BEAM SEARCH AS TEST-TIME SELF-DISTILLATION VIA COUNTERFACTUAL CONTEXTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-Distillation Fine-Tuning (SDFT) enables a language model to act as its own teacher: by conditioning on a demonstration, the model produces an implicit reward via pointwise mutual information, which guides on-policy learning without external supervision. However, SDFT operates at *training time*: it requires gradient updates and access to expert demonstrations, making it inapplicable at inference. We propose *test-time self-distillation*, a decoding-time method that extracts a steering signal from the self-distillation framework without any parameter updates, reward models, or training data. Our key insight is that *counterfactual contexts*, i.e. fixed textual templates that hypothetically prime the model for excellent versus poor reasoning, can substitute for the demonstration. The log-odds ratio of a candidate answer under these two counterfactual conditions defines a new reward signal. We derive the optimal KL-regularized policy under this reward, which takes the form of a Gibbs reweighting of the base distribution. Crucially, this reweighting is global: it cannot be decomposed into independent per-token operations without ignoring future trajectory quality. We therefore approximate the target distribution via beam search. Experiments on mathematical reasoning (MATH500), code generation (HumanEval), and graduate-level science QA (GPQA) across multiple model scales show that test-time self-distillation improves over standard sampling, low temperature, beam search and power sampling baselines on average, demonstrating that the self-distillation principle can be operationalized at inference time.

1 INTRODUCTION

During training, language models can serve as their own teachers through Self-Distillation Fine-Tuning (SDFT) Shenfeld et al. (2026); Hübotter et al. (2026). The model can be conditioned on an expert demonstration c and acts as a teacher for the same model without the demonstration. The key object is the *pointwise mutual information* (PMI) between the demonstration and the model's output, which serves as an implicit reward: $r(a, q, c) = \log \pi(a \mid q, c) - \log \pi(a \mid q)$. By optimizing this reward under a KL constraint via on-policy distillation, SDFT eliminates the need for external task-specific rewards. However, SDFT is fundamentally a *training-time* method. It requires (i) an expert demonstration c for each task, (ii) gradient updates to the model parameters, and (iii) an on-policy training loop that repeatedly samples from the student and computes loss against the teacher. At *test time*, when a user poses a question and the model must generate an answer, none of these components are available. The model is frozen, there is no demonstration, and there is no training loop. This raises a natural question: *can the self-distillation principle be operationalized at test time, without any parameter updates?*

We propose a solution to this question by replacing the expert demonstration with *counterfactual contexts*. Rather than conditioning on a real demonstration c, which is not available at test time, we condition the model on two fixed textual templates: a positive context c^+ that hypothetically primes the model for excellent reasoning (e.g., *"This is an example for a response to the question with excellent reasoning:"*) and a negative context c^- that primes it for poor reasoning (e.g., *"This is an example for a response to the question with wrong reasoning:"*). These contexts are never shown to the user and do not appear in the generated output. They are *counterfactual probes* that ask: *how would the model's probability of this answer change if it were reasoning excellently versus poorly?*

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

The contrastive PMI between these counterfactual contexts defines an implicit reward:

$$R(q, a) = \log \pi_\theta(a \mid q, c^+) - \log \pi_\theta(a \mid q, c^-). \tag{1}$$

Intuitively, an answer that is more likely under the excellent-reasoning counterfactual than under the poor-reasoning counterfactual receives a positive reward, steering generation toward responses the model associates with high-quality reasoning.

This contrastive formulation is necessitated by the absence of ground-truth supervision at test time. In SDFT, an expert demonstration c is available, the reward $r(y, x, c) = \log \pi(y \mid x, c) - \log \pi(y \mid x)$ measures absolute alignment with c. Lacking any such oracle, we resort to a preference-based approach: rather than measuring absolute quality, we elicit the model's relative assessment under two counterfactual conditions. Connecting this to the framework of Rafailov et al. (2024), the reward $R(q, a)$ can be interpreted as a Bradley-Terry preference signal: the probability that the answer a is "better" under c^+ than under c^- is $\sigma(R(q, a)/\beta)$, where β is a temperature parameter.

Maximizing this reward subject to a KL-divergence constraint yields a closed-form target distribution:

$$\pi_{\text{target}}(a \mid q) \propto \pi_\theta(a \mid q) \exp\left(\frac{1}{\alpha} R(q, a)\right) = \pi_\theta(a \mid q) \left(\frac{\pi_\theta(a \mid q, c^+)}{\pi_\theta(a \mid q, c^-)}\right)^{1/\alpha}, \tag{2}$$

where α controls the steering strength. This is a Gibbs reweighting of the base policy, tilted toward answers favored under the excellent-reasoning counterfactual.

Sampling from π_{target} exactly is intractable due to the global normalization constant $Z(q)$. However, as Ji et al. (2026); Nguyen et al. (2026) demonstrate in the context of power distribution sampling Karan & Du (2025), local token-level approximations ignore the future: the per-token factorization $\prod_t \pi(a_t \mid q, a_{<t}) \exp(r_t/\alpha)$ drops the global normalization $Z(q)$, which depends on the entire trajectory. This is analogous to the gap between *low-temperature sampling* (a local operation that sharpens each token independently) and the *power distribution* p^α (a global reweighting that accounts for future trajectory quality). In fact, the power distribution is a special case where the reward is self-reinforcement, $R(a, q) = (\alpha - 1) \log \pi_\theta(a \mid q)$; our contrastive PMI reward $R(a, q) = \log \pi_\theta(a \mid q, c^+) - \log \pi_\theta(a \mid q, c^-)$ extends it to counterfactual contexts.

Because the target distribution is a global reweighting, we approximate it via *beam search* as done by Ji et al. (2026). This approach evaluates the reward on partial trajectories. It requires two additional forward passes per beam compared to traditional beam search, however it does not introduce new parameters, training data, or external reward models.

We evaluate test-time self-distillation on mathematical reasoning (MATH500), code generation (HumanEval), and graduate-level science QA (GPQA) using Qwen3.5-0.8B, Qwen2.5-7B, Qwen3.5-2B and Deepseek-Math-7B-Instruct. Across these settings, the method improves accuracy over all baselines (vanilla sampling, low-temperature sampling, beam search and power sampling). These results suggest that the self-distillation principle can be recovered at inference time.

Contributions. We make the following contributions:

- We introduce *test-time self-distillation*, a framework that extends the SDFT self-distillation principle to inference time without parameter updates, external reward models, or training data.
- We show that sampling from this target distribution cannot be decomposed into independent per-token operations: exact per-token sampling requires a *future correction* term (Proposition 1) that is as intractable as the global normalization constant.
- We propose *contrastive beam search* and analyze that the truncation error of this approximation is bounded and vanishes as generation progresses (Theorem 1). Across four models and three benchmarks, it outperforms standard sampling, low-temperature decoding, beam search, and power sampling, with ablations confirming that the semantic contrast drives the gains.

2 Background

2.1 KL-Regularized Reward Maximization

A central framework in language model alignment is the maximization of an expected reward subject to a KL-divergence constraint (Korbak et al., 2022; Rafailov et al., 2024). Given a reward function

$r(a, q)$ that evaluates the action a for a question q and a reference policy $\pi_k(\cdot \mid q)$, the optimization problem is:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{a\sim\pi(\cdot|q)}[r(a, q)] - \beta\, D_{\mathrm{KL}}(\pi(\cdot \mid q) \,\|\, \pi_k(\cdot \mid q)), \tag{3}$$

where $\beta > 0$ controls the strength of the KL penalty. This objective has a well-known closed-form solution:

$$\pi^*(a \mid q) = \frac{1}{Z(q)}\, \pi_k(a \mid q)\, \exp\left(\frac{1}{\beta}\, r(a, q)\right), \tag{4}$$

where $Z(q) = \sum_a \pi_k(a \mid q) \exp(r(a, q)/\beta)$ is a normalization constant. This *tilted* or *Gibbs* distribution reweights the reference policy by the exponential of the reward, concentrating probability mass on high-reward outputs while remaining anchored to the base distribution.

2.2 Self-Distillation Fine-Tuning

Self-Distillation Fine-Tuning (SDFT) (Shenfeld et al., 2026) extends the on-policy distillation framework by removing external reward. Given a base model with policy π_θ and an expert demonstration c for a task with prompt q, SDFT constructs a *teacher* by conditioning the same model on the demonstration: $\pi(\cdot \mid q, c)$. The *student* is the base model $\pi_\theta(\cdot \mid q)$. Training aims at minimizing the reverse KL divergence between student and teacher: $\mathcal{L}(\theta) = \mathbb{E}_{a\sim\pi_\theta(\cdot|q)}\left[\log \frac{\pi_\theta(a|q)}{\pi(a|q,c)}\right]$.

The key insight of SDFT is that this distillation objective is equivalent to policy gradient under an implicit reward derived from the *pointwise mutual information* between the demonstration and the model's output. Substituting the teacher $\pi(\cdot \mid q, c)$ as the optimal policy π^* in the KL-regularized framework Eq. 3 yields the reward: $r(a, q, c) = \log \pi(a \mid q, c) - \log \pi(a \mid q)$. This PMI reward measures how much the demonstration c "explains" the output a: outputs whose probability increases under the demonstration receive positive reward.

SDFT requires three components that are unavailable at test time: (i) an expert demonstration c, (ii) gradient updates to θ, and (iii) an on-policy training loop. Our work replaces the demonstration with counterfactual contexts and replaces gradient-based optimization with decoding-time approximation, preserving the PMI reward structure while eliminating all training-time requirements.

3 Method

We now formalize test-time self-distillation. The method has three components: (1) counterfactual contexts that define a contrastive reward (Section 3.1), (2) a theoretical analysis showing that naive per-token decoding fails due to a future correction term (Section 3.2), and (3) a contrastive beam search algorithm that approximates the target distribution (Section 3.3).

3.1 Counterfactual Contexts and Per-Token Reward

We replace the expert demonstration c from SDFT with two question-independent fixed textual templates c^+ and c^-. These are appended to the question q when computing the model's log-probabilities. We propose to use the following contrastive PMI reward $R(q, a) = \log \pi_\theta(a \mid q, c^+) - \log \pi_\theta(a \mid q, c^-)$ (Eq. 1), it decomposes via the autoregressive factorization into per-token rewards:

$$R(a, q) = \sum_{t=1}^{T} r_t(a_t, q), \quad r_t(a_t, q) = \log \pi_\theta(a_t \mid q, a_{<t}, c^+) - \log \pi_\theta(a_t \mid q, a_{<t}, c^-), \tag{5}$$

where $r_t(a_t, q)$ measures the change in log-odds for token a_t under the two counterfactual conditions given prefix $a_{<t}$. The KL-regularized optimal policy under this reward is the Gibbs reweighting $\pi_{\text{target}}(a \mid q) \propto \pi_\theta(a \mid q) \exp(R(a, q)/\alpha)$ (Eq. 2). The central challenge is sampling from this distribution.

3.2 The Future Correction

A naive approach to sampling from π_{target} would greedily follow the locally reweighted distribution

$$\tilde{\pi}(a_t \mid q, a_{<t}) \propto \pi_\theta(a_t \mid q, a_{<t}) \exp\left(\frac{1}{\alpha} r_t(a_t, q)\right). \tag{6}$$

This applies the contrastive reweighting independently at each token, analogous to how low-temperature sampling applies the power transformation locally (Ji et al., 2026). The following proposition shows that exact per-token sampling from π_{target} instead requires a *future correction* ζ_t accounting for downstream reward.

Proposition 1 (Per-Token Decomposition with Future Correction)**.** *Let π_θ be an autoregressive language model over vocabulary $\mathcal{V}$, and let the target distribution be*

$$\pi_{\text{target}}(a \mid q) = \frac{1}{Z(q)} \prod_{t=1}^{T} \pi_\theta(a_t \mid q, a_{<t}) \exp\left(\frac{1}{\alpha} r_t(a_t, q)\right),$$

where $r_t(a_t, q) = \log \pi_\theta(a_t \mid q, a_{<t}, c^+) - \log \pi_\theta(a_t \mid q, a_{<t}, c^-)$. Then for any partial sequence $a_{<t}$, the per-token conditional of the target distribution is

$$\pi_{\text{target}}(a_t \mid q, a_{<t}) = \frac{\pi_\theta(a_t \mid q, a_{<t}) \exp\left(\frac{1}{\alpha} r_t(a_t, q)\right) \zeta_t(a_t, q)}{\sum_{a' \in \mathcal{V}} \pi_\theta(a' \mid q, a_{<t}) \exp\left(\frac{1}{\alpha} r_t(a', q)\right) \zeta_t(a', q)}, \tag{7}$$

where the future correction *is*

$$\zeta_t(a_t, q) = \sum_{a_{t+1:T}} \prod_{s=t+1}^{T} \pi_\theta(a_s \mid q, a_{<t}, a_t, a_{<s}) \exp\left(\frac{1}{\alpha} r_s(a_s, q)\right). \tag{8}$$

The proof is available in Appendix A. The naive per-token distribution (Eq. 6) corresponds to setting $\zeta_t \equiv 1$, which is exact only at the final token ($t = T$). At all earlier positions, it ignores how each token choice affects the reward of future completions, i.e. the same local–global gap that separates low-temperature sampling from the power distribution. Computing ζ_t exactly requires marginalizing over all future trajectories, which is as intractable as computing $Z(q)$.

3.3 Contrastive Beam Search (CBS)

Since the target distribution is a global reweighting, we approximate it via block-wise beam search with contrastive scoring. The search operates over chunks of K tokens rather than individual tokens, alternating between stochastic extension and contrastive pruning. At each iteration, N active beams are extended by one chunk, scored under three prompt contexts, and pruned to the top N/W beams, which are then replicated to restore the population. Completed trajectories are moved to a terminal pool. See Algorithm 1 for the full details.

The score $s(a, q)$ of a beam composed of the tokens a approximates the log of the unnormalized target distribution:

$$\frac{1}{|a|} \sum_{t=1}^{|a|} \left[\log \pi_\theta(a_t \mid q, a_{<t}) + \frac{1}{\alpha} \left(\log \pi_\theta(a_t \mid q, c^+, a_{<t}) - \log \pi_\theta(a_t \mid q, c^-, a_{<t}) \right) \right]. \tag{9}$$

The contrastive difference inside the inner parentheses corresponds to the per-token reward r_t (Eq. 5). We use the mean rather than the sum to avoid length bias across beams of different lengths. Each beam requires two forward passes per chunk (base log probabilities can be reused from the generation process), so the total cost is slightly larger than standard beam search. However, this additional cost remains manageable because generation is usually more expensive than batched prefill operations in GPUs.

The beam score accumulates the contrastive reward over all tokens generated so far, which is equivalent to setting the future correction $\hat{\zeta}_t \equiv 1$ at each scoring point. Unlike naive per-token sampling (Eq. 6), which truncates at every token position with horizon $T - t$, beam search generates K-token chunks and scores them exactly, so the truncation only affects the future *beyond* each chunk (horizon $T - t - K$). The following theorem bounds the resulting approximation error.

Theorem 1 (Truncation Bias of Contrastive Beam Search)**.** *Assume per-token rewards are bounded: $|r_t(a_t, q)| \le r_{\max}$ for all t, a_t, and let K be the chunk size. At a scoring point at position t (start of a chunk), the K tokens within the chunk are scored exactly; the future correction ζ_t is truncated for*

Algorithm 1 Contrastive Beam Search

Require: Question q, model π_θ, contexts c^+, c^-
Require: Population N, pruning factor W, block size K, iterations T, temperature τ, steering strength α
Ensure: Selected response $\hat{a}$

Initialise $\mathcal{B} \leftarrow \{\}^N$; $\mathcal{A} \leftarrow \emptyset$; $t = 0$
while $t < T$ **do**
  Set a temporary buffer: $\mathcal{C} \leftarrow \emptyset$
  for all $a \in \mathcal{B}$ **do**
    Sample $\delta \sim \pi_\theta(\cdot \mid q, a; \tau)$ with temperature τ at most K tokens
    Concatenate: $a' \leftarrow [a, \delta]$ and compute $s(a', q)$ using Equation 9
    if a' terminates or $t = T$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{(a', s(a', q))\}$
    else
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{(a', s(a', q))\}$
    end if
  end for
  Remove duplicate trajectories from $\mathcal{C}$
  if $\mathcal{C} = \emptyset$ **then**
    break
  end if
  $\mathcal{S} \leftarrow$ top $\min(N/W, |\mathcal{C}|)$ elements by highest score from $\mathcal{C}$
  $\mathcal{B} \leftarrow$ replicate $\mathcal{S}$ to N beams
  Increment the counter $t = t + 1$
end while

return $\hat{a} \leftarrow \arg\max_{(a,s)\in\mathcal{A}} s(a, q)$

the remaining $T - t - K$ tokens by setting $\hat{\zeta}_t \equiv 1$. The per-step TV distance between the truncated distribution $\tilde{\pi}_{\text{trunc}}$ and the target π_{target} satisfies

$$\|\tilde{\pi}_{\text{trunc}}(\cdot \mid q, a_{<t}) - \pi_{\text{target}}(\cdot \mid q, a_{<t})\|_{\text{TV}} \leq \tanh\left(\frac{(T-t-K)\, r_{\max}}{\alpha}\right), \tag{10}$$

and the trajectory-level error compounds over T/K scoring decisions to at most $\frac{r_{\max}}{\alpha} \cdot \frac{T(T-K)}{2K}$. Naive per-token sampling (Eq. 6) truncates at every token with no chunk lookahead, giving the looser bound $\frac{r_{\max}}{\alpha} \cdot \frac{T(T-1)}{2}$; the chunk size K yields an approximately K-fold reduction. Under the balanced-contexts condition $\mathbb{E}_{a_t\sim\pi_\theta}[r_t(a_t, q)] = 0$ (equivalently, the KL from the base model to c^+ equals the KL to c^-), the per-step bound improves to $\tanh\big(\frac{(T-t-K)\, r_{\max}^2}{4\alpha^2}\big)$ and the trajectory bound to $\frac{r_{\max}^2}{4\alpha^2} \cdot \frac{T(T-K)}{2K}$. In both cases, the per-step error vanishes as $t \to T - K$.

The proof is in Appendix A. Theorem 1 shows that beam search's truncation error is bounded and, crucially, decreases as generation progresses: early chunks (small t) incur the largest error because they ignore the most future reward, while later chunks are scored nearly exactly. The chunk size K provides two advantages over naive per-token truncation: (i) the effective horizon at each scoring point is K tokens shorter ($T - t - K$ vs. $T - t$), and (ii) scoring decisions are made T/K times rather than T times. Together these yield an approximately K-fold reduction in trajectory error, formalizing the advantage of chunk-level scoring. To further reduce bias, it is also possible to use lookahead Monte-Carlo rollouts at the cost of greater inference time (Ji et al., 2026).

4 Experiments

We evaluate our Contrastive Beam Search (CBS) algorithm on four models and three benchmarks to examine if counterfactual contrastive scoring provides a reliable test-time steering signal across different models and reasoning tasks. CBS consistently improves over standard sampling and search

baselines and achieves the best performance in most settings. We further analyze its inference cost and the beam-search design choices that contribute to these gains.

4.1 Performance Evaluation

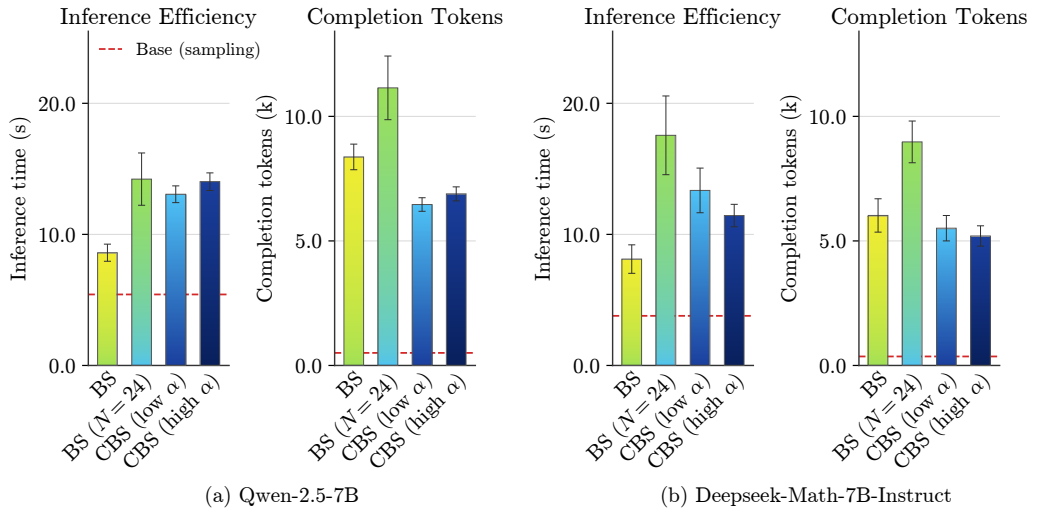
Models. We test four base models across different model scales and families to assess the robustness and generality of the CBS algorithm: Qwen3.5-0.8B, Qwen3.5-2B (Qwen Team, 2026), Qwen2.5-7B (Qwen Team, 2024), and DeepSeek-Math-7B-Instruct (Shao et al., 2024). All models are evaluated in their original pretrained or fine-tuned forms, without any additional fine-tuning.

Benchmarks. We evaluate on three benchmarks covering distinct task types. For mathematics, we use MATH500 (Lightman et al., 2024), which requires multi-step quantitative reasoning. For code, we use HumanEval (Chen et al., 2021), a set of Python programming tasks scored by unit tests. For question answering, we use GPQA (Rein et al., 2024), which covers graduate-level questions in biology, physics, and chemistry.

Baselines. We compare CBS against standard autoregressive sampling, low-temperature sampling, and beam search in our main results. Standard and low-temperature sampling use temperatures of $\tau = 1.0$ and $\tau = 0.25$, respectively. The beam search baseline (Snell et al., 2025) ranks partial trajectories using only the base-model log-likelihood. We additionally compare against Power Sampling Ji et al. (2026), a training-free and verifier-free test-time search method. Across all methods, we use the same task prompts and cap the maximum generation length at 3072 tokens. For standard beam search, we use a candidate pool of $N = 16$ beams and an expansion beam width of $W = 4$.

Model	Method	MATH500	HumanEval	GPQA
Qwen3.5-0.8B	Base $\tau = 1.0$	0.198	0.085	0.020
	Low Temperature $\tau = 0.25$	0.342	0.225	0.101
	Beam Search	0.398	0.207	0.091
	Power Sampling $\tau = 0.25$	0.430	0.287	**0.131**
	Power Sampling $\tau = 1.0$	0.300	0.262	0.060
	CBS $1/\alpha = 0.25$	**0.480**	0.238	0.106
	CBS $1/\alpha = 0.7$	0.436	**0.329**	**0.131**
Qwen 2.5-7B	Base $\tau = 1.0$	0.326	0.500	0.222
	Low Temperature $\tau = 0.25$	0.508	0.677	0.288
	Beam Search	0.570	0.707	0.247
	Power Sampling $\tau = 0.25$	0.626	0.738	0.283
	Power Sampling $\tau = 1.0$	0.450	0.665	0.242
	CBS $1/\alpha = 0.25$	0.610	**0.744**	**0.313**
	CBS $1/\alpha = 0.7$	**0.640**	0.732	0.235
Qwen3.5-2B	Base $\tau = 1.0$	0.444	0.439	0.106
	Low Temperature $\tau = 0.25$	0.632	0.600	0.217
	Beam Search	0.656	0.560	0.177
	Power Sampling $\tau = 0.25$	0.646	0.561	0.241
	Power Sampling $\tau = 1.0$	0.578	0.524	0.176
	CBS $1/\alpha = 0.25$	0.672	**0.604**	0.247
	CBS $1/\alpha = 0.7$	**0.694**	0.585	**0.258**
Deepseek-Math -7B-Instruct	Base $\tau = 1.0$	0.342	0.335	0.298
	Low Temperature $\tau = 0.25$	0.436	0.445	0.298
	Beam Search	0.432	0.494	0.293
	Power Sampling $\tau = 0.25$	0.430	**0.537**	0.278
	Power Sampling $\tau = 1.0$	0.410	0.500	0.303
	CBS $1/\alpha = 0.25$	**0.462**	0.512	**0.369**
	CBS $1/\alpha = 0.7$	0.458	0.518	0.258

Table 1: Performance comparison of CBS across 4 models and 3 benchmarks.



(a) Qwen-2.5-7B (b) Deepseek-Math-7B-Instruct

Figure 1: Inference efficiency of (a) Qwen2.5-7B and (b) DeepSeek-Math-7B-Instruct on the MATH500 benchmark.

Main Results. Table 1 reports pass@1 accuracy on MATH500, HumanEval and GPQA for four models. All comparisons in this work are absolute differences in accuracy, reported in percentage points. Across all 12 settings, CBS outperforms both low-temperature sampling and standard beam search, with gains of up to 13.8% and 12.2%, respectively. The margin over standard beam search indicates that the improvements cannot be attributed to multi-trajectory search alone. Against power sampling the margin is narrower: CBS wins or tied-best in 11 of the 12 settings, by up to 6.6%. Overall, CBS is best or tied-best in 11 of the 12 settings, and it leads on MATH500 and GPQA for all four models. The preferred coefficient is model and task-dependent.

Power sampling relies on a sharpened sampling distribution. Its $\tau = 0.25$ variant beats its $\tau = 1.0$ variant in 11 of the 12 settings. CBS improves accuracy for every model, which we attribute to reweighting tokens by the contrast between the positive and the negative contexts rather than by a single temperature. But, more importantly, CBS has the exact opposite behavior of Power Sampling concerning temperature. It works better with higher temperature, which implies that it is also able to produce more diverse trajectories as we will show in the next pass@k experiment.

Inference Efficiency. We next examine the inference cost of CBS. Figure 1 reports wall-clock inference time and total completion tokens per prompt, counting every candidate generated rather than only the selected one, for Qwen2.5-7B and DeepSeek-Math-7B-Instruct on MATH500. The standard sampling baselines are the fastest. Standard beam search costs about 2 times more. CBS is more expensive, requiring roughly $1.5\times$ the cost of standard beam search for both models. This overhead does not come from longer outputs. CBS returns 20% fewer completion tokens than beam search for Qwen2.5-7B and 11% fewer for DeepSeek-Math-7B-Instruct, yet it remains slower than beam search on both models. The added latency therefore reflects the repeated scoring of each surviving candidate under the positive and negative contexts rather than additional generation. On Qwen2.5-7B the lower α setting is both faster and more token-efficient, while on DeepSeek-Math-7B-Instruct the higher α wins on both counts.

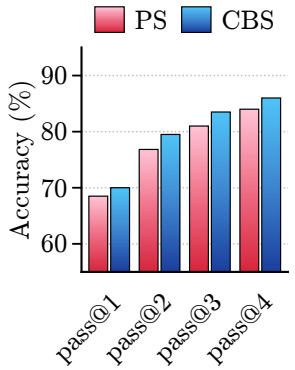


Figure 2: Pass@k on the MATH500 dataset between Power sampling and CBS (Qwen3.5-2B).

Pass@k Results. We report pass@k performance by sampling k independent completions on the MATH500 dataset obtained with Qwen3.5-2B in Figure 2. Overall, CBS consistently surpasses power sampling at every evaluated value of k. While pass@k inherently increases for any method, CBS explicitly

Context	Content	MATH500
No context	-	0.398
Contrastive	`Positive: ''This is an example for a response with excellent reasoning:''` `Negative: ''This is an example for a response with wrong reasoning:''`	**0.509**
Neutral	`Positive: ''This is an example for a response to the question assigned to control group one:''` `Negative: ''This is an example for a response to the question assigned to control group two:''`	0.380

Table 2: Context ablation for Qwen3.5-0.8B on MATH500.

steers the decoding process away from common reasoning pitfalls and toward the correct trajectory. Further pass@k results can be found in Appendix C.1.

4.2 Ablations and Analysis

Effect of Contrastive Context Semantics. We conduct a context ablation to determine whether the gains from contrastive scoring arise from the semantic opposition between the positive and negative reasoning contexts, rather than from added context alone. Using Qwen3.5-0.8B, we fix the model and decoding configuration and compare three conditions: no context, the proposed contrastive contexts (excellent versus wrong reasoning), and a neutral control context that assigns responses to two groups without expressing a quality preference. As shown in Table 2, the contrastive context improves accuracy by 11.1% and 12.9% over the empty-context and neutral-context conditions, respectively. The neutral context does not improve over the empty-context condition, suggesting that the gain is not explained by the presence of additional prompt text alone. Instead, these results support the interpretation that the contrastive semantic signal steers the model toward higher-performing responses on MATH500. See Appendix C.2 for an extended ablation of various contexts.

Effect of beam-population budget. Because CBS is taking more search time, we increase the standard beam search population from $N = 16$ to $N = 24$ to test whether allocating more candidate trajectories allows likelihood-based search to overtake CBS with $N = 16$. Relative to the mean of the two CBS settings, beam search at N = 24 uses $1.7\times$ as many completion tokens and takes $1.1\times$ to $1.4\times$ as long on Qwen2.5-7B and DeepSeek-Math-7B-Instruct respectively (Figure 1).

Method	Qwen3.5-0.8B	Qwen2.5-7B	Qwen3.5-2B	DeepSeek-Math-7B-Instruct
Beam search ($N = 24$)	0.391	0.600	0.676	0.452
CBS $1/\alpha = 0.25$	**0.480**	0.610	0.672	**0.462**
CBS $1/\alpha = 0.7$	0.436	**0.640**	**0.694**	0.458

Table 3: Beam-population budget ablation on MATH500.

As shown in Table 3, for every model, at least one CBS setting outperforms the larger standard beam search baseline. Thus, increasing the standard beam population can narrow the gap, but does not uniformly recover the gains from CBS, supporting the value of the contrastive ranking signal beyond simply expanding the candidate budget.

Effect of beam search block size. We vary the block size K to control how often contrastive scoring intervenes during beam search using Qwen3.5-0.8B on MATH500. As shown in Figure 3, accuracy follows an inverted-U pattern: it rises to a peak at $K = 32$, and then declines monotonically. Large blocks leave few opportunities to redirect unproductive trajectories, while $K = 16$ reranks on too little continuation context to score trajectories reliably. Hence, $K = 32$ balances these effects.

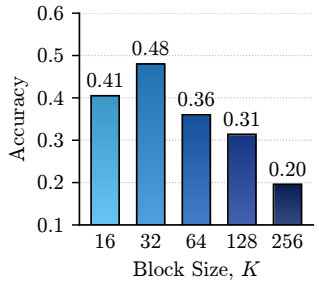


Figure 3: Block size K effect on CBS.

5 Related Work

Inference-Time Search. Test-time scaling improves language-model reasoning by allocating additional computation during inference rather than changing model parameters (Snell et al., 2025; Welleck et al., 2024; Ji et al., 2025). Parallel approaches generate multiple complete reasoning trajectories and aggregate their answers, for example through majority voting in self-consistency (Wang et al., 2023) or through verifier and reward-model scores (Lightman et al., 2024; Snell et al., 2025). Their inference cost grows with the number and length of sampled trajectories, and the most effective allocation of a fixed compute budget depends on problem difficulty (Snell et al., 2025). DeepConf filters low-confidence reasoning traces during or after generation (Fu et al., 2026), while Confidence-Informed Self-Consistency weights sampled answers by model-reported confidence to reduce the number of trajectories needed for aggregation (Taubenfeld et al., 2025). Sequential approaches instead spend inference compute refining a single response, as in Self-Refine's iterative feedback-and-revision loop (Madaan et al., 2023). Search-based methods allocate compute within generation: Tree of Thoughts explores explicit intermediate reasoning states (Yao et al., 2023), ARGS adjusts token probabilities using an external reward signal (Khanov et al., 2024), and Tree-BoN branches and prunes partial responses using token-level rewards derived from Direct Preference Optimization (Qiu et al., 2025). Our method belongs to this search-based family, but differs from conventional token-level beam search: it stochastically expands trajectories with multi-token chunks, scores their accumulated responses, and retains a fraction for further expansion. We therefore use beam-style search as the computational mechanism for allocating test-time compute, rather than treating the search algorithm itself as the contribution.

Contrastive and Context-Conditioned Decoding. Contrastive Decoding compares an expert and an amateur language model and favors tokens preferred by the expert, subject to a plausibility constraint (Li et al., 2023). DoLa obtains a contrastive signal within a single model by comparing logits from later and earlier layers (Chuang et al., 2024), while Asymptotic Probability Decoding interprets and modifies contrastive decoding through extrapolation toward a hypothetical larger model (Chang et al., 2024). Other methods construct contrasts through the input context: Context-Aware Decoding compares predictions with and without supporting context (Shi et al., 2024), and factual-versus-hallucination prompting contrasts output distributions induced by opposing prompts (Lv et al., 2024; Yang et al., 2025). Thinking by Subtraction applies contrastive correction selectively at low-confidence positions during reasoning (Tang et al., 2026). These methods primarily intervene in next-token prediction. In contrast, we use the likelihood difference induced by positive and negative reasoning contexts to score an accumulated trajectory after each sampled chunk. This contrastive score determines which partial trajectories receive further test-time computation, connecting context-conditioned decoding to the beam-style search procedure above.

6 Conclusion and Future Work

We have shown that the self-distillation principle, i.e. using a model's own conditional distributions as a supervisory signal, can be operationalized at test time. By replacing expert demonstrations with counterfactual contexts, we derived a contrastive PMI reward whose KL-regularized optimum is a Gibbs reweighting of the base policy. We analyzed that this reweighting is fundamentally global: naive per-token decoding ignores a future correction term. Contrastive beam search approximates the target distribution by steering generation block by block, and experiments across four models and three benchmarks confirm that it outperforms sampling, beam search, and power sampling baselines in average. Ablations show that the gains arise from the semantic opposition of the contrastive contexts and from intermediate steering during generation, not from increased inference time.

Several directions remain open. The current contrastive contexts are fixed and manually designed; learning or searching for stronger context pairs, potentially in a task-adaptive manner, could improve the steering signal. Reducing the scoring overhead, for example through KV-cache sharing across the three context-conditioned forward passes, would lower the practical cost. More broadly, we view test-time self-distillation as a step toward continual learning agents that leverage their own internal distributions to adapt behavior at inference time: an agent could systematically propose and evaluate counterfactual contexts to guide multi-step planning and action selection without parameter updates.

AI USE STATEMENT

In this work, we used generative AI for text refinement, grammatical editing, and generating code to plot figures. We did not use AI to formulate our scientific claims, mathematical derivations, or experimental results. We manually reviewed all AI-generated text for accuracy, and LLM-generated code was verified and tested for correctness. We take full responsibility for the final content of this paper.

ETHICS STATEMENT

This research complies with the ICLR Code of Ethics. All experiments were conducted using publicly available benchmark datasets free of personally identifiable information, and the study involves no human subjects or new data collection. Because our focus is strictly on improving large language model performance during test-time inference, this work does not involve the deployment of systems that cause negative societal impacts.

REPRODUCIBILITY STATEMENT

Our code will be released upon acceptance. All benchmark datasets (MATH500, HumanEval, GPQA) and models (Qwen2.5-7B, Qwen3.5-2B, Qwen3.5-0.8B, DeepSeek-Math-Instruct) used in this work are publicly available. To ensure reproducibility, we have included complete technical details for our proposed algorithms, mathematical derivations, and theoretical assumptions throughout the main text and appendices. Appendix B provides further experimental setups such as prompts, hyperparameter settings, and evaluation protocols. Additionally, all theoretical results and their underlying assumptions are formally stated in Propositon 1, Theorem 1 and in Appendix A.

REFERENCES

Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. Explaining and improving contrastive decoding by extrapolating the probabilities of a huge and hypothetical lm. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://arxiv.org/abs/2411.01610.

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

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Th6NyL07na.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8LqHs0KIM7.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Re-

inforcement learning via self-distillation, 2026. URL https://arxiv.org/abs/2601.20802.

Xiaotong Ji, Shyam Sundhar Ramesh, Matthieu Zimmer, Ilija Bogunovic, Jun Wang, and Haitham Bou Ammar. On almost surely safe alignment of large language models at inference-time, 2025. URL https://arxiv.org/abs/2502.01208.

Xiaotong Ji, Rasul Tutunov, Matthieu Zimmer, and Haitham Bou Ammar. Scalable power sampling: Unlocking efficient, training-free reasoning for llms via distribution sharpening, 2026. URL https://arxiv.org/abs/2601.21590.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think, 2025. URL https://arxiv.org/abs/2510.14901.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. ARGS: Alignment as reward-guided search. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=shgx0eqdw6.

Tomasz Korbak, Ethan Perez, and Christopher L. Buckley. Rl with kl penalties is better viewed as bayesian inference. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://api.semanticscholar.org/CorpusID:248987624.

David A Levin and Yuval Peres. *Markov chains and mixing times*. American Mathematical Society, 2026.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Bojie Lv, Ao Feng, and Chenlong Xie. Improving factuality by contrastive decoding with factual and hallucination prompts. *Sensors*, 2024. URL https://www.mdpi.com/1424-8220/24/21/7097.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Tu Nguyen, Matthieu Zimmer, Rasul Tutunov, Xiaotong Ji, and Haitham Bou Ammar. The model knows, the decoder finds: Future value guided particle power sampling, 2026. URL https://arxiv.org/abs/2605.02427.

Jiahao Qiu, Yifu Lu, Yifan Zeng, Jiacheng Guo, Jiayi Geng, Chenhao Zhu, Xinzhe Juan, Ling Yang, Huazheng Wang, Kaixuan Huang, Yue Wu, and Mengdi Wang. TreeBoN: Enhancing inference-time alignment with speculative tree-search and best-of-n sampling. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025. URL https://aclanthology.org/2025.findings-emnlp.1140/.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, 2024. URL https://aclanthology.org/2024.naacl-short.69/.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Lexiang Tang, Weihao Gao, Bingchen Zhao, Lu Ma, Qiao Jin, Bang Yang, and Yuexian Zou. Thinking by subtraction: Confidence-driven contrastive decoding for llm reasoning, 2026. URL https://doi.org/10.48550/arXiv.2602.18232.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20090–20111. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1030. URL http://dx.doi.org/10.18653/v1/2025.findings-acl.1030.

Alexandre B Tsybakov and Vladimir Zaiats. *Introduction to nonparametric estimation*, volume 11. Springer, 2009.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=eskQMcIbMS. Survey Certification.

Dingkang Yang, Dongling Xiao, Jinjie Wei, Mingcheng Li, Zhaoyu Chen, Ke Li, and Lihua Zhang. Improving factuality in large language models via decoding-time hallucinatory and truthful comparators. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence*, 2025. URL https://doi.org/10.1609/aaai.v39i24.34751.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2023.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A PROOFS

Proposition 1 is the contrastive-reward equivalent of a known result on the soft-Bellman view of entropy-regularized reinforcement learning (Haarnoja et al., 2018; Ziebart et al., 2008). Our contribution is its application to the contrastive PMI reward Eq. 1 and the resulting characterization of how truncating the future correction shapes the approximation error of contrastive beam search.

Proof of Proposition 1. Fix t and partial sequence $a_{<t}$. The per-token conditional is obtained by marginalizing over future continuations:

$$\begin{aligned}\pi_{\text{target}}(a_t \mid q, a_{<t}) &= \sum_{a_{t+1:T}} \pi_{\text{target}}(a_{t:T} \mid q, a_{<t}) \\ &= \frac{\sum_{a_{t+1:T}} \prod_{s=t}^{T} \pi_\theta(a_s \mid q, a_{<s}) \exp\big(\frac{1}{\alpha} r_s(a_s, q)\big)}{\sum_{a_{t:T}} \prod_{s=t}^{T} \pi_\theta(a_s \mid q, a_{<s}) \exp\big(\frac{1}{\alpha} r_s(a_s, q)\big)}.\end{aligned}$$

Factoring the numerator by separating position t from future terms $s > t$:

$$\sum_{a_{t+1:T}} \prod_{s=t}^{T} \pi_\theta(a_s \mid q, a_{<s}) \exp\left(\frac{1}{\alpha} r_s(a_s, q)\right) =$$

$$\pi_\theta(a_t \mid q, a_{<t}) \exp\left(\frac{1}{\alpha} r_t(a_t, q)\right) \cdot \underbrace{\sum_{a_{t+1:T}} \prod_{s=t+1}^{T} \pi_\theta(a_s \mid q, a_{<t}, a_t, a_{<s}) \exp\left(\frac{1}{\alpha} r_s(a_s, q)\right)}_{\zeta_t(a_t, q)}.$$

Applying the same factorization to each a'_t in the denominator yields Eq. 7. For the final token ($t = T$), the future correction reduces to $\zeta_T = 1$ (empty product), recovering the naive per-token distribution. □

Lemma 1 (TV Distance under Bounded Reweighting)**.** *Let* $p(a) = \frac{w(a)}{\sum_{a'} w(a')}$ *and* $q(a) = \frac{w(a) f(a)}{\sum_{a'} w(a') f(a')}$ *for positive weights* $w(a) > 0$ *and bounded function* $f(a) \in [c, C]$ *with* $0 < c \leq C < \infty$*. Then*

$$\|p - q\|_{\text{TV}} \leq \frac{C - c}{C + c}.$$

Proof. For any event A:

$$|p(A) - q(A)| = \left| \sum_{a \in A} w(a) \left(\frac{1}{Z_w} - \frac{f(a)}{Z_{wf}} \right) \right| = \left| \sum_{a \in A} \frac{w(a)}{Z_w} \left(1 - \frac{f(a) Z_w}{Z_{wf}} \right) \right|.$$

Since $c \leq f(a) \leq C$, we have $\frac{c\, Z_w}{Z_{wf}} \leq \frac{f(a) Z_w}{Z_{wf}} \leq \frac{C\, Z_w}{Z_{wf}}$. Noting that $\frac{Z_w}{Z_{wf}} \in [\frac{1}{C}, \frac{1}{c}]$ (because $c\, Z_w \leq Z_{wf} \leq C\, Z_w$), the ratio $\frac{f(a) Z_w}{Z_{wf}}$ ranges in $[\frac{c}{C}, \frac{C}{c}]$. Using the equivalent form for total variation distance:

$$||p - q||_{\text{TV}} = \frac{1}{2} \sum_a |p(a) - q(a)| = \frac{1}{2} \sum_a p(a) \left| 1 - \frac{q(a)}{p(a)} \right| = \frac{1}{2} \sum_a p(a) \left| 1 - \frac{f(a) Z_w}{Z_{wf}} \right| \quad (11)$$

Let us study the expression $\left| 1 - \frac{f(a) Z_w}{Z_{wf}} \right|$ with more care. Let $g(r) = \frac{|1 - r|}{1 + r}$ and parameter r belong to the interval $r \in \left[\frac{c}{C}, \frac{C}{c}\right]$. We can make the following two observations:

1. If $r \geq 1$, then $g(r) = \frac{r-1}{r+1}$ and $g'(r) = \frac{2}{(r+1)^2} > 0$ for any r. Hence, for $r \in [1, \frac{C}{c}]$ due to monotonic increment of function $g(r)$ we can write:

$$g(r) \leq g\left(\frac{C}{c}\right) = \frac{\frac{C}{c} - 1}{\frac{C}{c} + 1} = \frac{C - c}{C + c}$$

2. If $r < 1$, then $g(r) = \frac{1-r}{1+r}$ and $g'(r) = -\frac{2}{(r+1)^2} < 0$ for any r. Hence, for $r \in [\frac{c}{C}, 1)$ due to monotonic decrement of function $g(r)$ we can write:

$$g(r) \leq g\left(\frac{c}{C}\right) = \frac{1-\frac{c}{C}}{\frac{c}{C}+1} = \frac{C-c}{C+c}$$

Because $\frac{f(a)Z_w}{Z_{wf}} \in \left[\frac{c}{C}, \frac{C}{c}\right]$ we have from the above results:

$$\left|1 - \frac{f(a)Z_w}{Z_{wf}}\right| \leq \frac{C-c}{C+c}\left[1 + \frac{f(a)Z_w}{Z_{wf}}\right]$$

Applying this result in Eq 11 gives:

$$||p-q||_{\mathrm{TV}} = \frac{1}{2}\sum_a p(a)\left|1 - \frac{f(a)Z_w}{Z_{wf}}\right| \leq \frac{1}{2}\frac{C-c}{C+c}\sum_a p(a)\left[1 + \frac{f(a)Z_w}{Z_{wf}}\right] =$$

$$\frac{1}{2}\frac{C-c}{C+c}\left[\sum_a p(a) + \sum_a \frac{w(a)}{Z_w}\frac{f(a)Z_w}{Z_{wf}}\right] = \frac{1}{2}\frac{C-c}{C+c}\left[1 + \sum_a \frac{f(a)w(a)}{Z_{wf}}\right] = \frac{2}{2}\frac{C-c}{C+c} = \frac{C-c}{C+c}$$

□

This is the classical bounded-likelihood-ratio bound on total variation; we restate it for completeness (Tsybakov & Zaiats, 2009).

Proof of Theorem 1. **Part (i).** At a scoring point at position t (start of a chunk), the K tokens within the chunk (positions $t{+}1$ through $t{+}K$) are scored exactly as part of the accumulated reward. The future correction accounts for the remaining $T - t - K$ tokens after the chunk:

$$\zeta_t(a_t, q) = \mathbb{E}_{a_{t+K+1:T}\sim\pi_\theta}\left[\prod_{s>t+K} \exp(r_s/\alpha)\right].$$

Since $|r_s| \leq r_{\max}$ and there are $T - t - K$ future tokens:

$$e^{-(T-t-K)r_{\max}/\alpha} \leq \prod_{s>t+K} e^{r_s/\alpha} \leq e^{(T-t-K)r_{\max}/\alpha} \quad \Longrightarrow \quad \zeta_t(a_t, q) \in \left[e^{-(T-t-K)r_{\max}/\alpha},\ e^{(T-t-K)r_{\max}/\alpha}\right].$$

The truncation estimator sets $\hat{\zeta}_t \equiv 1$. Applying Lemma 1 with $c = e^{-(T-t-K)r_{\max}/\alpha}$, $C = e^{(T-t-K)r_{\max}/\alpha}$, and $f = \zeta_t$:

$$\|\tilde{\pi}_{\mathrm{trunc}} - \pi_{\mathrm{target}}\|_{\mathrm{TV}} \leq \frac{e^{(T-t-K)r_{\max}/\alpha} - e^{-(T-t-K)r_{\max}/\alpha}}{e^{(T-t-K)r_{\max}/\alpha} + e^{-(T-t-K)r_{\max}/\alpha}} = \tanh\left(\frac{(T-t-K)\,r_{\max}}{\alpha}\right).$$

Scoring decisions occur at chunk boundaries $t = 0, K, 2K, \ldots, T{-}K$, giving T/K decision points. By the standard Markov chain perturbation bound (Levin & Peres, 2026), the trajectory-level error compounds:

$$\|\tilde{\pi}_{\mathrm{traj}} - \pi_{\mathrm{target}}\|_{\mathrm{TV}} \leq \sum_{j=0}^{T/K-1} \tanh\left(\frac{(T-jK-K)\,r_{\max}}{\alpha}\right) =$$

$$\sum_{m=1}^{T/K} \tanh\left(\frac{(T-mK)\,r_{\max}}{\alpha}\right) \leq \frac{r_{\max}}{\alpha}\frac{T(T-K)}{2K},$$

using $\tanh(x) \leq x$ and the identity $\sum_{m=1}^{n}(T-mK) = K\cdot\frac{n(n-1)}{2} = \frac{T(T-K)}{2K}$ with $n = T/K$. For comparison, naive per-token sampling (Eq. 6) truncates at every token with horizon $T - t$, giving the looser bound $\frac{r_{\max}}{\alpha}\frac{T(T-1)}{2}$.

Part (ii). Under the zero-mean condition $\mathbb{E}_{a_t\sim\pi_\theta(\cdot|q,a_{<t})}[r_t(a_t, q)] = 0$, the variables $\{r_s(a_s, q)/\alpha\}_{s>t+K}$ are bounded martingale differences: $|r_s/\alpha| \leq r_{\max}/\alpha$ and $\mathbb{E}[r_s/\alpha \mid q, a_{<s}] = 0$. The future correction is the moment-generating function of their sum:

$$\zeta_t(a_t, q) = \mathbb{E}\left[\exp\left(\sum_{s>t+K}\frac{r_s}{\alpha}\right)\right].$$

By Hoeffding's lemma applied conditionally to each martingale difference and iterated via the tower property:

$$\zeta_t(a_t, q) \le \exp\left(\frac{(T-t-K)\, r_{\max}^2}{2\alpha^2}\right).$$

By Jensen's inequality, since $\mathbb{E}\big[\sum_{s>t+K} r_s/\alpha \,\big|\, q, a_{<t}, a_t\big] = 0$:

$$\zeta_t(a_t, q) \ge \exp\left(\mathbb{E}\left[\sum_{s>t+K} \frac{r_s}{\alpha}\right]\right) = 1.$$

Applying Lemma 1 with $c = 1$ and $C = \exp\big((T-t-K)\, r_{\max}^2/(2\alpha^2)\big)$:

$$\|\tilde{\pi}_{\text{trunc}} - \pi_{\text{target}}\|_{\text{TV}} \le \frac{e^{(T-t-K)r_{\max}^2/(2\alpha^2)} - 1}{e^{(T-t-K)r_{\max}^2/(2\alpha^2)} + 1} = \tanh\left(\frac{(T-t-K)\, r_{\max}^2}{4\alpha^2}\right).$$

The trajectory bound follows identically to part (i), summing over T/K scoring points:

$$\sum_{m=1}^{T/K} \tanh\left(\frac{(T-mK)\, r_{\max}^2}{4\alpha^2}\right) \le \frac{r_{\max}^2}{4\alpha^2}\frac{T(T-K)}{2K}. \qquad \square$$

B Experimental Setups and Implementation Details

Prompts. Table 4 lists the task prompt templates used for MATH500, HumanEval, and GPQA. All decoding methods within a model–benchmark setting use the same task prompt, so comparisons vary only the decoding and candidate-ranking procedure. For contrastive scoring, the positive or negative context is appended to the user message after the original query and immediately before the assistant-generation marker. Denoting the system instruction by s, the user query by q, and either context by $c^{\pm}$, the context-conditioned serialization is

$$[\textsc{System} : s][\textsc{User} : q \oplus c^{\pm}][\textsc{Assistant} :], \tag{12}$$

where $\oplus$ denotes textual concatenation. The system instruction is unchanged across the base, positive, and negative evaluations. The base context contains only q in the user message, while the context-conditioned prompts append c^+ or c^-, respectively. These contexts affect candidate scoring but are not included in the returned response. Table 5 gives a complete positive-context example using the Qwen chat format; the negative context is formed by replacing the final positive context with "This is an example for a response with wrong reasoning:".

Hyperparameter settings. Table 6 summarizes the principal settings used in the main evaluation. All methods use a block size of $K = 32$, 96 generation iterations, and a maximum generation length of $T_{\max} = 3072$ tokens, with early termination upon emitting an EOS token. Standard and low-temperature sampling use a single trajectory with $N = W = 1$ and disable contrastive scoring. Standard sampling, beam search, and both contrastive beam search variants use a generation temperature of $\tau = 1.0$, while the low-temperature baseline uses $\tau = 0.25$. Standard and contrastive beam search use an active population of $N = 16$ and pruning factor $W = 4$, retaining the top $N/W = 4$ unfinished trajectories after each scoring stage. These settings are used throughout the main experiments unless explicitly varied in an ablation. The baseline ranks candidates using only the mean base-context log-likelihood ($1/\alpha = 0$), whereas the contrastive variants use $1/\alpha \in \{0.25, 0.7\}$. To ensure numerical stability, all likelihood computations are carried out in log-space.

Inference efficiency measurement. We measure all inference efficiency in Figure 1 on a single accelerator with the same runtime stack. Each time is the wall-clock time per prompt for generation and candidate scoring, and excludes model loading, dataset loading, and metric computation. We count completion tokens as all generated tokens per prompt across every candidate, not just the final answer, and exclude prompt tokens and the scoring passes.

Task	Prompt Template
MATH500	**System:** You are a helpful AI Assistant that provides well-reasoned and detailed responses. You first think about the reasoning process as an internal monologue and then provide the user with the boxed answer. Respond in the following format: <think> ... </think> <answer> \boxed{...} </answer>. **User:** {problem}
HumanEval	**System:** Please reason step by step internally. Then output ONLY the final Python code that completes the task within triple backticks. Do not include explanations, markdown, or \boxed{}. **User:** Complete the following Python function: {prompt}
GPQA	**System:** You are a helpful AI Assistant. Please reason step by step, and put your final answer within \boxed{}. **User:** Answer the following multiple choice question. The last line of your response should be of the following format: '\boxed{$LETTER}' (without quotes) where LETTER is one of ABCD (ex. '\boxed{A}'). Think step by step before answering. {question} A) {choice_A} B) {choice_B} C) {choice_C} D) {choice_D}

Table 4: Task Prompt Templates

Role	Serialized Qwen prompt content
System	<|im_start|>system You are a helpful AI Assistant that provides well-reasoned and detailed responses. You first think about the reasoning process as an internal monologue and then provide the user with the boxed answer. Respond in the following format: <think> ... </think> <answer> \boxed{...} </answer>. <|im_end|>
User query and positive context	<|im_start|>user Solve for x: $2^{x+1} = 32$. This is an example for a response with excellent reasoning: <|im_end|>
Assistant prefix	<|im_start|>assistant

Table 5: Example serialization of a MATH500-style query with the positive contrastive context under the Qwen chat template.

Method	Hyperparameters
Standard sampling	$N = 1, W = 1, K = 32, \tau = 1.0$
Low-temperature sampling	$N = 1, W = 1, K = 32, \tau = 0.25$
Beam search	$N = 16, W = 4, K = 32, \tau = 1.0, 1/\alpha = 0$
Contrastive beam search (low α)	$N = 16, W = 4, K = 32, \tau = 1.0, 1/\alpha = 0.25$
Contrastive beam search (high α)	$N = 16, W = 4, K = 32, \tau = 1.0, 1/\alpha = 0.7$
Power Sampling	$K_t = 16$, $M_t = 16$, $B = 32$ (equivalent to K), $\tau = 0.25$ (low) and $\tau = 1.0$ (high), $\alpha = 4$
Best-of-N	$N = 16, \tau = 1.0$

Table 6: Main hyperparameter settings for each method.

C Further Experimental Results

C.1 Additional Pass@k results and analysis.

Figure 4 reports pass@k on MATH500 for DeepSeek-Math-7B-Instruct. Standard beam search marginally surpass power sampling at $k = 1$ and $k = 2$ but yields diminishing returns, falling behind at $k = 3$ and $k = 4$. In contrast, CBS and power sampling scale consistently, with CBS outperforming both methods across all k. Because both beam search variants use a temperature of $\tau = 1.0$ to maintain generation diversity, this divergence demonstrates that unguided likelihood maximization is insufficient for complex reasoning. Instead, the contrastive contexts in CBS effectively steer the search trajectory toward correct solutions.

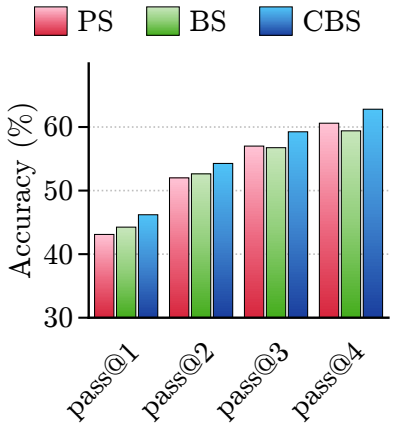


Figure 4: Pass@k performance $k \in \{1, 2, 3, 4\}$ on MATH500 dataset for Power sampling (PS), standard beam search (BS) and contrastive beam search (CBS) using Deepseek-Math-7B-Instruct.

C.2 Context content ablation

Having established that performance gains arise from the contrastive semantic signal, we further explore how context content impacts performance. We generate 10 contrastive contexts pairs with varying contexts (Table 7). Each positive contexts is paired with its counterfactual negative (e.g., "excellent reasoning" vs. "wrong reasoning" or "complete" vs. "incomplete" treatment). We use Best-of-N to generate $N = 16$ candidates, recompute log probabilities under the positive and negative prefixes, and rank them by $lp_{\text{base}} + \frac{1}{\alpha} \cdot (lp_{+} - lp_{-})$. We test this across three benchmarks with distinct demands: MATH500 (arithmetic reasoning), HumanEval (functional correctness in coding), and GPQA (expert-level science).

As shown in Figure 5 , on MATH500 dataset, the reasoning and step verification contexts yield the highest accuracy. This aligns with how math models usually fail, where one incorrect step breaks the whole solution, making these specific contexts highly effective. The remaining contexts perform near the baseline, with coherence and self correction ranking lowest. Conversely, HumanEval task exhibits minimal contexts differentiation. Most contexts, including reasoning, completeness, reliability, and logical validity, cluster at similar accuracy levels, while coherence and decomposition fall marginally below the baseline. Because the reranker evaluates text rather than executing the code, modifying the evaluation criteria provides no leverage for detecting hidden runtime errors. Similar to MATH500, reasoning remains the strongest context on GPQA, but the subsequent rankings shift to favor reliability and logical validity. This benchmark evaluates complex scientific questions that depend on strict factual consistency and sound argumentation. This explains why logical validity and reliability provide an advantage while structural contexts like coherence do not. Although contrastive contexts yield only marginal accuracy gains during best-of-N terminal reranking, our results demonstrate that applying them during intermediate steering produces substantial performance improvements.

Content	Contrastive Context Pair
Reasoning	Positive: ``This is an example for a response with excellent reasoning:'' Negative: ``This is an example for a response with wrong reasoning:''
Completeness	Positive: ``This is an example for a response with a complete treatment of the problem:'' Negative: ``This is an example for a response with an incomplete treatment of the problem:''
Step verification	Positive: ``This is an example for a response that carefully verifies each step of its reasoning:'' Negative: ``This is an example for a response that skips verification and rushes to a conclusion:''
Reliability	Positive: ``This is an example for a response that is reliable and trustworthy:'' Negative: ``This is an example for a response that is unreliable and error-prone:''
Logical validity	Positive: ``This is an example for a response with logically sound arguments:'' Negative: ``This is an example for a response with logical fallacies:''
Reviewer judgment	Positive: ``This is an example for a response that a careful reviewer would approve without hesitation:'' Negative: ``This is an example for a response that a careful reviewer would reject immediately:''
Coherence	Positive: ``This is an example for a response that is internally consistent from start to finish:'' Negative: ``This is an example for a response that contradicts itself partway through:''
Decomposition	Positive: ``This is an example for a response that breaks the problem into clear, manageable steps:'' Negative: ``This is an example for a response that attempts the problem in one confused leap:''
Attention to detail	Positive: ``This is an example for a response that pays careful attention to every detail:'' Negative: ``This is an example for a response that overlooks important details:''
Self-correction	Positive: ``This is an example for a response that catches and corrects its own mistakes:'' Negative: ``This is an example for a response that persists in its mistakes without noticing them:''

Table 7: Examples of 10 contrastive context pairs. Every pair shares same initial framing with "This is an example for a response . . . ". Each positive context is paired with its counterfactual negative.

Overall, the generic reasoning context consistently performs best across all three benchmarks. However, performance of secondary contexts is domain-dependent, such as step verification for math and logical validity for science, demonstrating that the nature of the task determines which context works best.

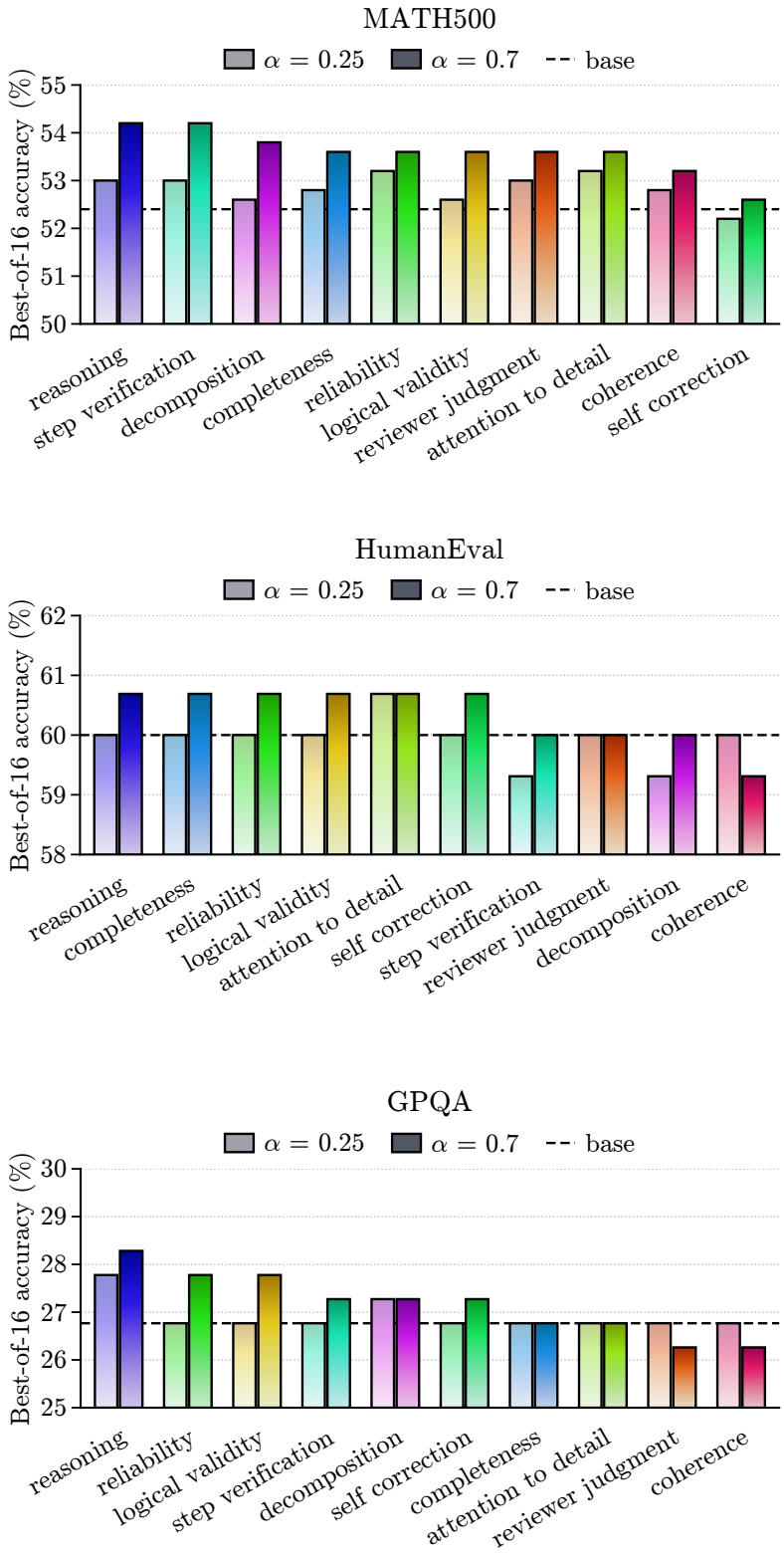


Figure 5: Best-of-16 accuracy for ten contrastive context pairs on MATH500, HumanEval and GPQA (Qwen2.5-7B), at the two deployed steering strengths. Dashed line: no-context baseline.