rho^2/d)\mathbf{I}_d$. □

The proof uses only covariance. We choose a Gaussian law for RSP because it adds the open-set reachability property proved next.

D.3. Open-Set Reachability of Isotropic Gaussian Prompts

Proposition D.2 (Open-set reachability)**.** *Let $\bar{h} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_d)$ with $\sigma > 0$. Then every nonempty open set $U \subseteq \mathbb{R}^d$ satisfies*

$$\Pr(\bar{h} \in U) > 0.$$

Proof. The Gaussian density

$$(2\pi\sigma^2)^{-d/2} \exp\left(-\frac{\|\bar{h}\|^2}{2\sigma^2}\right)$$

is strictly positive at every point of $\mathbb{R}^d$. Every nonempty open set contains a ball of positive Lebesgue measure, and integrating a strictly positive density over that ball gives positive probability. □

This is the only support property used in the informal branching argument of §3.2 and in Appendix E.3.3.

E. Proofs for the Transformer-Level Mechanism

This appendix follows the same order as the main theory section: exploration, annealing, and performance gain. The prompt-law facts used before these proofs are collected in Appendix D.

E.1. Exploration: attention decomposition and perturbation size

Proposition E.1 (Attention decomposition)**.** *For one attention head at query position i in layer ℓ, let $\alpha_{ij}^{(\ell)}$ be the attention weights over $n \ge 1$ unmasked real tokens and $L \ge 1$ random tokens, with finite attention logits. Define*

$$w_{r,i}^{(\ell)} := \sum_{j=1}^{L} \alpha_{i,n+j}^{(\ell)}$$

and, for $j \in [n]$,

$$\tilde{\alpha}_{ij}^{(\ell)} := \frac{\alpha_{ij}^{(\ell)}}{1 - w_{r,i}^{(\ell)}}.$$

Then

$$\mathbf{o}_i^{(\ell)} = (1 - w_{r,i}^{(\ell)})\, \tilde{\mathbf{o}}_i^{(\ell)} + \boldsymbol{\eta}_i^{(\ell)},$$

where

$$\tilde{\mathbf{o}}_i^{(\ell)} := \sum_{j=1}^{n} \tilde{\alpha}_{ij}^{(\ell)} \mathbf{v}_j^{(\ell)} \qquad \textit{and} \qquad \boldsymbol{\eta}_i^{(\ell)} := \sum_{j=1}^{L} \alpha_{i,n+j}^{(\ell)} \mathbf{v}_{r_j}^{(\ell)}.$$

Proof. The head output is

$$\mathbf{o}_i^{(\ell)} = \sum_{j=1}^{n} \alpha_{ij}^{(\ell)} \mathbf{v}_j^{(\ell)} + \sum_{j=1}^{L} \alpha_{i,n+j}^{(\ell)} \mathbf{v}_{r_j}^{(\ell)}.$$

By the softmax positivity and the existence of at least one unmasked real token, $\sum_{j=1}^{n} \alpha_{ij}^{(\ell)} = 1 - w_{r,i}^{(\ell)} > 0$, so

$$\sum_{j=1}^{n} \alpha_{ij}^{(\ell)} \mathbf{v}_j^{(\ell)} = (1 - w_{r,i}^{(\ell)}) \sum_{j=1}^{n} \frac{\alpha_{ij}^{(\ell)}}{1 - w_{r,i}^{(\ell)}} \mathbf{v}_j^{(\ell)} = (1 - w_{r,i}^{(\ell)})\, \tilde{\mathbf{o}}_i^{(\ell)}.$$

Substituting this identity gives the decomposition. □

Corollary E.2 (Magnitude scales with random-token attention)**.** *For every realization of the random values,*

$$\|\boldsymbol{\eta}_i^{(\ell)}\| \le w_{r,i}^{(\ell)} \max_{j \in [L]} \|\mathbf{v}_{r_j}^{(\ell)}\|.$$

Proof. By the triangle inequality,

$$\|\boldsymbol{\eta}_i^{(\ell)}\| = \left\| \sum_{j=1}^{L} \alpha_{i,n+j}^{(\ell)} \mathbf{v}_{r_j}^{(\ell)} \right\| \le \sum_{j=1}^{L} \alpha_{i,n+j}^{(\ell)} \|\mathbf{v}_{r_j}^{(\ell)}\| \le w_{r,i}^{(\ell)} \max_{j \in [L]} \|\mathbf{v}_{r_j}^{(\ell)}\|.$$

□

Corollary E.2 is the only quantitative fact needed in the main text. No independence assumption between attention weights and random keys is required. Once $w_{r,i}^{(\ell)}$ is small, the random contribution must be small as well.

E.2. Corollaries of Theorem 3.2 on KV cache growth

The fixed-gap decay bound yields the corollaries below.

Corollary E.3 (Sequence-wise annealing of the fixed-gap envelope)**.** *For fixed L and $\Delta_i^{(\ell)}$, the upper bound*

$$f(n) = \frac{L}{L + n \exp(\Delta_i^{(\ell)} / \sqrt{d_{head}})}$$

is strictly decreasing in n and tends to 0 as $n \to \infty$. KV cache growth alone therefore narrows the largest RSP attention mass compatible with that gap.

During autoregressive decoding $\Delta_i^{(\ell)}$ also moves because queries, keys, and hidden states co-evolve. The accurate reading is not "the realized mass decreases at every step" but "the envelope compatible with a fixed gap shrinks as the KV cache grows."

Proof. Differentiate $f(n)$ with respect to n:

$$f'(n) = -\frac{L \exp(\Delta_i^{(\ell)} / \sqrt{d_{\text{head}}})}{\left(L + n \exp(\Delta_i^{(\ell)} / \sqrt{d_{\text{head}}})\right)^2} < 0.$$

Thus $f(n)$ is strictly decreasing, and its denominator diverges linearly in n while its numerator is fixed, so $f(n) \to 0$. □

Corollary E.4 (Vanishing random contribution under annealing)**.** *If $n \to \infty$ while L and $\Delta_i^{(\ell)}$ remain fixed, $w_{r,i}^{(\ell)} \to 0$, and for every realization of the random values*

$$\|\boldsymbol{\eta}_i^{(\ell)}\| \le w_{r,i}^{(\ell)} \max_{j \in [L]} \|\mathbf{v}_{r_j}^{(\ell)}\| \to 0.$$

Proof. $w_{r,i}^{(\ell)} \to 0$ by Corollary E.3. Apply Corollary E.2 to bound $\|\boldsymbol{\eta}_i^{(\ell)}\|$. The maximum norm is finite for any realization of finitely many sampled values. □

E.3. Performance gain: from local admission to Pass@N

From here we work at the vocab-logit and task levels, dropping the per-layer index ℓ. The main text uses a first-order surrogate to state two claims: the §3.2 claim that a baseline-excluded vocab token can enter the local top-K set, and the §3.3 closing that independent reruns turn such local openings into Pass@N gains. The proofs below use only one support condition on the centered prompt law D:

$$\text{(S1)} \qquad D(U) > 0 \quad \text{for every nonempty open set } U \subseteq \mathbb{R}^d.$$

Proposition D.2 shows that the isotropic Gaussian law of §3.2 satisfies (S1).

E.3.1. Autoregressive Formulation

In autoregressive decoding, the joint distribution over a sequence $y_{1:T}$ factorizes as

$$\Pr(y_{1:T} \mid x) = \prod_{t=1}^{T} p(y_t \mid x, y_{<t}),$$

where each step is governed by a prefix-conditioned distribution. Any perturbation introduced by RSP affects generation through a sequence of local changes to $p(y_t \mid x, y_{<t})$, rather than a single global distribution.

E.3.2. Local Linearization of Logits under RSP

Throughout this subsection, $\bar{h} \in \mathbb{R}^d$ denotes *one* centered prompt vector from the RSP definition (Eq. (1) in §3.1), treated as a per-token surrogate. The actual implementation uses a length-L prompt $\mathbf{H} = (h_1, \ldots, h_L)$ with L independent draws; the arguments below isolate how one local perturbation can shift the logits. We model the perturbed logits as a function $z_a(\bar{h})$ at vocab token a, assuming local smoothness in a neighborhood of $\bar{h} = 0$:

$$z_a(\bar{h}) = z_a(0) + \nabla_{\bar{h}} z_a(0)^\top \bar{h} + r_a(\bar{h}),$$

where $r_a(\bar{h}) = O(\|\bar{h}\|^2)$. Defining $c_a := \nabla_{\bar{h}} z_a(0)$ and $z_a := z_a(0)$, we obtain the local affine surrogate

$$z_a^{\text{lin}}(\bar{h}) := z_a + c_a^\top \bar{h}.$$

This approximation is used as a first-order surrogate for branch opening, not as an exact global model of the transformer logits.

E.3.3. Top-K entry region (Remark)

This complements the informal branching argument in §3.2. Define the entry region

$$\mathcal{G}_{a,K} \;:=\; \big\{\bar{h} \in \mathbb{R}^d : \#\{b \neq a : z_b^{\text{lin}}(\bar{h}) > z_a^{\text{lin}}(\bar{h})\} \leq K - 1\big\}.$$

If $\mathcal{G}_{a,K}$ contains a nonempty open set U, then by (S1) $D(U) > 0$, and $U \subseteq \mathcal{G}_{a,K}$ gives $\Pr_{\bar{h}}(\bar{h} \in \mathcal{G}_{a,K}) \geq D(U) > 0$. A useful sufficient condition is the existence of some $\bar{h}_0$ at which token a strictly outranks all but at most $K - 1$ other tokens in the affine surrogate; by continuity, a neighborhood of $\bar{h}_0$ is then contained in $\mathcal{G}_{a,K}$. This is a rank-changing event unreachable by the monotone temperature transform, but it does not by itself imply (M1).

E.3.4. Pass@N ensemble bound (Remark)

The bound $\Pr(\text{Pass@}N \text{ on } x) \geq 1 - (1 - p_{\min}(x))^N$ in the main text is a standard independent-Bernoulli union argument. Let A_n be the event that the n-th run reaches a correct trajectory on x; under (M1) $\Pr(A_n) \geq p_{\min}(x)$, and under (M2)

$$\Pr\Big(\bigcap_{n=1}^{N} A_n^c\Big) \leq (1 - p_{\min}(x))^N, \qquad \Pr\Big(\bigcup_{n=1}^{N} A_n\Big) \geq 1 - (1 - p_{\min}(x))^N \;\geq\; 1 - e^{-N p_{\min}(x)}.$$

This is a generic fact, not specific to RSP; RSP's role is to make branch opening possible and to support (M2) through independent resampling, while whether a branch satisfies the task-side correctness condition (M1) remains task-dependent.

F. Attention Mass Analysis

This appendix formalizes the per-token attention mass reported in Figure 2 and the exclusion criteria applied to the reasoning span and the layer axis. For each of the 500 MATH-500 problems, a fresh RSP $\mathbf{H} \in \mathbb{R}^{L \times d}$ with $L = 10$ is sampled independently, so the reported heatmaps average over both problem variability and RSP-vector variability.

F.1. Per-Token Attention Mass

For each problem, we measure attention on the concatenated sequence [prompt; $\mathbf{H}$; generation], where the generation is the trajectory produced by the same model under matching RSP injection. Let $\alpha_{i,j}^{(\ell,m)}$ denote the post-softmax attention weight at layer ℓ, head index m (using m to avoid collision with the vocab token a of §3.2 and the RSP matrix $\mathbf{H}$), from query position i to key position j, satisfying $\sum_j \alpha_{i,j}^{(\ell,m)} = 1$ under the causal mask. Let Q and R denote the index sets of question and RSP tokens with sizes $|Q|$ and $|R| = L$, and let N_{head} be the number of heads. The per-token attention mass that query i directs to each group at layer ℓ is

$$\mathrm{PTM}_Q[\ell, i] = \frac{1}{|Q|\, N_{\text{head}}} \sum_{m=1}^{N_{\text{head}}} \sum_{j \in Q} \alpha_{i,j}^{(\ell,m)}, \qquad \mathrm{PTM}_R[\ell, i] = \frac{1}{|R|\, N_{\text{head}}} \sum_{m=1}^{N_{\text{head}}} \sum_{j \in R} \alpha_{i,j}^{(\ell,m)}. \tag{6}$$

Normalization by $|Q|$ and $|R|$ controls for group size, so both quantities express how much attention a single question (resp. RSP) token receives on average under the same softmax denominator. Question tokens thereby serve as a size-matched baseline that absorbs sequence-length effects common to both groups.

F.2. Heatmap Aggregation

We partition the layer indices and the reasoning position indices each into five equal-size bins $\{B_L^{(b)}\}_{b=1}^5$ and $\{B_P^{(b)}\}_{b=1}^5$. For sample s and target group $\bullet \in \{Q, R\}$, the value in cell (b_L, b_P) is

$$M_\bullet^{(s)}[b_L, b_P] = \frac{1}{|B_L^{(b_L)}| \cdot |B_P^{(b_P)}|} \sum_{\ell \in B_L^{(b_L)}} \sum_{i \in B_P^{(b_P)}} \mathrm{PTM}_\bullet[\ell, i]. \tag{7}$$

Figure 2 reports the sample average of Eq. (7) over the 500 valid samples.

F.3. Excluding the `\boxed{...}` Span

The reasoning position range terminates strictly before the final `\boxed{...}` span. Chain-of-thought outputs decompose into a *text* (content) component and a *pattern* (template) component that contribute independently to downstream performance (Madaan & Yazdanbakhsh, 2022). The `\boxed{...}` wrapper is a canonical pattern: a short, stereotyped span whose role is to complete the answer format rather than to extend reasoning. Including it would therefore inject a template-driven attention signature unrelated to reasoning dynamics.

F.4. Excluding the Final Two Layers

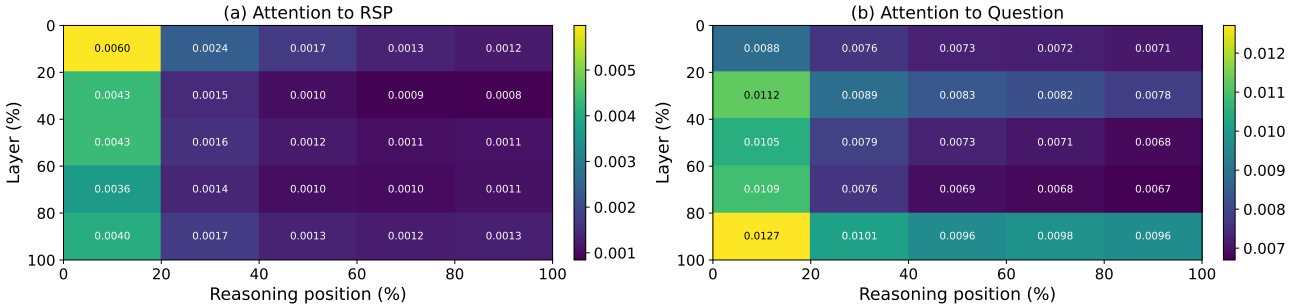
The layer axis further excludes the last two transformer layers. This choice is motivated by a standard late-layer effect rather than by any model-specific tuning decision.

Output-projection specialization. Late-layer hidden states lie in an approximately linear relationship with vocabulary logits and therefore function as a progressive linear readout rather than as representation-transforming blocks (Belrose et al., 2023). Consistently, late-layer feed-forward modules have been characterized as output-distribution shaping mechanisms (Geva et al., 2021). Attention in these layers therefore reflects output formation rather than reasoning computation.

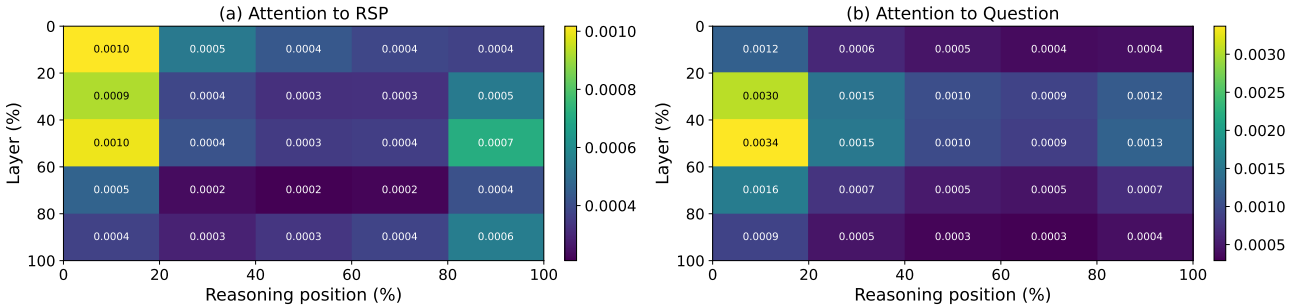
Excluding the final two layers keeps the heatmap focused on reasoning-phase dynamics instead of readout-phase dynamics. This exclusion is not load-bearing for the main claim: the sequence-direction attenuation emphasized in the main text is also visible without it, and Theorem 3.2 concerns sequence-direction attenuation rather than layer-wise monotonicity.

F.5. Additional Suffix Heatmaps

For the late-layer reasons discussed above, the pattern in Figure 7 does not generalize uniformly across every cell; nevertheless, the overall trend is consistent across both models: question-token attention varies broadly across layers, whereas RSP attention shows sequence-wise decay consistent with Theorem 3.2 alongside an empirical layer-wise attenuation that the theorem itself does not predict.



(a) Qwen2.5-Math-1.5B-Instruct, *suffix*



(b) LLaMA-3.1-8B-Instruct, *suffix*

Figure 7. Per-token RSP attention mass under *suffix* injection for the remaining two models (500 MATH-500 samples each). Axes and preprocessing match Figure 2: x-axis is reasoning position binned into five quantiles, y-axis is layer depth binned into five quantiles (top = shallow), color encodes the 500-sample mean per-token attention assigned to RSP tokens, and tokens inside `\boxed{}` spans and the last two layers are excluded.

G. DAPO Implementation

This appendix complements Section 4.5 with the DAPO loss modifications relative to GRPO, the DAPO + RSP implementation, and full per-step results. The training code is included in the anonymized supplementary material; pretrained DAPO and DAPO + RSP checkpoints across training steps 10–100 will be released at the camera-ready stage.

G.1. From GRPO to DAPO

The vanilla GRPO objective (Shao et al., 2024) for a group of G rollouts $\{y_i\}_{i=1}^{G}$ (each $y_i = (y_{i,1}, \ldots, y_{i,|y_i|})$ a token sequence; we use y rather than o throughout this appendix to avoid collision with the head output $\mathbf{o}_i^{(\ell)}$ of §3.2) with rewards $\{R_i\}_{i=1}^{G}$ is

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\Big(\min\big(r_{i,t}\hat{A}_{i,t},\ \text{clip}(r_{i,t}, 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\big) - \beta\, D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\Big)\right], \tag{8}$$

with importance ratio $r_{i,t} = \pi_\theta(y_{i,t} \mid q, y_{i,<t})/\pi_{\theta_{\text{old}}}(y_{i,t} \mid q, y_{i,<t})$, group-relative advantage $\hat{A}_{i,t} = (R_i - \text{mean}(\{R_i\}))/\text{std}(\{R_i\})$, clipping range ε, and KL penalty $\beta\, D_{\text{KL}}$ against the reference π_{ref}.

We build on the SimpleRL-Zoo (Zeng et al., 2025) training pipeline and implement DAPO on top of VeRL (Sheng et al., 2025). DAPO modifies Eq. (8) in four ways: (i) token-level loss aggregation $1/\sum_i |y_i|$ instead of $1/|y_i|$, (ii) asymmetric clipping

with $(\varepsilon_{\text{low}}, \varepsilon_{\text{high}}) = (0.2, 0.28)$, (iii) an implementation-side negative-advantage dual-clipping safeguard $\max(L_{\text{clipped}}, c\hat{A})$ with $c = 10$ (active only when $\hat{A} < 0$; omitted from Eq. (9) below for readability), and (iv) KL terms removed. The resulting DAPO objective is

$$\mathcal{J}_{\text{DAPO}}(\theta) \;=\; \mathbb{E}\left[\frac{1}{\sum_i |y_i|}\sum_{i,t}\min\Big(r_{i,t}\hat{A}_{i,t},\; \text{clip}\big(r_{i,t},\; 1{-}\varepsilon_{\text{low}},\; 1{+}\varepsilon_{\text{high}}\big)\hat{A}_{i,t}\Big)\right]. \tag{9}$$

G.2. DAPO + RSP Implementation

For DAPO + RSP, we additionally inject a per-rollout RSP $\mathbf{H}_i \in \mathbb{R}^{L\times d}$ ($L = 20$) into each rollout. Each $\mathbf{H}_i$ is generated deterministically from a per-step rollout seed and injected via a forward hook; it is not a trainable parameter, so the learning signal is the policy adapting to arbitrary $\mathbf{H}_i$ rather than a learned prompt. Including $\mathbf{H}_i$ in the log-probabilities at both rollout and update time keeps the update on-policy with respect to the RSP-conditioned distribution (avoiding off-policy mismatch). With $G = 8$ rollouts per prompt, each rollout receives an independently drawn $\mathbf{H}_i$.

G.3. Evaluation Protocol

Each saved checkpoint is converted to HuggingFace format and evaluated using the SimpleRL-Zoo `eval_math_nodes.sh` pipeline, which selects sampling parameters per benchmark to reduce variance on the smaller competition sets:

Table 11. Per-benchmark evaluation protocol used by SimpleRL-Zoo `sh/eval.sh`.

Benchmark	# Problems	Temperature	n_{sampling}	Metric
AIME 2024	30	1.0	32	Avg@32
AMC 2023	40	1.0	32	Avg@32
MATH-500	500	0.0	1	accuracy (mean@1)
GSM8K	1,319	0.0	1	accuracy
OlympiadBench	675	0.0	1	accuracy
Minerva Math	272	0.0	1	accuracy
GaoKao 2023-EN	385	0.0	1	accuracy

The maximum number of generated tokens is 16,000 across all benchmarks. The reported average is the unweighted mean over the five benchmarks (GSM8K, MATH-500, College Math, Minerva Math, AIME 2024).

G.4. Data, Batch, and Hyperparameters

Table 12. DAPO / DAPO + RSP training setup on Qwen2.5-Math-7B.

Field	Value
Train data	MATH Level 3–5 subset (`simplelr_math_35`, ~8,500 prompts)
Prompt template	`qwen-boxed`
Train batch size	1024
PPO mini-batch size	256 (4 PPO updates per rollout)
Rollouts per prompt G	8
Max prompt / response length	2,028 / 2,048
Rollout sampling	temperature 1.0, top-p 1.0, top-k 50
$(\varepsilon_{\text{low}}, \varepsilon_{\text{high}})$	$(0.2,\ 0.28)$
Dual clip c	10.0
Loss aggregation	token-mean ($1/\sum_i |y_i|$)
KL terms	disabled
Entropy coefficient	0
Optimizer	AdamW, learning rate 5×10^{-7} (constant), no warmup
Total training	~10 epochs, ≥ 80 optimization steps
Save / eval frequency	every 10 steps
RSP (DAPO + RSP only)	*suffix* injection, $L = 20$, freshly drawn per rollout
Hardware	4× NVIDIA B200 GPUs
Wall-clock training time	∼12 hours per run

G.5. Per-Step Average Accuracy

Table 13. Five-benchmark average accuracy (%) at every checkpoint, computed over GSM8K, MATH-500, College Math, Minerva Math, and AIME 2024. **Bold** marks each method's peak.

Step	DAPO	DAPO + RSP
0	30.06	30.06
10	46.40	46.22
20	49.02	49.54
30	49.58	50.72
40	50.62	52.06
50	50.88	52.40
60	51.84	52.96
70	52.52	53.86
80	**53.20**	54.04
90	52.52	**54.36**
100	50.54	53.64

H. Broader Impacts

This work provides empirical and theoretical analysis of how random embedding injection affects the reasoning behavior of large language models. The contributions are primarily foundational, with potential downstream implications for reinforcement learning-based post-training of reasoning models such as GRPO.

Positive impacts. RSP-induced trajectory diversity could improve the sample efficiency of RL-based LLM post-training by providing richer learning signals within group-relative advantage estimation. This has the potential to reduce compute costs associated with alignment and reasoning-oriented fine-tuning, making such methods more accessible to research groups with limited resources.

Potential concerns. Methods that expand the effective output support of LLMs may increase the reachability of low-probability tokens, which includes both desirable alternative reasoning paths and potentially unsafe or off-distribution outputs. Practitioners applying RSP in production systems should combine it with existing safety filtering and alignment mechanisms rather than treating it as a standalone diversity source.

Limitations on impact scope. Since this work focuses on analysis rather than deploying a new model or dataset, the direct societal impact is limited. The broader implications described above are contingent on future work integrating RSP into applied training pipelines.