EXPBON: EXPONENTIAL-NOISE BEST-OF-n FOR EFFICIENT TEST-TIME LLM ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Best-of-n (BoN) sampling is a simple yet effective inference-time alignment method, but hard maximization provides only coarse control over the trade-off between reward and distribution shift. Soft Best-of-n by Verdun et al. (2025) provides smoother control and converges to the optimal distribution associated with KL-regularized reward maximization, but its known guarantees decay only polynomially in n. In this paper, we introduce ExpBoN, an alternative soft BoN method based on the exponential-noise report-noisy-max mechanism: it perturbs the scaled rewards with exponential instead of Gumbel noise before the argmax. Its output distribution admits an exact finite-n decomposition into the optimal tilted distribution plus a residual whose weight vanishes geometrically, and this single structural property has two consequences. Statistically, it yields geometric convergence in total variation, expected reward, and both directions of KL divergence, together with regret guarantees under proxy-reward misspecification. Computationally, it permits an exact early exit: candidates can be examined sequentially in random order, and the first whose noisy score crosses a fixed threshold is already an exact sample, so the remaining candidates need not be scored. As a case study, we integrate ExpBoN into the guided speculative inference (GSI) framework by Geuter et al. (2026), where every candidate score requires both a reward-model and a target-model evaluation. On MATH500, MMLU-STEM, and Minerva Math with the Qwen2.5-Math and Qwen3 model families, the resulting ExpGSI maintains comparable accuracy while reducing GSI's estimated computation by 14%-39% across candidate budgets for Qwen2.5-Math and by up to 45% at $n = 16$ for Qwen3. The exact early exit applies whenever candidate scores are bounded above; we evaluate it inside GSI.

1 INTRODUCTION

Large language models (LLMs) are usually trained to predict the next token rather than to directly optimize human preferences, and may therefore generate responses that are inconsistent with the intended behavior (Bender et al., 2021; Bommasani et al., 2021). *Alignment* methods seek to modify the model distribution so that high-quality responses receive more probability while the aligned distribution remains close to the reference model, either during training, as in Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022), SLiC (Zhao et al., 2022), and Direct Preference Optimization (Rafailov et al., 2023), or at inference time without updating the model parameters, as in controlled decoding (Mudgal et al., 2024) and Best-of-n (BoN) sampling (Stiennon et al., 2020; Beirami et al., 2024).

BoN sampling is a simple inference-time alignment method that draws n candidates from a reference distribution P and returns the one with the largest reward. Prior work reports that BoN is empirically competitive with RLHF-style pipelines in terms of the *reward–KL divergence trade-off* (Mudgal et al., 2024), and it asymptotically approximates the solution of the KL-regularized reward optimization problem (Yang et al., 2024b). Its theoretical properties, coverage, and optimality have been studied extensively (Beirami et al., 2024; Gui et al., 2024; Mroueh & Nitsure, 2025; Amini et al., 2025; Huang et al., 2025; Sriraman & Block, 2026), and it has been further applied to different scenarios (Sun et al., 2024; Raman et al., 2025; Jinnai et al., 2025; Kalayci et al., 2025; Ichihara et al., 2025; Kang et al., 2025; Kobayashi, 2026; Mukherjee et al., 2026; Hsu et al., 2026).

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.

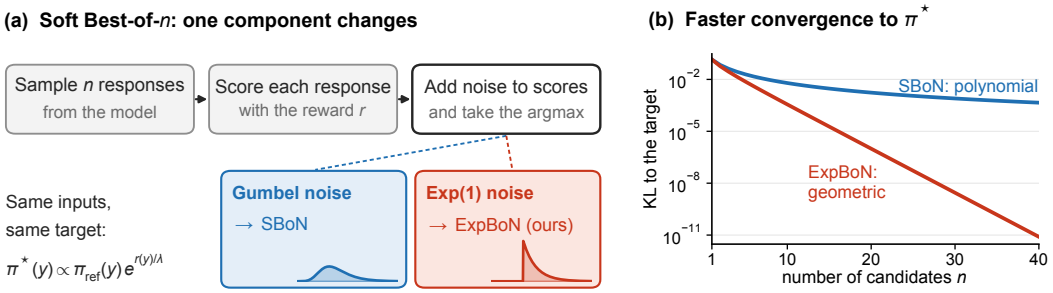


Figure 1: **ExpBoN replaces the Gumbel noise of soft Best-of-n with exponential noise and converges geometrically in n.** (a) Soft Best-of-n (SBoN) draws n responses from the reference model π_{ref} (P in the text), scores each with the reward r, adds independent noise to r/λ, and returns the response with the largest noisy score. Gumbel noise gives SBoN and $\text{Exp}(1)$ noise gives ExpBoN; both target the tilted distribution $\pi^\star$ ($P^\star_\lambda$). (b) $D_{\text{KL}}(\pi^\star \,\|\, \text{output})$ versus n (log scale), computed without sampling on the three-outcome example of Verdun et al. (2025) ($\lambda = 0.5$).

However, standard BoN, which we refer to as *hard* BoN, has two limitations. First, the number of candidates n provides only coarse control: increasing n simultaneously increases reward optimization and distribution shift. Second, hard maximization can exploit errors in a learned proxy reward model. As the candidate pool becomes larger, the selected response may have a high proxy reward but a lower true reward, known as reward hacking or overoptimization (Gao et al., 2023; Khalaf et al., 2025; Bu et al., 2025; Aminian et al., 2026).

Soft Best-of-n (SBoN) was introduced to provide finer control over this trade-off (Verdun et al., 2025). Given samples $X_1, \ldots, X_n \overset{\text{iid}}{\sim} P$, SBoN returns X_i with probability proportional to $e^{r(X_i)/\lambda}$, and hence interpolates between sampling from P and hard BoN. For every temperature λ, the output distribution of SBoN converges to the *optimal* target distribution (Csiszár, 1975), the exponentially tilted distribution $P^\star_\lambda(x) \propto P(x)e^{r(x)/\lambda}$, as n grows. It was further analyzed by Aminian et al. (2026), who studied its KL divergence from the reference policy and its true-reward regret under proxy-reward misspecification. See also recent applications of SBoN to reward hacking (Khalaf et al., 2025), diffusion language models (Bu et al., 2026), and guided speculative inference (GSI) (Geuter et al., 2026). However, SBoN's guarantees decay only polynomially in n (Verdun et al., 2025).

In this paper, we introduce an alternative soft BoN mechanism, called **ExpBoN**. While SBoN is equivalent to report-noisy-max with Gumbel noise, ExpBoN replaces the Gumbel perturbation with **exponential noise**: it draws the same n candidates, adds independent $\text{Exp}(1)$ noise to the scaled rewards $r(X_i)/\lambda$, and returns the noisy argmax (Figure 1a). The memorylessness of the exponential distribution makes this small change consequential: at every finite n the output distribution contains an exact tilted component whose weight approaches one geometrically (Theorem 3.1). This single property has two consequences. Statistically, ExpBoN converges geometrically to $P^\star_\lambda$ in TV and KL; the exact finite-n output distributions show the separation from SBoN (Figure 1b). Computationally, it permits an exact early exit (Section 4) whenever candidate scores are bounded above, which saves expensive evaluations in settings such as GSI.

Exponential-noise report-noisy-max is a classical mechanism in differential privacy (McSherry & Talwar, 2007; Dwork & Roth, 2014), equivalent to permute-and-flip (McKenna & Sheldon, 2020; Ding et al., 2021), and has recently been used in LLM decoding (Zhao et al., 2025). SBoN is closely related to the Poisson functional representation (Li & El Gamal, 2018), whose connection to differential privacy has also been studied (Liu et al., 2024b; Flamich et al., 2026). Khalaf et al. (2025) also introduce Best-of-Poisson (BoP), which approximates the tilted target with a Poisson-distributed number of candidates and a hard maximum; ExpBoN instead keeps the budget fixed at n and characterizes its finite-n output distribution exactly, which enables the exact early exit. Truncated rejection sampling with a known envelope, used for alignment by Huang et al. (2025), has the same mixture form; ExpBoN needs no envelope and always returns one of its n candidates, yet contains such a sampler (Theorem 3.1). The Gumbel form of SBoN and the utility dominance

(Appendix A.3) follow from known results; the decomposition, its consequences, and the exact early exit are our contributions. Here, we analyze this mechanism for LLM alignment and integrate it into GSI.

In summary, our main contributions are as follows.

- **Exponential-noise soft BoN.** We introduce ExpBoN, a soft BoN mechanism that replaces the Gumbel noise of SBoN with exponential noise, and show that its output distribution decomposes exactly, at every finite n, into the tilted target plus a residual whose weight vanishes geometrically (Section 3, Theorem 3.1).
- **Statistical guarantees.** We derive the statistical consequences of this decomposition, in parallel with the analyses of SBoN by Verdun et al. (2025) and Aminian et al. (2026): geometric convergence in TV, expected reward, and both KL directions, reference-KL localization, and regret bounds under proxy-reward misspecification whose finite-n term decays geometrically rather than polynomially (Section 3).
- **Exact early exit and ExpGSI.** We show that the same decomposition yields an exact early exit for any score bounded above, and instantiate it in GSI as ExpGSI: across two model families and three benchmarks, it reduces GSI's macro-averaged estimated computation by 14–45% at comparable accuracy (Sections 4 and 5).

2 BoN with Exponential Noise

Let a reference distribution P over finite $\mathcal{X}$ denote the distribution of an LLM's responses to a prompt, and $r : \mathcal{X} \to \mathbb{R}$ denote a reward function that scores responses and is nonconstant on $\mathcal{X}_P := \{x \in \mathcal{X} : P(x) > 0\}$. The *alignment* problem seeks a distribution $P^\star$ that maximizes the expected reward $\mathbb{E}_{P^\star}[r(X)]$ over $\Delta(\mathcal{X})$ subject to $D_{\mathrm{KL}}(P^\star \| P) \le \epsilon$, remaining close to P. This is an information projection problem (Csiszár & Matus, 2003), whose solution is the exponentially tilted distribution (Csiszár, 1975) $P^\star_\lambda(x) := P(x)e^{r(x)/\lambda}/\mathbb{E}_P[e^{r(X)/\lambda}]$, which assigns greater weight to high-reward responses; the temperature $\lambda > 0$, determined by ϵ, trades off reward maximization against closeness to P (Verdun et al., 2025).

However, we generally have only *sampling access* to P (no analytical expression or normalizing constants) and *query access* to r (its value can be computed on drawn samples), which is not enough to evaluate $P^\star_\lambda$ directly. Accordingly, one simple yet effective test-time alignment scheme is BoN sampling (Stiennon et al., 2020), which draws n samples from P and selects the one with the highest reward. All three mechanisms in this paper share the same first two steps—draw $X_1, \dots, X_n$ i.i.d. from P and compute the rewards $r(X_1), \dots, r(X_n)$—and differ only in the selection step. BoN selects greedily: it returns $Y = X_K$ with $K = \arg\max_{k=1,\dots,n} r(X_k)$.

Verdun et al. (2025) proposed soft BoN (SBoN), which replaces the hard argmax by a softmax draw: it returns $Y = X_K$ with $\Pr(K = i) = e^{r(X_i)/\lambda} \big/ \sum_{j=1}^n e^{r(X_j)/\lambda}$. SBoN recovers BoN as $\lambda \to 0$, and this softmax draw is equivalent (Appendix A.7) to report-noisy-max with Gumbel noise (Gumbel, 1954; Huijben et al., 2022; Li & El Gamal, 2018), $Y = X_K$ with $K = \arg\max_k\{r(X_k)/\lambda + G_k\}$, $G_k \overset{\text{iid}}{\sim} \mathrm{Gumbel}(0, 1)$.

ExpBoN instead uses exponential noise:

Exponential-Noise Best-of-n (ExpBoN)

1. Draw $X_1, \dots, X_n$ i.i.d. from P and compute the rewards $r(X_1), \dots, r(X_n)$;
2. Draw $E_1, \dots, E_n$ i.i.d. from $\mathrm{Exp}(1)$, independently;
3. Return $Y = X_K$, where $K = \arg\max_{k=1,\dots,n}\{r(X_k)/\lambda + E_k\}$.

ExpBoN has the same limiting behavior as SBoN: when $n = 1$ or $\lambda \to \infty$, the output follows P; when $\lambda \downarrow 0$, it approaches hard BoN; and, for every fixed $\lambda > 0$, it converges to the tilted target $P^\star_\lambda$ as $n \to \infty$. The difference is that exponential noise creates an exact tilted component at every finite n (Theorem 3.1). Table 2 in the appendix compares SBoN, Best-of-Poisson, and ExpBoN; Sections 3 and 4 develop the statistical and computational consequences.

3 THEORETICAL GUARANTEES

We provide theoretical analyses of ExpBoN, and compare to BoN and SBoN (Verdun et al., 2025) baselines. We derive an exact finite-n decomposition representation of ExpBoN, which shows geometric convergence to the optimal distribution, in contrast to the known $O(1/n)$ guarantees for SBoN.

3.1 EXACT DECOMPOSITION AND CONVERGENCE

For $k = 1, \dots, n$ and $E_k \sim \mathrm{Exp}(1)$, we denote $T_k := r(X_k)/\lambda + E_k$ and $M_n := \max_{k \le n} T_k$.

The following exact decomposition underlies most subsequent results.

Theorem 3.1. *For every $n \ge 1, \lambda > 0$, there exists a distribution $Q_{n,\lambda}$ on $\mathcal{X}_P$ such that the ExpBoN output distribution*

$$\widetilde{P}_{n,\lambda} = (1 - \rho_\lambda^n)\, P_\lambda^\star + \rho_\lambda^n Q_{n,\lambda},$$

where $\rho_\lambda := 1 - \mathbb{E}_P\left[e^{(r(X) - r_{\max})/\lambda}\right]$, $r_{\max} := \max_x r(x)$.

In comparison, SBoN only admits finite-sample approximation bounds (Verdun et al., 2025, Lemma 1). We call $\alpha_{n,\lambda} := 1 - \rho_\lambda^n$ the *tilted-hit probability*. Here $s_{\max} := r_{\max}/\lambda$ is the deterministic ceiling of the noisy scores T_k; conditional on the coupling event $\{M_n \ge s_{\max}\}$, the ExpBoN output distribution is exactly $P_\lambda^\star$. The mechanism is a one-line calculation: for a single draw, $\Pr\left(X = x,\ r(x)/\lambda + E \ge s_{\max}\right) = P(x)\, e^{-(s_{\max} - r(x)/\lambda)} \propto P(x) e^{r(x)/\lambda}$ by the exponential tail, so any candidate crossing the fixed threshold $s_{\max}$ is already an exact draw from $P_\lambda^\star$; the rejection-sampling and early-exit constructions of Section 4 run on the same calculation. Unlike truncated rejection sampling (Huang et al., 2025), ExpBoN does not need $r_{\max}$, which enters only the analysis. For a fixed temperature λ, increasing n increases the weight of the exact tilted component.

The exact decomposition leads to tight TV and KL guarantees for ExpBoN. Unlike Verdun et al. (2025), we provide results for both the forward and reverse KL divergences.

Theorem 3.2. *For every $n \ge 1$ and $\lambda > 0$, assuming r is not constant on $\mathcal{X}_P$, we have*

$$\begin{aligned}
q_\lambda \rho_\lambda^n &\le \mathrm{TV}\big(P_\lambda^\star, \widetilde{P}_{n,\lambda}\big) \le \rho_\lambda^n, \\
2 q_\lambda^2 \rho_\lambda^{2n} &\le D_{\mathrm{KL}}\big(\widetilde{P}_{n,\lambda} \| P_\lambda^\star\big) \le \log\big(1 + (p_{\min,\lambda}^{-1} - 1)\rho_\lambda^{2n}\big), \\
2 q_\lambda^2 \rho_\lambda^{2n} &\le D_{\mathrm{KL}}\big(P_\lambda^\star \| \widetilde{P}_{n,\lambda}\big) \le \min\big\{ -\log \alpha_{n,\lambda},\ (p_{\min,\lambda}^{-1} - 1)\rho_\lambda^{2n}/\alpha_{n,\lambda}\big\},
\end{aligned}$$

where $q_\lambda := P_\lambda^\star(\{x : r(x) = r_{\max}\})$ and $p_{\min,\lambda} := \min_{x \in \mathcal{X}_P} P_\lambda^\star(x)$. Consequently, as $n \to \infty$, $\mathrm{TV}(P_\lambda^\star, \widetilde{P}_{n,\lambda}) = \Theta(\rho_\lambda^n)$ and both KL divergences are $\Theta(\rho_\lambda^{2n})$.

Compared to SBoN, which only has $D_{\mathrm{KL}}(P_\lambda^\star \| \widetilde{P}^{\mathrm{S}}_{n,\lambda}) = O(1/n)$ (Verdun et al., 2025), where $\widetilde{P}^{\mathrm{S}}_{n,\lambda}$ denotes the SBoN output distribution, for every finite (P, r, λ), ExpBoN improves this rate to a geometric one. Here $1/(1 - \rho_\lambda) = \max_x P_\lambda^\star(x)/P(x)$, the envelope constant of rejection sampling and the coverage coefficient of Huang et al. (2025); $n \ge \max_x \frac{P_\lambda^\star(x)}{P(x)} \log \frac{1}{\epsilon}$ suffices for $\mathrm{TV} \le \epsilon$, so the geometric rate takes effect once n exceeds this coverage. On real pools it is close to one for the reward score (median 1.01) but heavy-tailed for the clipped GSI score (median 4.9, 90th percentile 421; Table 6).

3.2 REWARD AND KL CONVERGENCE

We then study how finite-n ExpBoN approaches the tilted target in expected reward and in the KL-regularized objective. We provide a one-sided reward guarantee and show that the expected reward improves monotonically in n.

Theorem 3.3 (Reward convergence)**.** *For $n \ge 1$ and $\lambda > 0$, $0 \le \mathbb{E}_{P_\lambda^\star}[r(X)] - \mathbb{E}_{\widetilde{P}_{n,\lambda}}[r(X)] \le \Delta_r \rho_\lambda^n$, where $\Delta_r := r_{\max} - r_{\min}$. For every $\lambda > 0$, $\mathbb{E}_{\widetilde{P}_{n,\lambda}}[r(X)]$ is nondecreasing in n, and converges to $\mathbb{E}_{P_\lambda^\star}[r(X)]$ as $n \to \infty$.*

With the centered reward $\overline{r}(x) := r(x) - r_{\min}$, we have the following.

Corollary 3.4. *For every $n \geq 1$ and $\lambda > 0$, $0 \leq 1 - \mathbb{E}_{\widetilde{P}_{n,\lambda}}[\overline{r}(X)]/\mathbb{E}_{P^\star_\lambda}[\overline{r}(X)] \leq \rho^n_\lambda$.*

Compared to Verdun et al. (2025, Theorem 3), which provided an $O(1/n)$ relative-reward guarantee for SBoN, Corollary 3.4 gives a geometric $O(\rho^n_\lambda)$ guarantee. The polynomial rate is not an artifact of loose upper bounds: Appendix A.8 exhibits a two-point instance on which SBoN's TV distance to the target is $\Omega(1/n)$ for every fixed temperature, while ExpBoN's decays geometrically.

The following result shows that the policy-to-reference KL of finite-n ExpBoN approaches that of $P^\star_\lambda$.

Corollary 3.5. *For every $n \geq 1$ and $\lambda > 0$,*

$$D_{\mathrm{KL}}(P^\star_\lambda \| P) - \tfrac{\Delta_r}{\lambda}\rho^n_\lambda \leq D_{\mathrm{KL}}(\widetilde{P}_{n,\lambda} \| P) \leq D_{\mathrm{KL}}(P^\star_\lambda \| P) + \log\left(1 + (p^{-1}_{\min,\lambda} - 1)\rho^{2n}_\lambda\right).$$

This shows that ExpBoN approaches the reward and reference-KL coordinates of the optimal tilted policy geometrically; we can also derive a distribution-free form $D_{\mathrm{KL}}(\widetilde{P}_{n,\lambda} \| P) \leq \min\{\log n, \Delta_r^2/(2\lambda^2)\}$, which has the correct quadratic order near the reference regime, whereas Aminian et al. (2026, Lemma 4.1) is linear in the inverse temperature. Through its equivalence with permute-and-flip, ExpBoN also attains at least the expected reward of SBoN on every candidate set (Theorem A.2 in Appendix A.3).

3.3 Regret Analysis

While the preceding results characterize how accurately finite-n ExpBoN approaches a prescribed distribution, this target may be defined using an imperfect proxy reward. Faster convergence does not by itself guarantee better true-reward performance and may amplify reward over-optimization. We therefore study the true-reward regret of ExpBoN, in parallel with the SBoN analysis of Aminian et al. (2026).

Suppose the proxy reward and the true reward are r_p and r_t, respectively. For $u \in \{p, t\}$, let $\widetilde{P}^u_{n,\lambda}$ be the ExpBoN output distribution when r_u is used, and define $P^\star_{\lambda,u}(x) := P(x)e^{r_u(x)/\lambda}/\mathbb{E}_P[e^{r_u(X)/\lambda}]$. Let $d(x) := r_t(x) - r_p(x)$ and define the centered reward-error quantity $\omega_d := \max_{x\in\mathcal{X}_P} d(x) - \min_{x\in\mathcal{X}_P} d(x)$.

Theorem 3.6 (Proxy-Reward Stability)**.** *The ExpBoN policies induced by the true and proxy rewards satisfy*

$$\max\left\{D_{\mathrm{KL}}(\widetilde{P}^t_{n,\lambda} \| \widetilde{P}^p_{n,\lambda}),\, D_{\mathrm{KL}}(\widetilde{P}^p_{n,\lambda} \| \widetilde{P}^t_{n,\lambda})\right\} \leq \min\left\{\tfrac{(n-1)\operatorname{Var}_P(d(X))}{\lambda^2},\, \tfrac{\omega_d}{\lambda}\tanh\left(\tfrac{\omega_d}{2\lambda}\right)\right\},$$

$$\mathrm{TV}\left(\widetilde{P}^t_{n,\lambda}, \widetilde{P}^p_{n,\lambda}\right) \leq \min\left\{\sqrt{\tfrac{(n-1)\operatorname{Var}_P(d(X))}{2\lambda^2}},\, \tanh\left(\tfrac{\omega_d}{2\lambda}\right),\, 1\right\}.$$

In comparison, Aminian et al. (2026, Lemma 4.2) analyzed the true-versus-proxy SBoN policy using an uncentered exponential mean-square error. Theorem 3.6 can be sharper for small centered errors, though there is no uniform dominance.

We then provide guarantees for the regret of ExpBoN. We first define $\Delta_t := \max_{x\in\mathcal{X}_P} r_t(x) - \min_{x\in\mathcal{X}_P} r_t(x)$ and $\mathrm{Reg}_{n,\lambda} := r_{t,\max} - \mathbb{E}_{\widetilde{P}^p_{n,\lambda}}[r_t(X)]$.

Theorem 3.7 (Direct Regret Bound)**.** *For every $n \geq 1$ and $\lambda > 0$,*

$$\mathrm{Reg}_{n,\lambda} \leq B_t(\lambda) + \Delta_t\left[\tanh\left(\frac{\omega_d}{4\lambda}\right) + \rho^n_{\lambda,p}\right],$$

where $B_t(\lambda) := r_{t,\max} - \mathbb{E}_{P^\star_{\lambda,t}}[r_t(X)]$ is the softening bias of the true tilted policy and $\rho_{\lambda,u} := 1 - \mathbb{E}_P\left[e^{(r_u(X) - r_{u,\max})/\lambda}\right]$ is the tilted-hit failure probability associated with reward r_u.

Theorem 3.7 separates the total regret into softening bias, proxy-target mismatch, and a finite-n target-approximation term. The last term is advantageous when the proxy-tilted target is trustworthy; under severe proxy misspecification, faster target approximation need not imply better true reward.

Aminian et al. (2026) studied the special scenario where $0 \leq r_t(x), r_p(x) \leq R_{\max}$ for all $x \in \mathcal{X}_P$. Here we define $\bar{\varepsilon}_\lambda := \inf_{c\in\mathbb{R}} \lambda \log \mathbb{E}_P\left[\exp\left((d(X) - c)^2/\lambda\right)\right]$, measuring the proxy-reward error:

Theorem 3.8 (Coverage-Based Regret Bound)**.** *For every $n \geq 1$ and $\lambda > 0$,*

$$\mathrm{Reg}_{n,\lambda} \leq \sqrt{\bar{\varepsilon}_\lambda}\big(\sqrt{C_{\infty,t}} + \sqrt{C_{\infty,p}}\big) + \lambda \log C_{\infty,t} + R_{\max}\big(1 - \tfrac{1}{C_{\infty,p}}\big)^n,$$

where $C_{\infty,u} := P(\{x \in \mathcal{X}_P : r_u(x) = r_{u,\max}\})^{-1}$.

Aminian et al. (2026, Theorem 5.2) gives the same reward-model-error and regularization-bias structure, but its displayed finite-n term is $O(n^{-1/2})$ for fixed coverage, whereas the corresponding ExpBoN term is $R_{\max}\left(1 - \frac{1}{C_{\infty,p}}\right)^n$. Therefore, making the finite-n term at most η needs $n = O\left(C_{\infty,p} \log \frac{R_{\max}}{\eta}\right)$ for ExpBoN, compared with $n = O\left(\frac{R_{\max}^2 (C_{\infty,p}-1)}{\eta^2}\right)$ for the SBoN (Aminian et al., 2026).

On the example of Verdun et al. (2025, Figure 1), the exact finite-n laws exhibit this separation (Figure 1b); Appendix A.10 compares them with the bounds above, also on real candidate pools.

4 ExpBoN with GSI

The exact early exit is a property of ExpBoN itself: it applies whenever candidates are scored by an expensive evaluator and the score is bounded above. We instantiate it in GSI (Geuter et al., 2026), where every score requires both a reward-model and a target-model evaluation.

GSI accelerates test-time alignment with SBoN by using a small draft model π_S to propose candidate reasoning steps and a large base (target) model π_B, together with a likelihood-ratio-corrected reward, to verify and select among them. At each reasoning state h, with draft model $\pi_S(\cdot \mid h)$ and base model $\pi_B(\cdot \mid h)$, the original GSI applies SBoN with inverse temperature β to a modified score $r(h,y) + d(h,y)/\beta$, i.e., it selects y_i with probability proportional to $e^{\beta r(h,y_i) + d(h,y_i)}$, where $d(h,y) := \log \frac{\pi_B(y|h)}{\pi_S(y|h)}$ and $\beta > 0$ is a parameter that trades off maximizing the reward versus fidelity to π_B; its $n \to \infty$ limit is $\pi^\star_{\beta,B}(y \mid h) \propto \pi_B(y \mid h)\, e^{\beta r(h,y)}$. Steps below a threshold u are regenerated by π_B.

We replace the SBoN component by ExpBoN. The early exit needs an upper bound on the score, but the likelihood ratio $d(h,y)$ need not have a finite upper bound; we therefore use the *clipped* score $s_C(h,y) := \beta r(h,y) + \min\{d(h,y), C\}$ with a fixed $C > 0$, as in the BoN analysis of Sriraman & Block (2026). If $r(h,y) \leq R$, then $s_C(h,y) \leq U_C := \beta R + C$, the early-exit threshold.

Let $\pi^\star_{\beta,C}(y \mid h) \propto \pi_S(y \mid h)\, e^{s_C(h,y)}$. Applied to the proposal π_S and the score s_C with ceiling U_C, the argument of Theorem 3.1 gives the output distribution

$$\widetilde{\pi}_{n,\beta,C}(\cdot \mid h) = (1 - \rho_{\beta,C}(h)^n)\, \pi^\star_{\beta,C}(\cdot \mid h) + \rho_{\beta,C}(h)^n Q_{n,\beta,C}(\cdot \mid h)$$

where $\rho_{\beta,C}(h) := 1 - \mathbb{E}_{\pi_S}\left[e^{s_C(h,Y) - U_C}\right]$ and $Q_{n,\beta,C}(\cdot \mid h)$ is some residual distribution. By the calculation after Theorem 3.1, a candidate whose noisy score crosses U_C is an *exact* sample from $\pi^\star_{\beta,C}(\cdot \mid h)$; equivalently, draw $Y \sim \pi_S(\cdot \mid h)$ and accept it with probability $e^{s_C(h,Y) - U_C}$. Moreover, by exponential memorylessness, conditional on crossing the threshold, the overshoot $s_C(h,Y) + E - U_C$ is $\mathrm{Exp}(1)$ and is independent of Y; therefore, selecting the largest overshoot, as full ExpBoN does, and selecting the first accepted candidate in an independent random order induce the same output distribution on the event that some candidate crosses. Thus, clipped ExpBoN can be implemented as a truncated rejection sampler: return the first accepted proposal in an independent random order if any of the n proposals is accepted, and otherwise evaluate all candidates and return the exponential-noise maximizer. Algorithm 1 implements this with two batches of sizes b and $n - b$; Line 7 is GSI's threshold gate. Its Lines 1–6 return an exact finite-n clipped ExpBoN sample (Figure 2; proof in Appendix A.5). By contrast, under Gumbel noise the analogous fixed-threshold crossing probability $1 - \exp\left(-e^{s_i - U}\right)$ is not proportional to e^{s_i}, so the same first-crossing construction would not preserve the SBoN output distribution (Verdun et al., 2025).

Clipping changes the limiting target from $\pi^\star_{\beta,B}$ to $\pi^\star_{\beta,C}$; Theorem 4.1 separates this bias from the finite-candidate error.

Theorem 4.1. *Assume $\beta > 0$, $r(h,y) \leq R$, and $\pi_B(\cdot \mid h) \ll \pi_S(\cdot \mid h)$. For every $n \geq 1$,*

$$\mathrm{TV}\left(\widetilde{\pi}_{n,\beta,C}(\cdot \mid h), \pi^\star_{\beta,B}(\cdot \mid h)\right) \leq \rho_{\beta,C}(h)^n + \tau_C(h) \tag{1}$$

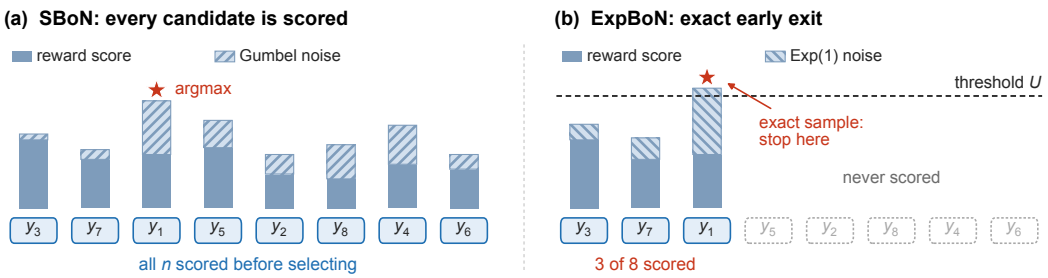


Figure 2: **With exponential noise, the sampler can stop before scoring all candidates.** Eight candidates in random order, the same in both panels; solid bars: reward scores; hatched: added noise; star: returned candidate. (a) SBoN scores all candidates before taking the argmax. (b) ExpBoN stops at the first candidate whose noisy score exceeds U, a known upper bound on the scores; that candidate is a sample from the tilted distribution, so the rest are not scored (3 of 8). If none exceeds U, all are scored and the noisy argmax is returned. Bar heights are illustrative.

Algorithm 1: ExpBoN-based GSI (ExpGSI), one reasoning step

Require: state h; policies π_S, π_B; reward model r; candidate count n; first-batch size b; inverse temperature β; clipping level C; reward bound R; GSI threshold u

1: Sample $y_1, \ldots, y_n \overset{\text{iid}}{\sim} \pi_S(\cdot \mid h)$, record $\ell_{S,i} \leftarrow \log \pi_S(y_i \mid h)$; draw an independent uniformly random permutation σ and $E_i \overset{\text{iid}}{\sim} \text{Exp}(1)$; set $U_C \leftarrow \beta R + C$
2: **for** $\mathcal{B} \in \{\sigma(1{:}b),\ \sigma(b{+}1{:}n)\}$ **do**
3: In parallel for $i \in \mathcal{B}$: $r_i \leftarrow r(h, y_i)$, $d_i \leftarrow \log \pi_B(y_i \mid h) - \ell_{S,i}$, $s_{C,i} \leftarrow \beta r_i + \min\{d_i, C\}$
4: $\mathcal{H} \leftarrow \{i \in \mathcal{B} : s_{C,i} + E_i \geq U_C\}$; **if** $\mathcal{H} \neq \varnothing$: $i^\star \leftarrow$ first of $\mathcal{H}$ in order σ, go to Line 7
5: **end for**
6: $i^\star \leftarrow \arg\max_i \{s_{C,i} + E_i\}$ ▷ no candidate passed
7: **if** $r_{i^\star} + d_{i^\star}/\beta \geq u$ **then return** $y_{i^\star}$; **else** run the base-model GSI fallback

where $\tau_C(h) := \mathbb{E}_{\pi^\star_{\beta,B}(\cdot|h)}\left[1 - \min\left\{1, e^{C - d(h,y)}\right\}\right]$. *Consequently, for every bounded* g,

$$\left|\mathbb{E}_{\widetilde{\pi}_{n,\beta,C}(\cdot|h)}[g(Y)] - \mathbb{E}_{\pi^\star_{\beta,B}(\cdot|h)}[g(Y)]\right| \leq \left(\sup_y g(y) - \inf_y g(y)\right)\left(\rho_{\beta,C}(h)^n + \tau_C(h)\right). \tag{2}$$

Theorem 4.1 separates the geometric finite-candidate error $\rho_{\beta,C}(h)^n$ from the nonvanishing clipping bias $\tau_C(h)$. Increasing n reduces only the former, while increasing C is required to reduce the distortion of the original GSI target. As for Theorem 1 of Geuter et al. (2026), the guarantee concerns the step selected before the threshold gate. In our experiments, C is set to the 95th percentile of d on held-out calibration pools, so that about 5% of calibration candidates are clipped (Appendix A.9). There, $\tau_C(h)$ is below 0.01 on half of the steps but reaches 0.57 at the 90th percentile (Table 6): the distortion is concentrated on a small fraction of steps, where the target strongly prefers a candidate the draft rarely proposes.

Expected verifier work. With the two-batch schedule $[b, n-b]$ of Algorithm 1, the expected number of candidates evaluated at state h is $b + \rho_{\beta,C}(h)^b\,(n-b)$; with $b = \max(1, \lfloor n/4 \rfloor)$, the scored fraction decreases toward $1/4$ as n grows. On the calibration pools this predicts 9.5 of 16 scored candidates (Table 6), close to the 9.2 measured on MATH500.

5 Experiments

Our experiments have two parts: Section 5.1 compares ExpBoN with SBoN on real candidate pools (Figure 3a,b), and Sections 5.2–5.3 compare ExpGSI with GSI and its baselines (Table 1, Figure 3c).

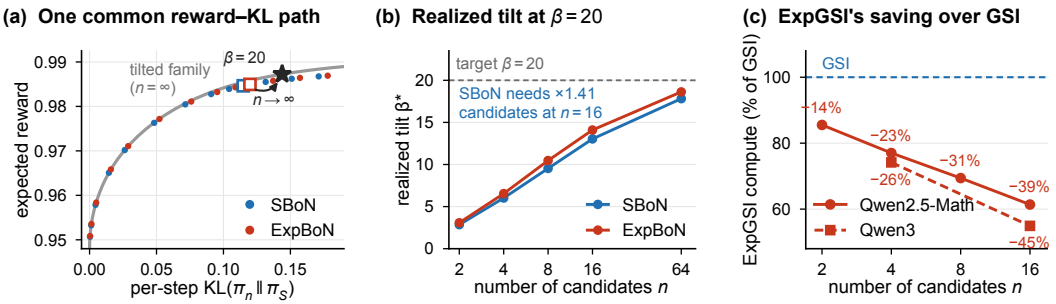


Figure 3: **ExpBoN reaches the target tilt with fewer candidates than SBoN, and ExpGSI costs less than GSI.** (a, b) Output distributions computed without sampling on 652 real candidate pools (64 Qwen2.5-Math draft candidates per reasoning step). (a) Expected reward versus per-step KL from the draft π_S at $n = 16$, one marker per β: both methods trace one path. Grey: tilted family ($n = \infty$); squares: $\beta = 20$; star: its $n \to \infty$ limit. (b) Realized tilt β^*, the β of the tilted distribution with equal KL, at $\beta = 20$. (c) ExpGSI's estimated compute relative to GSI (accuracy within 0.8 points).

Setup. We follow the evaluation protocol of GSI (Geuter et al., 2026). *Models:* the math-specialized Qwen2.5-Math pair (Yang et al., 2024a) (1.5B-INSTRUCT draft, 7B-INSTRUCT target) and the general-purpose Qwen3 pair (Yang et al., 2025) (1.7B draft, 14B target, thinking mode disabled), both scored by QWEN2.5-MATH-PRM-7B ($r \in [0, 1]$). *Benchmarks:* MATH500 (Lightman et al., 2024), MMLU-STEM (Hendrycks et al., 2020), and Minerva Math (Lewkowycz et al., 2022), on fixed 400-problem subsets (all 272 problems for Minerva), with identical problems and seeds across methods for paired comparison. *Baselines:* GSI, the method ExpGSI builds on, and the three baselines of its evaluation: S-BoN(π_S) and S-BoN(π_B), stepwise soft best-of-n over draft or target candidates (the draft-only and target-only references), and RSD (Liao et al., 2025), which gates between draft and target steps by a reward threshold, in GSI's shared-n instantiation. *Metrics:* final-answer accuracy, wall-clock time per reasoning step, acceptance (the fraction of steps resolved by a draft candidate), and estimated TFLOPs per problem under RSD's accounting. ExpGSI inherits all GSI hyperparameters; its only added sampling hyperparameter, the clipping level C, is calibrated once and never retuned (the first-batch size b affects only computation). Details: Appendix A.9.

5.1 EXPBON VERSUS SBON ON REAL CANDIDATE POOLS

ExpBoN and SBoN trace the same reward–KL path; ExpBoN moves farther along it with the same n. On 652 real candidate pools (Appendix A.10), ExpBoN's output at a fixed β lies farther from the reference than SBoN's, but on the same reward–KL path: sweeping the inverse temperature β at every budget n, the exact output distributions of the two attain rewards within 5×10^{-4} of each other at the same KL divergence (Figure 3a). The difference is therefore not a better trade-off but reaching the tilt set by β with fewer candidates: SBoN needs $1.41\times$ as many candidates at $n = 16$ (Figure 3b; Table 5).

5.2 MAIN RESULTS

ExpGSI's accuracy is comparable to GSI's. Across the four operating points of Table 1, macro accuracy differs from GSI by at most 0.8 points with no consistent sign, and acceptance rates agree within 0.3 points (Table 8). A per-problem paired bootstrap on the Qwen2.5-Math grid, pooled over benchmarks, gives ExpGSI−GSI differences of -0.09 and -0.62 points at $n = 4$ and 16, and every 95% confidence interval lies within $[-1.9, +1.4]$ points (Table 3). Only S-BoN(π_B), which generates every token with the target model, is more accurate (Table 8); on Qwen3, and on MMLU-STEM at small n, it is also cheaper in estimated TFLOPs, which charge a target-scored token as much as a target-generated one; the advantage of GSI and ExpGSI is latency, since draft candidates are scored in parallel, and ExpGSI is faster per step in every setting (Table 8).

Table 1: Macro-average results over MATH500, MMLU-STEM, and Minerva Math (mean $\pm$ s.d. over 3 seeds for Qwen2.5-Math, 2 for Qwen3; sizes: draft / target / PRM; time in seconds per step; estimated TFLOPs per problem). Bold: efficiency gains over GSI; †as implemented in GSI.

		Qwen2.5-Math (1.5B / 7B / 7B)			Qwen3 (1.7B / 14B / 7B)		
n	Method	Acc. (%)	Time (s)	TFLOPs	Acc. (%)	Time (s)	TFLOPs
4	RSD†	57.3 ± 1.2	0.58 ± 0.01	257 ± 7	55.8 ± 0.2	0.31 ± 0.00	541 ± 3
	GSI	59.9 ± 0.6	0.87 ± 0.01	370 ± 5	61.2 ± 0.3	0.60 ± 0.00	1202 ± 9
	ExpGSI (ours)	59.9 ± 0.8	$\mathbf{0.84 \pm 0.02}$	$\mathbf{285 \pm 4}$	60.4 ± 0.0	$\mathbf{0.55 \pm 0.00}$	$\mathbf{891 \pm 2}$
16	RSD†	58.5 ± 1.0	1.17 ± 0.03	1164 ± 37	56.9 ± 1.4	0.74 ± 0.00	2203 ± 19
	GSI	60.1 ± 0.5	1.71 ± 0.03	1627 ± 23	61.0 ± 0.1	1.50 ± 0.01	5769 ± 40
	ExpGSI (ours)	59.7 ± 0.8	$\mathbf{1.40 \pm 0.01}$	$\mathbf{999 \pm 24}$	61.8 ± 0.6	$\mathbf{1.05 \pm 0.00}$	$\mathbf{3169 \pm 6}$

ExpGSI costs less than GSI, and the margin grows with n. At every budget ExpGSI has lower estimated compute than GSI (Figures 3c and 4), saving 23% and 39% on Qwen2.5-Math at $n = 4$ and 16 and cutting time per step by 18% at n=16 (Table 1; all budgets in Table 3). The same holds for the general-purpose Qwen3 pair with the clipping level reused without retuning: at n=16 the saving is 45% in compute and 30% in time. RSD is cheaper at $n \leq 8$ and on Qwen3 and faster per step, at 1.2–4.9 points lower accuracy.

The early exit reduces both scoring costs. For Qwen2.5-Math at n=16, PRM scoring falls from 1176.0 to 760.5 TFLOPs per problem (66% of the reduction) and target-likelihood evaluation from 420.8 to 208.2 (34%), while draft generation is unchanged (Table 4): on MATH500 the early exit scores on average 9.2 of the 16 candidates per step (Table 14).

5.3 Ablation and Robustness Analysis

The conclusions hold with a smaller, general-purpose PRM. Replacing the 7B math-specialized PRM with Skywork-o1-Open-PRM-Qwen-2.5-1.5B (Team, 2024) (RSD's PRM) leaves the conclusions unchanged on MATH500 at $n \in \{4, 16\}$: accuracy and acceptance stay at GSI's level, ExpGSI still lowers GSI's estimated compute, and stays more accurate than RSD.

Accuracy is stable over a range of clipping levels. Sweeping the clipping level on MATH500 at n=16 keeps accuracy within seed-level noise from $0.5\widehat{C}$ to $1.5\widehat{C}$, with a mild drop only at $2\widehat{C}$, while estimated compute grows with C (Table 14).

Limitations. We evaluate the early exit end to end only inside GSI; for response-level Best-of-n we report an offline evaluation on saved responses of public reward models (Table 7). The saving depends on how tight the known score bound is: with a loose bound, almost all candidates are scored. ExpGSI's target differs from GSI's by a clipping bias that does not decrease with n (Theorem 4.1); it is small on most steps, and Section 5.3 examines its effect on accuracy.

6 Conclusion and Future Work

We introduced ExpBoN, an exponential-noise alternative to soft Best-of-n sampling for inference-time alignment. Its exact finite-n tilted decomposition yields geometric convergence, regret guarantees under proxy-reward misspecification, and an exact early exit. On real candidate pools, ExpBoN follows the same reward–KL path as SBoN while reaching a given tilt with fewer candidates. In GSI, the resulting ExpGSI reduces GSI's estimated computation by 14–45% at comparable accuracy.

Future directions include jointly performing alignment and watermarking through the connection to Permute-and-Flip (Zhao et al., 2025; Verma et al., 2026); combining ExpBoN with reward-hacking mitigation (Khalaf et al., 2025), since faster convergence may reach an overoptimized target sooner; and applying it to diffusion language models (Bu et al., 2026).

AI USE STATEMENT

We used large language models as coding and writing assistants: to implement and debug the experimental code, to design and run supplementary analyses of saved results, to help interpret results, and to edit the text, check reported numbers, and format references. They were not used to develop the theory, formulate mathematical claims, or write proofs; synthetic data, translation, and dataset cleaning do not apply to this work. The authors reviewed all AI-assisted content, regenerated every reported number from the saved records, and take full responsibility for the paper.

REPRODUCIBILITY STATEMENT

Section 5 and Appendix A.9 specify the models, benchmarks, hyperparameters, clipping-level calibration, and compute accounting; Appendix A.10 reports all results with the number of seeds. Every theorem states its assumptions, and all proofs are in the appendix. Anonymized code for the experiments and for regenerating the tables is provided in the supplementary material.

REFERENCES

Afra Amini, Tim Vieira, Elliott Ash, and Ryan Cotterell. Variational best-of-n alignment. In *International Conference on Learning Representations (ICLR)*, 2025.

Gholamali Aminian, Idan Shenfeld, Amir R Asadi, Ahmad Beirami, and Youssef Mroueh. Best-of-n through the smoothing lens: KL divergence and regret analysis. In *International Conference on Learning Representations*, 2026.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. *arXiv preprint arXiv:2401.01879*, 2024.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Dake Bu, Wei Huang, Andi Han, Atsushi Nitanda, Bo Xue, Qingfu Zhang, Hau-San Wong, and Taiji Suzuki. Consistency is not always correct: Towards understanding the role of exploration in post-training reasoning. *arXiv e-prints*, pp. arXiv:2511.07368, 2025.

Dake Bu, Wei Huang, Andi Han, Hau-San Wong, Qingfu Zhang, Taiji Suzuki, and Atsushi Nitanda. Dprm: A plug-in token-ordering module for diffusion language models. *arXiv preprint arXiv:2604.24357*, 2026.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30, 2017.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

Imre Csiszár and Frantisek Matus. Information projections revisited. *IEEE Transactions on Information Theory*, 49(6):1474–1490, 2003.

Zeyu Ding, Daniel Kifer, Thomas Steinke, Yuxin Wang, Yingtai Xiao, and Danfeng Zhang. The permute-and-flip mechanism is identical to report-noisy-max with exponential noise. *arXiv preprint arXiv:2105.07260*, 2021.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Gergely Flamich, Oykü Sıla Güner, Yanxiao Liu, and Deniz Gündüz. Scalable differentially private data compression via diffusion and stochastic codes. *arXiv preprint arXiv:2607.03392*, 2026.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios N. Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. How to evaluate reward models for rlhf. In *International Conference on Learning Representations*, 2025.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 10835–10866, 2023.

Jonathan Geuter, Youssef Mroueh, and David Alvarez-Melis. Guided speculative inference for efficient test-time alignment of LLMs. In *International Conference on Learning Representations*, volume 2026, pp. 75016–75056, 2026.

Lin Gui, Cristina Gârbacea, and Victor Veitch. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *Advances in Neural Information Processing Systems*, 37: 2851–2885, 2024.

Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hsiang Hsu, Eric Lei, and Chun-Fu Chen. Best-of-tails: Bridging optimism and pessimism in inference-time alignment. *arXiv preprint arXiv:2603.06797*, 2026.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. *arXiv preprint arXiv:2503.21878*, 2025.

Iris AM Huijben, Wouter Kool, Max B Paulus, and Ruud JG Van Sloun. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1353–1371, 2022.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Kenshi Abe, Kaito Ariu, Mitsuki Sakamoto, and Eiji Uchibe. Evaluation of best-of-n sampling strategies for language model alignment. *Transactions on Machine Learning Research*, 2025.

Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, and Kenshi Abe. Regularized best-of-n sampling with minimum bayes risk objective for language model alignment. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9321–9347, 2025.

Yusuf Kalayci, Vinod Raman, and Shaddin Dughmi. Optimal stopping vs best-of-n for inference time optimization. *arXiv preprint arXiv:2510.01394*, 2025.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *Advances in Neural Information Processing Systems*, 38:19720–19745, 2025.

Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio Calmon. Inference-time reward hacking in large language models. *Advances in Neural Information Processing Systems*, 38:61720–61760, 2025.

Taisuke Kobayashi. Flexible empowerment at reasoning with extended best-of-n sampling. *arXiv preprint arXiv:2604.15614*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Cheuk Ting Li and Abbas El Gamal. Strong functional representation lemma and applications to coding theorems. *IEEE Transactions on Information Theory*, 64(11):6967–6978, 2018.

Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. Reward-guided speculative decoding for efficient llm reasoning. *arXiv preprint arXiv:2501.19324*, 2025.

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 *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024a.

Yanxiao Liu, Wei-Ning Chen, Ayfer Özgür, and Cheuk Ting Li. Universal exact compression of differentially private mechanisms. *Advances in Neural Information Processing Systems*, 37:91492–91531, 2024b.

Ryan McKenna and Daniel R Sheldon. Permute-and-flip: A new mechanism for differentially private selection. *Advances in Neural Information Processing Systems*, 33:193–203, 2020.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103. IEEE, 2007.

Youssef Mroueh and Apoorva Nitsure. Information theoretic guarantees for policy alignment in large language models. *Transactions On Machine Learning Research*, 2025.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, Yaguang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 36486–36503. PMLR, 2024. URL `https://proceedings.mlr.press/v235/mudgal24a.html`.

Arpan Mukherjee, Marcello Bullo, Debabrota Basu, and Deniz Gündüz. Test-time verification via optimal transport: Coverage, ROC, & sub-optimality. In *International Conference on Learning Representations (ICLR)*, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Vinod Raman, Hilal Asi, and Satyen Kale. Adabon: Adaptive best-of-n alignment. *arXiv preprint arXiv:2505.12050*, 2025.

Ved Sriraman and Adam Block. Revisiting the (sub) optimality of best-of-n for inference-time alignment. *arXiv preprint arXiv:2603.05739*, 2026.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. Fast best-of-n decoding via speculative rejection. *Advances in Neural Information Processing Systems*, 37:32630–32652, 2024.

Skywork Team. Skywork-o1 open series. *https://huggingface. co/Skywork*, 2024.

Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio P Calmon. Soft best-of-n sampling for model alignment. In *2025 IEEE International Symposium on Information Theory (ISIT)*, pp. 1–6. IEEE, 2025.

Apurv Verma, NhatHai Phan, and Shubhendu Trivedi. Watermarking degrades alignment in language models: Analysis and mitigation. *Transactions on Machine Learning Research*, 2026.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10582–10592, 2024.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024a.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pp. 2027–2032. IEEE, 2024b.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. Permute-and-flip: An optimally stable and watermarkable decoder for LLMs. In *International Conference on Learning Representations, (ICLR)*, 2025.

Yao Zhao, Misha Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. Calibrating sequence likelihood improves conditional language generation. *arXiv preprint arXiv:2210.00045*, 2022.

A Appendices

A.1 Proofs for Section 3.1

We define

$$s(x) := \frac{r(x)}{\lambda}, \qquad s_{\max} := \frac{r_{\max}}{\lambda},$$

and, for $t \in \mathbb{R}$,

$$B_t := \{x \in \mathcal{X}_P : s(x) \le t\}.$$

Lemma A.1. *For almost every t with respect to the distribution of M_n, and every $x \in \mathcal{X}_P$,*

$$\Pr(Y = x \mid M_n = t) = \frac{P(x)e^{s(x)}\mathbf{1}\{x \in B_t\}}{\sum_{z\in\mathcal{X}_P} P(z)e^{s(z)}\mathbf{1}\{z \in B_t\}}.$$

Thus, conditional on $M_n = t$, the output follows $P^\star_\lambda$ restricted to B_t and renormalized. In particular, for almost every $t \ge s_{\max}$,

$$\Pr(Y = x \mid M_n = t) = P^\star_\lambda(x).$$

Proof. Let $T = s(X) + E$ and let F_T denote its distribution function. Since $E \sim \mathrm{Exp}(1)$, for every $x \in \mathcal{X}_P$,

$$\begin{aligned}\Pr(X = x, T \in dt) &= P(x)e^{-(t-s(x))}\mathbf{1}\{t \ge s(x)\}\, dt \\ &= e^{-t}P(x)e^{s(x)}\mathbf{1}\{t \ge s(x)\}\, dt.\end{aligned}$$

The noisy scores are independent and continuously distributed, so their maximum is attained at a unique index almost surely. By exchangeability,

$$\begin{aligned}\Pr(Y = x, M_n \in dt) = ne^{-t}P(x)e^{s(x)}&\mathbf{1}\{t \ge s(x)\} \\ &\times F_T(t)^{n-1}\, dt.\end{aligned}$$

For fixed t, all factors except $P(x)e^{s(x)}\mathbf{1}\{s(x) \le t\}$ are independent of x. Normalizing over $x \in \mathcal{X}_P$ proves the result. □

Proof of Theorem 3.1

Proof. For one noisy score,

$$\begin{aligned}\Pr(T < s_{\max}) &= \mathbb{E}_P\left[1 - e^{-(s_{\max}-s(X))}\right] \\ &= 1 - \mathbb{E}_P\left[e^{s(X)-s_{\max}}\right] \\ &= \rho_\lambda.\end{aligned}$$

Since the T_i are independent,

$$\Pr(M_n < s_{\max}) = \rho_\lambda^n.$$

If $\rho_\lambda = 0$, then $r(X) = r_{\max}$ P-almost surely, so $\widetilde{P}_{n,\lambda} = P^\star_\lambda = P$, and the claimed decomposition holds for any distribution $Q_{n,\lambda}$ on $\mathcal{X}_P$. Hence, for the remainder of the proof, suppose that $\rho_\lambda > 0$. By Lemma A.1, conditional on $M_n \ge s_{\max}$, the output follows $P^\star_\lambda$. Define

$$Q_{n,\lambda}(A) := \Pr(Y \in A \mid M_n < s_{\max}), \qquad A \subseteq \mathcal{X}_P.$$

Conditioning on $\{M_n \ge s_{\max}\}$ and $\{M_n < s_{\max}\}$ gives

$$\widetilde{P}_{n,\lambda} = (1 - \rho_\lambda^n)\, P^\star_\lambda + \rho_\lambda^n Q_{n,\lambda}.$$

We also record the property needed for the lower bounds below. Let

$$A_\lambda := \{x \in \mathcal{X}_P : r(x) = r_{\max}\}.$$

If $X_i \in A_\lambda$, then

$$T_i = s_{\max} + E_i \ge s_{\max}$$

almost surely. Hence, on $\{M_n < s_{\max}\}$, none of the candidates, and therefore not the selected output, belongs to A_λ. Thus,

$$Q_{n,\lambda}(A_\lambda) = 0.$$

□

PROOF OF THEOREM 3.2

Proof. Since r is nonconstant on $\mathcal{X}_P$, $\rho_\lambda \in (0,1)$, while $q_\lambda > 0$ and $p_{\min,\lambda} > 0$. Write

$$a := \rho_\lambda^n, \qquad P^\star := P_\lambda^\star, \qquad Q := Q_{n,\lambda}.$$

Theorem 3.1 gives

$$\widetilde{P}_{n,\lambda} = (1-a)P^\star + aQ.$$

Therefore,

$$\mathrm{TV}\big(P^\star, \widetilde{P}_{n,\lambda}\big) = a\,\mathrm{TV}(P^\star, Q) \le a.$$

Moreover, since $Q(A_\lambda) = 0$,

$$\begin{aligned} &P^\star(A_\lambda) - \widetilde{P}_{n,\lambda}(A_\lambda) \\ &\quad = q_\lambda - (1-a)q_\lambda = q_\lambda a. \end{aligned}$$

Taking the event A_λ in the variational definition of total variation yields

$$\mathrm{TV}\big(P^\star, \widetilde{P}_{n,\lambda}\big) \ge q_\lambda a.$$

Pinsker's inequality in either direction then gives

$$\begin{aligned} D_{\mathrm{KL}}\big(P^\star \| \widetilde{P}_{n,\lambda}\big) &\ge 2q_\lambda^2 a^2, \\ D_{\mathrm{KL}}\big(\widetilde{P}_{n,\lambda} \| P^\star\big) &\ge 2q_\lambda^2 a^2. \end{aligned}$$

For the first upper bound in the target-to-sampler direction,

$$\widetilde{P}_{n,\lambda}(x) \ge (1-a)P^\star(x),$$

and hence

$$D_{\mathrm{KL}}\big(P^\star \| \widetilde{P}_{n,\lambda}\big) \le -\log(1-a).$$

For the sharper finite-alphabet bounds, define

$$\chi^2(Q_1 \| Q_2) := \sum_x \frac{(Q_1(x) - Q_2(x))^2}{Q_2(x)}.$$

Using $D_{\mathrm{KL}}(Q_1 \| Q_2) \le \chi^2(Q_1 \| Q_2)$,

$$\begin{aligned} &D_{\mathrm{KL}}\big(P^\star \| \widetilde{P}_{n,\lambda}\big) \\ &\quad \le \chi^2\big(P^\star \| \widetilde{P}_{n,\lambda}\big) \\ &\quad = \sum_x \frac{a^2(P^\star(x) - Q(x))^2}{(1-a)P^\star(x) + aQ(x)} \\ &\quad \le \frac{a^2}{1-a}\chi^2(Q \| P^\star). \end{aligned}$$

Furthermore,

$$\begin{aligned} \chi^2(Q \| P^\star) &= \sum_x \frac{Q(x)^2}{P^\star(x)} - 1 \\ &\le p_{\min,\lambda}^{-1} - 1. \end{aligned}$$

This proves the second target-to-sampler KL upper bound.

Finally,

$$\chi^2\big(\widetilde{P}_{n,\lambda} \| P^\star\big) = a^2 \chi^2(Q \| P^\star).$$

The inequality $D_{\mathrm{KL}}(Q_1 \| Q_2) \le \log(1 + \chi^2(Q_1 \| Q_2))$ therefore gives

$$D_{\mathrm{KL}}\big(\widetilde{P}_{n,\lambda} \| P^\star\big) \le \log\left(1 + \left(p_{\min,\lambda}^{-1} - 1\right) a^2\right).$$

Substituting $a = \rho_\lambda^n$ proves all finite-n bounds. Since the constants q_λ and $p_{\min,\lambda}$ are fixed and positive, the stated asymptotic orders follow. □

A.2 Proofs for Section 3.2

Proof of Theorem 3.3

Proof. By Lemma A.1, conditional on $M_n = t$, the selected output follows $P_\lambda^\star$ restricted to $\{x : s(x) \le t\}$. As t increases, only points with larger reward are added to the conditioning set. Hence the conditional expected reward is nondecreasing in t and is at most $\mathbb{E}_{P_\lambda^\star}[r(X)]$. Under the natural coupling $M_{n+1} = \max\{M_n, T_{n+1}\} \ge M_n$ almost surely. Since the preceding conditional expected reward is a nondecreasing function of t, $\mathbb{E}_{\widetilde{P}_{n,\lambda}}[r(X)]$ is nondecreasing in n and is upper bounded by the tilted-target reward.

By Theorem 3.1,

$$\begin{aligned}&\mathbb{E}_{P_\lambda^\star}[r(X)] - \mathbb{E}_{\widetilde{P}_{n,\lambda}}[r(X)] \\ &\quad = \rho_\lambda^n \left(\mathbb{E}_{P_\lambda^\star}[r(X)] - \mathbb{E}_{Q_{n,\lambda}}[r(X)]\right).\end{aligned}$$

The expression in parentheses lies in $[0, \Delta_r]$, proving the displayed bound. Since r is not constant, $\rho_\lambda \in (0,1)$, and convergence follows. □

Proof of Corollary 3.4

Proof. Since r is nonconstant and $P_\lambda^\star$ has full support on $\mathcal{X}_P$,

$$\mathbb{E}_{P_\lambda^\star}[\overline{r}(X)] > 0.$$

The exact decomposition gives

$$\begin{aligned}&\mathbb{E}_{P_\lambda^\star}[\overline{r}(X)] - \mathbb{E}_{\widetilde{P}_{n,\lambda}}[\overline{r}(X)] \\ &\quad = \rho_\lambda^n \left(\mathbb{E}_{P_\lambda^\star}[\overline{r}(X)] - \mathbb{E}_{Q_{n,\lambda}}[\overline{r}(X)]\right).\end{aligned}$$

The left-hand side is nonnegative by Theorem 3.3. Since $\overline{r} \ge 0$,

$$\mathbb{E}_{P_\lambda^\star}[\overline{r}(X)] - \mathbb{E}_{Q_{n,\lambda}}[\overline{r}(X)] \le \mathbb{E}_{P_\lambda^\star}[\overline{r}(X)].$$

Dividing by the positive denominator proves the result. □

Proof of Corollary 3.5

Proof. For every distribution Q on $\mathcal{X}_P$,

$$\begin{aligned}&D_{\mathrm{KL}}(Q\|P) - D_{\mathrm{KL}}(P_\lambda^\star\|P) \\ &\quad = D_{\mathrm{KL}}(Q\|P_\lambda^\star) + \frac{\mathbb{E}_Q[r(X)] - \mathbb{E}_{P_\lambda^\star}[r(X)]}{\lambda}.\end{aligned}$$

Setting $Q = \widetilde{P}_{n,\lambda}$, using the nonnegativity of KL and Theorem 3.3, gives

$$\begin{aligned}&D_{\mathrm{KL}}\big(\widetilde{P}_{n,\lambda}\|P\big) - D_{\mathrm{KL}}\big(P_\lambda^\star\|P\big) \\ &\quad \ge -\frac{\Delta_r}{\lambda}\rho_\lambda^n.\end{aligned}$$

The same reward theorem also gives

$$\mathbb{E}_{\widetilde{P}_{n,\lambda}}[r(X)] - \mathbb{E}_{P_\lambda^\star}[r(X)] \le 0.$$

Therefore, Theorem 3.2 yields

$$\begin{aligned}&D_{\mathrm{KL}}\big(\widetilde{P}_{n,\lambda}\|P\big) - D_{\mathrm{KL}}\big(P_\lambda^\star\|P\big) \\ &\quad \le D_{\mathrm{KL}}\big(\widetilde{P}_{n,\lambda}\|P_\lambda^\star\big) \\ &\quad \le \log\left(1 + \left(p_{\min,\lambda}^{-1} - 1\right)\rho_\lambda^{2n}\right),\end{aligned}$$

which proves the localization bound. □

PROOF OF THE DISTRIBUTION-FREE REFERENCE-KL BOUND

Proof. Let

$$a_\lambda := \frac{\Delta_r}{\lambda}, \qquad Z(x) := \frac{\widetilde{P}_{n,\lambda}(x)}{P(x)}.$$

By exchangeability,

$$Z(x) = n \Pr\left(s(x) + E_1 \geq \max_{j=2,\dots,n} \{s(X_j) + E_j\} \right).$$

The winning probability of the fixed candidate is maximized by assigning it reward $r_{\max}$ and assigning every competitor reward $r_{\min}$. Hence

$$\begin{aligned} Z(x) &\leq n \int_0^\infty e^{-e} \left(1 - e^{-a_\lambda - e}\right)^{n-1} de \\ &= e^{a_\lambda} \left[1 - \left(1 - e^{-a_\lambda}\right)^n\right] =: u_{n,\lambda}. \end{aligned}$$

Similarly, the winning probability is minimized by assigning the fixed candidate reward $r_{\min}$ and every competitor reward $r_{\max}$, which gives

$$\begin{aligned} Z(x) &\geq n \int_{a_\lambda}^\infty e^{-e} \left(1 - e^{-(e - a_\lambda)}\right)^{n-1} de \\ &= e^{-a_\lambda}. \end{aligned}$$

Moreover, $\mathbb{E}_P[Z(X)] = 1$, $u_{n,\lambda} \leq n$, and $u_{n,\lambda} \leq e^{a_\lambda}$. Therefore,

$$D_{\mathrm{KL}}\big(\widetilde{P}_{n,\lambda} \| P\big) = \mathbb{E}_P[Z(X) \log Z(X)] \leq \log u_{n,\lambda} \leq \log n.$$

Also, $Z(X) \in [e^{-a_\lambda}, e^{a_\lambda}]$. Applying the chord bound for the convex function $z \mapsto z \log z$ under the constraint $\mathbb{E}_P[Z(X)] = 1$ gives

$$\mathbb{E}_P[Z(X) \log Z(X)] \leq a_\lambda \tanh\left(\frac{a_\lambda}{2}\right) \leq \frac{a_\lambda^2}{2}.$$

Combining the two bounds proves

$$D_{\mathrm{KL}}\big(\widetilde{P}_{n,\lambda} \| P\big) \leq \min\left\{ \log n, \frac{\Delta_r^2}{2\lambda^2} \right\}.$$

□

A.3 UTILITY DOMINANCE VIA PERMUTE-AND-FLIP

As discussed previously, report-noisy-max with exponential noise is equivalent to permute-and-flip (McKenna & Sheldon, 2020; Ding et al., 2021), which enjoys desirable properties. Here, we utilize them to establish the following result.

Theorem A.2 (Utility Dominance)**.** *For every $n \geq 1$, $\lambda > 0$, and candidates $x_{1:n} = (x_1, \dots, x_n)$, let K_{E} and K_{S} denote the indices selected by ExpBoN and SBoN, respectively. Then $\mathbb{E}[r(X_{K_{\mathrm{E}}}) \mid X_{1:n} = x_{1:n}] \geq \mathbb{E}[r(X_{K_{\mathrm{S}}}) \mid X_{1:n} = x_{1:n}]$, and consequently $\mathbb{E}_{\widetilde{P}_{n,\lambda}}[r(X)] \geq \mathbb{E}_{\widetilde{P}^{\mathrm{S}}_{n,\lambda}}[r(X)]$.*

ExpBoN inherits the *Pareto-optimality* of permute-and-flip (McKenna & Sheldon, 2020): within the equally stable candidate-level schemes, no one can uniformly improve the expected selection reward of ExpBoN over all reward vectors.

Proof. Conditional on $X_{1:n} = x_{1:n}$, ExpBoN is report-noisy-max with independent exponential noise and is therefore distributionally equivalent to Permute-and-Flip (Ding et al., 2021). Conditional SBoN is softmax sampling applied to the same candidate reward vector at the same temperature λ. The "never worse" property of Permute-and-Flip (Zhao et al., 2025, Theorem 3.1(3)), equivalently McKenna & Sheldon (2020, Theorem 2), therefore gives the conditional inequality. Averaging over $X_{1:n}$ gives the marginal result. □

A.4 Proofs for Section 3.3

We first record the mechanism-level stability result used below.

Lemma A.3. *For deterministic score vectors $u, v \in \mathbb{R}^n$, let q_u and q_v denote the distributions of*

$$\arg\max_{i=1,\dots,n}\{u_i + E_i\} \quad \textit{and} \quad \arg\max_{i=1,\dots,n}\{v_i + E_i\},$$

respectively, where the E_i are i.i.d. $\mathrm{Exp}(1)$. *Define*

$$D := \max_i (v_i - u_i) - \min_i (v_i - u_i).$$

Then, for every i,

$$e^{-D} q_u(i) \le q_v(i) \le e^{D} q_u(i).$$

Consequently, in either KL direction,

$$D_{\mathrm{KL}}(q_u \| q_v) \le D \tanh\left(\frac{D}{2}\right),$$

and

$$\mathrm{TV}(q_u, q_v) \le \tanh\left(\frac{D}{2}\right).$$

Proof. Fix i and subtract the common constant $v_i - u_i$ from all coordinates of v, which does not change q_v. The ith score then equals u_i, while each competitor score differs from the corresponding score under u by a quantity in $[-D, D]$. Monotonicity of the winning probability in each competitor score gives

$$q_{u - De_i}(i) \le q_v(i) \le q_{u + De_i}(i),$$

where e_i is the ith standard basis vector.

Let F_E be the distribution function of an $\mathrm{Exp}(1)$ variable. Then

$$\begin{aligned} q_{u+De_i}(i) &= \int_0^\infty e^{-e} \prod_{j \ne i} F_E(e + D + u_i - u_j)\, de \\ &= e^{D} \int_D^\infty e^{-t} \prod_{j \ne i} F_E(t + u_i - u_j)\, dt \\ &\le e^{D} q_u(i). \end{aligned}$$

Similarly,

$$\begin{aligned} q_{u-De_i}(i) &= e^{-D} \int_{-D}^\infty e^{-t} \prod_{j \ne i} F_E(t + u_i - u_j)\, dt \\ &\ge e^{-D} q_u(i). \end{aligned}$$

This proves the likelihood-ratio bound. The KL and total-variation bounds follow by applying the chord bounds to a likelihood ratio constrained to $[e^{-D}, e^{D}]$ and having mean one. □

Proof of Theorem 3.6

Proof. Condition on the candidate tuple $X_1, \dots, X_n$ and apply Lemma A.3 to

$$u_i = \frac{r_p(X_i)}{\lambda}, \qquad v_i = \frac{r_t(X_i)}{\lambda}.$$

The conditional score perturbation has oscillation at most ω_d/λ, giving the range-based conditional KL and TV bounds. Since the candidate marginal is P^n under both mechanisms, the KL chain rule gives

$$D_{\mathrm{KL}}(P^n q_t \| P^n q_p) = \mathbb{E}_{P^n}\left[D_{\mathrm{KL}}\big(q_t(\cdot \mid X_{1:n}) \| q_p(\cdot \mid X_{1:n})\big)\right],$$

and the analogous total-variation distance is the P^n-average of the conditional total-variation distances. The returned response is a deterministic function of this joint object, so data processing gives the range-based output-policy bounds.

For the variance-sensitive bound, Lemma A.3 and $\tanh(z) \le z$ give the conditional KL bound

$$\frac{1}{2\lambda^2}\left(\max_i d(X_i) - \min_i d(X_i)\right)^2.$$

Writing $d_i = d(X_i)$ and $\overline{d}_n = n^{-1}\sum_{i=1}^n d_i$,

$$\left(\max_i d_i - \min_i d_i\right)^2 \le 2\sum_{i=1}^n (d_i - \overline{d}_n)^2.$$

Taking expectations and using

$$\mathbb{E}\left[\sum_{i=1}^n (d_i - \overline{d}_n)^2\right] = (n-1)\operatorname{Var}_P(d(X))$$

proves the variance-sensitive KL bound in both directions. Pinsker's inequality gives the corresponding TV bound. □

PROOF OF THEOREM 3.7

Proof. The log-likelihood ratio between the true- and proxy-tilted targets satisfies

$$\log \frac{P^\star_{\lambda,t}(x)}{P^\star_{\lambda,p}(x)} = \frac{d(x)}{\lambda} + \text{constant},$$

and therefore has oscillation ω_d/λ. For completeness, if a likelihood ratio $L = dP/dQ$ satisfies $\operatorname{osc}(\log L) \le w$, then the chord bound for L under $\mathbb{E}_Q[L] = 1$, optimized over its possible endpoints, gives $\mathrm{TV}(P, Q) \le \tanh(w/4)$. The sharp likelihood-ratio range bound gives

$$\mathrm{TV}\big(P^\star_{\lambda,t}, P^\star_{\lambda,p}\big) \le \tanh\left(\frac{\omega_d}{4\lambda}\right).$$

The exact ExpBoN decomposition gives

$$\mathrm{TV}\big(P^\star_{\lambda,p}, \widetilde{P}^p_{n,\lambda}\big) \le \rho^n_{\lambda,p}.$$

Therefore,

$$\begin{aligned}\mathrm{Reg}_{n,\lambda} &= B_t(\lambda) + \mathbb{E}_{P^\star_{\lambda,t}}[r_t(X)] - \mathbb{E}_{\widetilde{P}^p_{n,\lambda}}[r_t(X)] \\ &\le B_t(\lambda) + \Delta_t \mathrm{TV}\big(P^\star_{\lambda,t}, \widetilde{P}^p_{n,\lambda}\big),\end{aligned}$$

and the triangle inequality proves the claim. □

PROOF OF THEOREM 3.8

Proof. For $u \in \{p, t\}$, write

$$D_u := D_{\mathrm{KL}}\big(P^\star_{\lambda,u} \| P\big),$$

and define

$$A_u := \{x \in \mathcal{X}_P : r_u(x) = r_{u,\max}\}, \qquad C_{\infty,u} := \frac{1}{P(A_u)}.$$

For any $c \in \mathbb{R}$, adding c to r_p changes neither $P^\star_{\lambda,p}$ nor $\widetilde{P}^p_{n,\lambda}$ and replaces d by $d - c$. The optimality of $P^\star_{\lambda,p}$ for the proxy KL-regularized objective gives

$$\mathbb{E}_{P^\star_{\lambda,t}}[r_p(X)] - \mathbb{E}_{P^\star_{\lambda,p}}[r_p(X)] \le \lambda(D_t - D_p).$$

Moreover, Cauchy–Schwarz gives

$$\begin{aligned}&\left|\mathbb{E}_{P^\star_{\lambda,u}}[d(X) - c]\right| \\ &\le \left(\sum_{x \in \mathcal{X}_P} \frac{P^\star_{\lambda,u}(x)^2}{P(x)}\right)^{1/2} \left(\mathbb{E}_P[(d(X) - c)^2]\right)^{1/2}.\end{aligned}$$

Since

$$\frac{P^\star_{\lambda,u}(x)}{P(x)} = \frac{e^{(r_u(x)-r_{u,\max})/\lambda}}{\mathbb{E}_P\left[e^{(r_u(X)-r_{u,\max})/\lambda}\right]} \le \frac{1}{P(A_u)} = C_{\infty,u},$$

we have

$$\sum_{x\in\mathcal{X}_P} \frac{P^\star_{\lambda,u}(x)^2}{P(x)} \le C_{\infty,u}.$$

Jensen's inequality further gives

$$\mathbb{E}_P[(d(X)-c)^2] \le \lambda \log \mathbb{E}_P\left[e^{(d(X)-c)^2/\lambda}\right].$$

For every $c \in \mathbb{R}$,

$$\begin{aligned}
&\mathbb{E}_{P^\star_{\lambda,t}}[r_t(X)] - \mathbb{E}_{P^\star_{\lambda,p}}[r_t(X)] \\
&= \mathbb{E}_{P^\star_{\lambda,t}}[r_p(X)] - \mathbb{E}_{P^\star_{\lambda,p}}[r_p(X)] \\
&\qquad + \mathbb{E}_{P^\star_{\lambda,t}}[d(X)-c] - \mathbb{E}_{P^\star_{\lambda,p}}[d(X)-c].
\end{aligned}$$

Combining this identity with the preceding bounds and the optimality inequality for the proxy objective, optimizing over c yields

$$\begin{aligned}
&\mathbb{E}_{P^\star_{\lambda,t}}[r_t(X)] - \mathbb{E}_{P^\star_{\lambda,p}}[r_t(X)] \\
&\qquad \le \lambda(D_t - D_p) + \sqrt{\bar{\varepsilon}_\lambda}\left(\sqrt{C_{\infty,t}} + \sqrt{C_{\infty,p}}\right).
\end{aligned}$$

Let $P^\star_{\infty,t} = P(\cdot \mid A_t)$. Since $\mathbb{E}_{P^\star_{\infty,t}}[r_t(X)] = r_{t,\max}$ and $D_{\mathrm{KL}}(P^\star_{\infty,t}\|P) = \log C_{\infty,t}$, comparing $P^\star_{\lambda,t}$ with $P^\star_{\infty,t}$ in the true KL-regularized objective gives

$$B_t(\lambda) \le \lambda\left(\log C_{\infty,t} - D_t\right).$$

Finally, the exact ExpBoN decomposition gives

$$\mathbb{E}_{P^\star_{\lambda,p}}[r_t(X)] - \mathbb{E}_{\widetilde{P}^p_{n,\lambda}}[r_t(X)] \le R_{\max}\rho^n_{\lambda,p},$$

and

$$\begin{aligned}
\rho_{\lambda,p} &= 1 - \mathbb{E}_P\left[e^{(r_p(X)-r_{p,\max})/\lambda}\right] \\
&\le 1 - P(A_p) = 1 - \frac{1}{C_{\infty,p}}.
\end{aligned}$$

Indeed,

$$\begin{aligned}
\mathrm{Reg}_{n,\lambda} = B_t(\lambda) &+ \left(\mathbb{E}_{P^\star_{\lambda,t}}[r_t(X)] - \mathbb{E}_{P^\star_{\lambda,p}}[r_t(X)]\right) \\
&+ \left(\mathbb{E}_{P^\star_{\lambda,p}}[r_t(X)] - \mathbb{E}_{\widetilde{P}^p_{n,\lambda}}[r_t(X)]\right).
\end{aligned}$$

Adding the softening-bias, proxy-mismatch, and finite-n terms cancels D_t. Dropping the nonpositive term $-\lambda D_p$ and applying the last bound on $\rho_{\lambda,p}$ proves the result. □

A.5 Proofs for Section 4

Fix a reasoning state h. Assume that $\pi_B(\cdot \mid h) \ll \pi_S(\cdot \mid h)$ and that $r(h,y) \le R$ for every relevant y. Define

$$Z_{\beta,B}(h) := \mathbb{E}_{\pi_B(\cdot|h)}\left[e^{\beta r(h,Y)}\right],$$

and

$$\pi^\star_{\beta,B}(y \mid h) := \frac{\pi_B(y \mid h)e^{\beta r(h,y)}}{Z_{\beta,B}(h)}.$$

Recall that

$$s_C(h,y) = \beta r(h,y) + \min\{d(h,y), C\}, \qquad U_C = \beta R + C.$$

PROOF OF THEOREM 4.1

Proof. Throughout this proof, we suppress the conditioning on the fixed state h when no ambiguity can arise. Define the clipped tilted target

$$\pi^{\star}_{\beta,C}(y \mid h) := \frac{\pi_S(y \mid h) e^{s_C(h,y)}}{Z_{\beta,C}(h)},$$

where

$$Z_{\beta,C}(h) := \mathbb{E}_{\pi_S(\cdot|h)}\left[e^{s_C(h,Y)}\right].$$

For one candidate $Y \sim \pi_S(\cdot \mid h)$ and $E \sim \mathrm{Exp}(1)$, let

$$H := \{s_C(h,Y) + E \geq U_C\}.$$

Since $s_C(h,y) \leq U_C$,

$$\Pr(Y = y, H \mid h) = e^{-U_C} \pi_S(y \mid h) e^{s_C(h,y)}.$$

Consequently,

$$\Pr(Y = y \mid H, h) = \pi^{\star}_{\beta,C}(y \mid h),$$

and the per-candidate no-hit probability is

$$\rho_{\beta,C}(h) = 1 - e^{-U_C} Z_{\beta,C}(h).$$

Conditional on H, exponential memorylessness gives

$$s_C(h,Y) + E - U_C \sim \mathrm{Exp}(1),$$

independently of Y. Hence, conditional on the set of hit indices, the hit labels are i.i.d. from $\pi^{\star}_{\beta,C}(\cdot \mid h)$ and are independent of their exponential overshoots. Selecting the largest overshoot, as full clipped ExpBoN does, or selecting the first hit in an independent random order, as Algorithm 1 does, therefore yields the same marginal hit-branch law $\pi^{\star}_{\beta,C}(\cdot \mid h)$. More explicitly, conditional on any nonempty hit set, both rules select an index uniformly from that set, independently of the i.i.d. hit labels. If no hit occurs, Algorithm 1 evaluates all candidates and returns the exponential-noise maximizer formed from the same clipped scores and noises. Thus, the index $i^{\star}$ selected by Lines 1–6 has exactly the finite-n clipped ExpBoN law; the subsequent GSI acceptance gate and fallback are not part of $\widetilde{\pi}_{n,\beta,C}$. Therefore,

$$\mathrm{TV}\left(\widetilde{\pi}_{n,\beta,C}, \pi^{\star}_{\beta,C}\right) \leq \rho_{\beta,C}(h)^n.$$

Define

$$w_C(h,y) := \min\left\{1, e^{C - d(h,y)}\right\}.$$

Then

$$\pi_S(y \mid h) e^{s_C(h,y)} = \pi_B(y \mid h) e^{\beta r(h,y)} w_C(h,y).$$

Writing

$$m_C(h) := \mathbb{E}_{\pi^{\star}_{\beta,B}(\cdot|h)}[w_C(h,Y)] = 1 - \tau_C(h),$$

we obtain

$$\pi^{\star}_{\beta,C}(y \mid h) = \frac{\pi^{\star}_{\beta,B}(y \mid h) w_C(h,y)}{m_C(h)}.$$

Since $0 \leq w_C \leq 1$ and $0 < m_C \leq 1$,

$$\begin{aligned}
&\mathrm{TV}\left(\pi^{\star}_{\beta,C}, \pi^{\star}_{\beta,B}\right) \\
&\quad = 1 - \mathbb{E}_{\pi^{\star}_{\beta,B}}\left[\min\left\{\frac{w_C}{m_C}, 1\right\}\right] \\
&\quad \leq 1 - \mathbb{E}_{\pi^{\star}_{\beta,B}}[w_C] = \tau_C(h).
\end{aligned}$$

The triangle inequality proves (1). The first bound uses only that a hit occurs with probability $1 - \rho_{\beta,C}(h)^n$ and returns a sample from $\pi^{\star}_{\beta,C}$; it therefore holds for any rule applied on the no-hit event, including the unclipped rule of our implementation (Appendix A.9). Finally,

$$|\mathbb{E}_Q[g] - \mathbb{E}_{Q'}[g]| \leq \left(\sup_y g(y) - \inf_y g(y)\right) \mathrm{TV}(Q, Q')$$

gives (2). □

A.6 Comparison of Soft Best-of-n Mechanisms

Table 2 compares ExpBoN with SBoN and with Best-of-Poisson (Khalaf et al., 2025).

Table 2: SBoN, Best-of-Poisson (BoP; Khalaf et al., 2025), and ExpBoN. The exact finite-n structure of ExpBoN (Theorem 3.1) yields both its geometric guarantees and its exact early exit.

	SBoN	BoP	ExpBoN (ours)
Candidates	fixed n	$1 + \mathrm{Poisson}(\mu)$	fixed n
Selection	argmax, Gumbel noise	hard argmax	argmax, $\mathrm{Exp}(1)$ noise
Relation to $P_\lambda^\star$	limit as $n \to \infty$	approximation via μ	limit as $n \to \infty$
Guarantees	$O(1/n)$	near-exact for uniform r	geometric (TV, reward, KL)
Scoring	all n candidates	all drawn candidates	exact early exit

A.7 Report-Noisy-Max with Gumbel Noise

We show that the final selection step of SBoN is equivalent in distribution to report-noisy-max with independent standard Gumbel noises: Fix $X_1, \dots, X_n$ and let $G_1, \dots, G_n \overset{\text{iid}}{\sim} \mathrm{Gumbel}(0,1)$, independently of the candidates. Define

$$K := \arg\max_{i=1,\dots,n} \left\{ \frac{r(X_i)}{\lambda} + G_i \right\}.$$

Then, for every $i \in \{1, \dots, n\}$,

$$\Pr\left(K = i \mid X_1, \dots, X_n\right) = \frac{e^{r(X_i)/\lambda}}{\sum_{j=1}^n e^{r(X_j)/\lambda}}. \tag{3}$$

Proof. Condition on $X_1, \dots, X_n$ and write

$$a_i := \frac{r(X_i)}{\lambda}, \qquad i = 1, \dots, n.$$

The cumulative distribution function and density of a standard Gumbel random variable are

$$F_G(g) = \exp\left(-e^{-g}\right)$$

and

$$f_G(g) = e^{-g} \exp\left(-e^{-g}\right),$$

respectively. Since the Gumbel distribution is continuous, the maximizer is unique almost surely. Hence,

$$\begin{aligned} &\Pr\left(K = i \mid X_1, \dots, X_n\right) \\ &\quad = \int_{-\infty}^{\infty} f_G(t - a_i) \prod_{j \neq i} F_G(t - a_j)\, dt. \end{aligned}$$

Using the expressions for f_G and F_G, the integrand becomes

$$\begin{aligned} &f_G(t - a_i) \prod_{j \neq i} F_G(t - a_j) \\ &= e^{-(t-a_i)} \exp\left(-e^{-(t-a_i)}\right) \prod_{j \neq i} \exp\left(-e^{-(t-a_j)}\right) \\ &= e^{a_i} e^{-t} \exp\left(-e^{-t} \sum_{j=1}^n e^{a_j}\right). \end{aligned}$$

Let

$$S := \sum_{j=1}^n e^{a_j}$$

and make the change of variables

$$u := Se^{-t}.$$

Then

$$du = -Se^{-t}\,dt,$$

and therefore

$$\begin{aligned}
&\Pr\left(K = i \mid X_1, \ldots, X_n\right) \\
&= e^{a_i} \int_{-\infty}^{\infty} e^{-t} \exp\left(-Se^{-t}\right) dt \\
&= \frac{e^{a_i}}{S} \int_{0}^{\infty} e^{-u}\,du \\
&= \frac{e^{a_i}}{\sum_{j=1}^{n} e^{a_j}}.
\end{aligned}$$

Substituting $a_j = r(X_j)/\lambda$ proves (3). □

A.8 A Polynomial Lower Bound for SBoN

The $O(1/n)$ upper bounds for SBoN are tight in rate: already on a two-point instance, SBoN's distance to the tilted target decays no faster than $1/n$, for every fixed temperature.

Proposition A.4. *Let $\mathcal{X} = \{a, b\}$, $P(b) = q \in (0,1)$, $r(a) = 0$, $r(b) = 1$, and write $w := e^{1/\lambda} > 1$. Let $\widetilde{P}^{\mathrm{S}}_{n,\lambda}$ denote the SBoN output law. Then, for every $n \geq 1$,*

$$\mathrm{TV}\big(P^{\star}_{\lambda}, \widetilde{P}^{\mathrm{S}}_{n,\lambda}\big) = P^{\star}_{\lambda}(b) - \widetilde{P}^{\mathrm{S}}_{n,\lambda}(b) \geq \frac{q(1-q)\,(w-1)}{w^2\,n}.$$

By contrast, Theorem 3.2 gives $\mathrm{TV}(P^{\star}_{\lambda}, \widetilde{P}_{n,\lambda}) \leq \rho_{\lambda}^{n}$ for ExpBoN on the same instance, with $\rho_{\lambda} = (1-q)\big(1 - e^{-1/\lambda}\big) < 1$.

Proof. Let $N \sim \mathrm{Bin}(n, q)$ count the draws equal to b. Conditional on the draws, the softmax selection returns b with probability $f(N)$, where $f(m) := \frac{mw}{mw+(n-m)} = \frac{mw}{n+m(w-1)}$, so $\widetilde{P}^{\mathrm{S}}_{n,\lambda}(b) = \mathbb{E}[f(N)]$. The map f is twice differentiable on $[0, n]$ with $f''(x) = -2wn(w-1)\big(n + x(w-1)\big)^{-3} < 0$, and $f(\mathbb{E}N) = f(nq) = \frac{qw}{1+q(w-1)} = P^{\star}_{\lambda}(b)$. A second-order Taylor expansion of f around nq with Lagrange remainder gives, pointwise for some ξ between N and nq, $f(N) = f(nq) + f'(nq)(N - nq) + \frac{1}{2} f''(\xi)(N - nq)^2$; taking expectations,

$$P^{\star}_{\lambda}(b) - \mathbb{E}[f(N)] = -\tfrac{1}{2}\,\mathbb{E}\big[f''(\xi)\,(N - nq)^2\big] \geq \tfrac{1}{2} \min_{x \in [0,n]} |f''(x)| \cdot \mathrm{Var}(N).$$

Since $|f''(x)| \geq 2wn(w-1)/(nw)^3 = 2(w-1)/(w^2 n^2)$ on $[0, n]$ and $\mathrm{Var}(N) = nq(1-q)$, the claim follows. The first equality in the proposition holds because the two laws differ only on the two atoms. □

A.9 More Implementation Details

Models and serving. The Qwen2.5-Math (Yang et al., 2024a) configuration uses Qwen2.5-Math-1.5B-Instruct as the draft policy π_S, Qwen2.5-Math-7B-Instruct as the target policy π_B, and Qwen2.5-Math-PRM-7B as the reward model; the Qwen3 (Yang et al., 2025) configuration uses Qwen3-1.7B and Qwen3-14B with the same PRM and thinking mode disabled. All models are served as separate vLLM (Kwon et al., 2023) instances with prefix caching enabled. Qwen2.5-Math runs use two NVIDIA A100-40 GPUs per evaluation run (draft and target colocated on one device, PRM on the other); Qwen3 runs host all three models on a single A100-80GB. Step timing excludes one-time engine initialization. Because the two families use different serving layouts, wall-clock numbers should be compared within, not across, model families. Draft- and target-policy log-probabilities are obtained through vLLM's prompt-logprobs interface with prefix caching enabled. Since prefix-cached positions return placeholder entries, we apply a thin runtime

patch that marks such rows; a marked row is filled from an already computed row with the same position and token within the same reasoning step, and otherwise the candidate is re-fetched with the cache bypassed (rows with NaN or positive values are rejected). GSI and ExpGSI use the same procedure; the patch does not alter vLLM's sampling or scheduling behavior and is provided in the supplementary material.

Benchmarks and evaluation. We evaluate on MATH500 (Lightman et al., 2024), MMLU-STEM (Hendrycks et al., 2020), and Minerva Math (Lewkowycz et al., 2022). For MATH500 and MMLU-STEM we use fixed 400-problem subsets (for MATH500, drawn from the pool that remains after removing the 60-problem calibration block); for Minerva Math we use the full 272-problem set. The same problems and random seeds are shared across all methods and candidate budgets, enabling paired comparisons. Results are reported as means and standard deviations over three random seeds for Qwen2.5-Math and two random seeds for Qwen3.

Inference configuration. Following GSI (Geuter et al., 2026), all methods use candidate budgets $n \in \{2, 4, 8, 16\}$ ($n \in \{4, 16\}$ for Qwen3), inverse temperature $\beta = 20$, acceptance threshold $u = 0.5$, sampling temperature 0.7, top-$p = 1.0$, and at most 512 new tokens per reasoning step. RSD (Liao et al., 2025) follows the implementation and hyperparameter configuration adopted by GSI, with acceptance threshold $\delta = 0.7$. The early-exit scan first evaluates a batch of $b = \max(1, \lfloor n/4 \rfloor)$ candidates (i.e., $b = 1, 1, 2, 4$ for $n = 2, 4, 8, 16$); if none passes the exit test, all remaining candidates are evaluated in a single second batch, which improves prefix-cache reuse and reduces scheduling overhead. The batch schedule is a serving-level choice fixed a priori and not tuned on any evaluation set: given the presampled candidate order, the algorithm always selects the first candidate that passes the exit test, so batching changes only the realized computation, never the selected step. When no candidate passes the exit test, our implementation returns the exponential-noise maximizer of the unclipped GSI score $\beta r + d$ rather than of s_C (Algorithm 1, Line 6). This changes only the residual component: the first-passer branch is unchanged, and the bound of Theorem 4.1 holds for any residual distribution (Appendix A.5), so it applies to the implemented sampler.

Calibration of the clipping level. The only hyperparameter ExpGSI adds over GSI is the clipping level C, which acts as a working upper bound on the per-step log-likelihood ratio $d = \log \pi_B(y \mid h) - \log \pi_S(y \mid h)$: the clipped score is bounded by $\beta + C$ by construction, and C only determines how often d is clipped and hence the clipping bias τ_C of Theorem 4.1. We calibrate C as a robust estimate of this upper bound. Rolling out 60 MATH500 problems held out from all evaluation subsets, with 64 draft candidates per reasoning step, we compute d for every candidate and set C to the empirical 95th percentile, yielding $C = 0.45$. A percentile is preferred over the sample maximum because d is heavy-tailed: the maximum would inflate the ceiling–and thus suppress early exits–for the sake of rare outliers, whereas the $\sim 5\%$ of candidates with $d > C$ are simply clipped–a change to the target distribution rather than a sampling error, bounded in Theorem 4.1–and, in practice, the selection rule is left nearly intact. Calibrated once, C is held fixed across all benchmarks, candidate budgets, and both model families without retuning; accuracy is insensitive to this choice over a wide range (Table 14).

Computational accounting. Estimated FLOPs follow the convention of RSD (Liao et al., 2025): each token processed by a model costs $2N$ FLOPs, where N is the model's parameter count, and we sum over the forward passes of the main selection loop. This comprises (i) generation, charged to the generating policy, including GSI's target-policy fallback regeneration; (ii) reward-model scoring, where each call re-reads the full sequence (prompt, accepted prefix, and candidate steps), consistent with the official RSD implementation, since no key–value cache persists across PRM calls; and (iii) the target-policy log-likelihoods used by GSI and ExpGSI, which are served with prefix caching and hence charged only for the candidate suffix, while cache cold starts and refetches are charged at full sequence length. Per-problem totals are averaged over the evaluation set. Two costs of the fallback taken on GSI-rejected steps are not included: the reward-model scoring of the fallback candidates (for both GSI and ExpGSI) and ExpGSI's fallback regeneration, which was not logged. Estimating the latter from GSI's fallback generations and adding both terms raises the absolute estimated TFLOPs of both methods and changes ExpGSI's saving relative to GSI by at most 1.1 points (e.g., from 38.6% to 38.1% at n=16 for Qwen2.5-Math and from 45.1% to 44.9% for Qwen3); the recount script is included in the supplementary material.

Computing infrastructure. Qwen2.5-Math experiments (including the reward-model and clipping-level ablations) ran on a server with 8× NVIDIA A100-SXM4-40GB GPUs, AMD EPYC 7542 CPUs (128 hardware threads), 2 TB RAM, and Linux, allocating two GPUs per evaluation run as described above. Qwen3 experiments ran on a cloud instance with 6× NVIDIA A100-SXM4-80GB GPUs and 2 TB RAM (Linux); for seed 0 each run used two GPUs (draft and target on one, the PRM on the other), and for seed 1 all three models of a run shared a single GPU. Both machines used the same software stack: Python 3.12, PyTorch 2.11 (CUDA 13), vLLM 0.24, and Transformers 5.13; a complete dependency freeze is provided in the supplementary material.

A.10 Additional experimental results

Numbers quoted in the main text. Table 3 reports GSI and ExpGSI at every budget, including the paired accuracy differences; Table 4 splits the n=16 compute saving by component; Table 5 summarizes the matched-KL comparison of Figure 3a,b; Table 6 reports coverage, clipping and early-exit diagnostics on the calibration pools.

Table 3: GSI and ExpGSI at every budget (macro average over the three benchmarks; 3 seeds for Qwen2.5-Math, 2 for Qwen3). Paired ΔAcc: ExpGSI−GSI in points, per-problem paired bootstrap over benchmarks and seeds (3,216 pairs per budget, 10^4 resamples), Qwen2.5-Math only; it pools problems, so each problem has equal weight, whereas the Acc. columns average the three benchmarks. Saving: reduction relative to GSI; source of Figure 3c.

		Acc. (%)		Paired ΔAcc	Time (s)		TFLOPs		Saving	
Model	n	GSI	ExpGSI	[95% CI]	GSI	ExpGSI	GSI	ExpGSI	Time	TFLOPs
Qwen2.5	2	59.1	59.0	−0.28 [−1.80, +1.24]	0.75	0.70	168	144	6%	14%
	4	59.9	59.9	−0.09 [−1.34, +1.15]	0.87	0.84	370	285	4%	23%
	8	60.0	60.2	+0.12 [−1.18, +1.40]	1.14	1.05	788	547	8%	31%
	16	60.1	59.7	−0.62 [−1.90, +0.62]	1.71	1.40	1627	999	18%	39%
Qwen3	4	61.2	60.4	–	0.60	0.55	1202	891	8%	26%
	16	61.0	61.8	–	1.50	1.05	5769	3169	30%	45%

Table 4: Estimated compute by component, Qwen2.5-Math, n=16 (macro average, 3 seeds). Totals reproduce Table 1.

Component (TFLOPs per problem)	GSI	ExpGSI	Share of saving
Draft generation	29.9	29.9	–
PRM scoring	1176.0	760.5	66%
Target likelihood (incl. fallback)	420.8	208.2	34%
Total	1626.7	998.6	

Table 5: Matched-KL comparison on the 652 real candidate pools of Figure 3a,b (exact finite-n output distributions). Reward gap: largest difference in expected reward between ExpBoN and SBoN at equal per-step KL over the swept β (expected rewards lie in $[0.95, 0.99]$). Realized tilt: the tilt whose KL matches the finite-n output distribution. Last column: number of SBoN candidates reaching ExpBoN's realized tilt, relative to n (> 64: not reached within the grid).

	Max. reward gap	Realized tilt at β=20		Candidates SBoN needs
n	at equal KL	SBoN	ExpBoN	(multiple of n)
2	3.6×10^{-4}	2.9	3.1	1.05×
4	4.7×10^{-4}	6.0	6.5	1.14×
8	2.9×10^{-4}	9.5	10.5	1.23×
16	1.8×10^{-4}	13.0	14.1	1.41×
64	6.3×10^{-5}	17.8	18.6	> 64

Table 6: Diagnostics on the 652 calibration pools ($\beta = 20$, $C = 0.45$, $R = 1$; empirical reference = uniform over the pool's draft candidates). Coverage: $1/(1-\rho) = \max_x P^\star(x)/P(x)$. Scored fraction: $(b + \rho^b(n-b))/n$ for the two-batch early exit at n=16; for the reward-only score, the ceiling is the known bound βR, as in response-level Best-of-n with a bounded reward model. The mean coverage of the clipped GSI score is dominated by pools whose rewards are all far below R.

Quantity	10th pct.	Median	90th pct.	Mean
Coverage, reward score βr (ceiling $\beta \max r$)	1.00	1.01	2.13	1.96
Coverage, clipped GSI score s_C (ceiling U_C)	1.64	4.93	421	–
Clipping bias τ_C	0.000	0.007	0.567	0.165
$\mathrm{TV}(\pi^\star_{\beta,C}, \pi^\star_{\beta,B})$	0.000	0.005	0.292	0.088
Fraction of candidates with $d > C$	0.000	0.016	0.170	0.050
Scored fraction at n=16, clipped GSI score	0.27	0.55	0.99	0.59
Scored fraction at n=16, reward-only score	0.25	0.25	0.33	0.30

Response-level Best-of-n with public reward models. To check the exact early exit outside GSI without new generation, we use the MATH, MMLU-Pro, and GPQA Best-of-K sets of PPE (Frick et al., 2025): 512 prompts each, 32 responses per prompt with correctness labels, scored by Skywork-Reward-Llama-3.1-8B (Liu et al., 2024a) and ArmoRM-Llama3-8B (Wang et al., 2024). Scores are mapped to $(0, 1)$ by a sigmoid fitted on half of the prompts, and results are reported on the other half. As in ExpGSI, the known bound R is the 95th percentile of the calibration scores and scores are clipped at R. For each prompt we draw n responses i.i.d. from its 32 responses (400 draws per prompt) and apply SBoN, ExpBoN, and ExpBoN with the two-batch early exit. Table 7 shows that the early exit returns the same accuracy as full ExpBoN (largest difference 0.44 points over all 144 settings, within Monte-Carlo error), that ExpBoN is at least as accurate as SBoN in 124 of 144 settings, and that the number of reward-model calls falls with n and rises with β, as the coverage analysis of Section 3 predicts; with the loose bound $R = 1$, most candidates must be scored.

Table 7: Response-level Best-of-n on PPE (mean over three benchmarks and two reward models; accuracy in %). Scored: fraction of the n responses whose reward is evaluated by the two-batch early exit (range over the six benchmark–reward-model pairs); last column: the same fraction with the loose bound $R = 1$.

		Accuracy			Scored fraction	
β	n	SBoN	ExpBoN	ExpBoN (early exit)	R: 95th pct.	$R = 1$
5	16	45.2	45.4	45.5	0.54 [0.48, 0.59]	0.72
5	32	45.3	45.5	45.5	0.42 [0.36, 0.45]	0.58
20	16	50.2	50.4	50.4	0.84 [0.81, 0.89]	0.98
20	32	50.5	50.6	50.6	0.79 [0.73, 0.85]	0.96

Table 8 extends Table 1 with the two S-BoN baselines. Figure 4 places all five methods on the compute–accuracy plane at each budget.

Results across candidate budgets (Qwen2.5-Math). Tables 9–11 report the complete per-benchmark results. The computational advantage of ExpGSI over GSI increases with the candidate budget on every benchmark, reaching 33–46% of estimated compute at n=16 with latency reductions of 16–23%. Acceptance rates remain close to GSI's throughout, and accuracy differences show no consistent direction. ExpGSI thus reduces the selection overhead of GSI without systematically changing its solution quality.

Per-benchmark results for Qwen3. Table 12 breaks the Qwen3 rows of Table 1 down by benchmark. The picture from the Qwen2.5-Math experiments transfers: accuracy and acceptance remain at GSI's level on every benchmark (the largest deviation, on Minerva Math, lies within its seed spread), while at n=16 estimated computation drops by 38–52% and time per step by 23–40%. The larger savings relative to Qwen2.5-Math are consistent with the larger target model: the avoided target-policy evaluations account for a larger share of total cost.

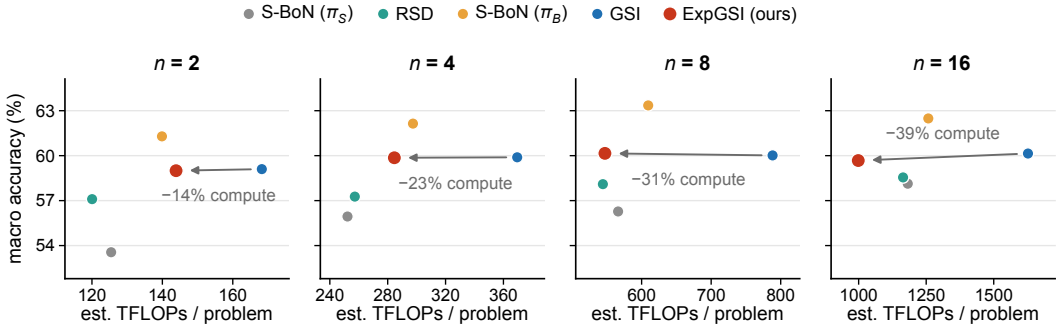


Figure 4: Per-budget view of Figure 3c: compute–accuracy planes per candidate budget n (Qwen2.5-Math; macro average over the three benchmarks, 3-seed means). Arrows mark the GSI $\rightarrow$ ExpGSI compute reduction, which widens from 14% at n=2 to 39% at n=16 at comparable accuracy.

Table 8: Full version of Table 1, including the S-BoN baselines. Macro-average results over MATH500, MMLU-STEM, and Minerva Math and multiple seeds. ExpGSI maintains accuracy and acceptance comparable to GSI while reducing time and computation. At $n = 16$ it reduces time and TFLOPs by 18% and 39% for Qwen2.5-Math, and by 30% and 45% for Qwen3. Bold values indicate ExpGSI efficiency improvements over GSI. †RSD follows the GSI implementation and hyperparameter configuration. Entries are mean $\pm$ standard deviation over seeds (3 for Qwen2.5-Math, 2 for Qwen3). ‡S-BoN(π_B) generates every token with the target policy: it is a quality reference at the highest per-step latency (1.84 s vs. ExpGSI's 1.40 s at Qwen2.5, n=16), not a comparable operating point; the direct comparison for ExpGSI is GSI.

Model family	n	Method	Acc. (%)	Time/step (s)	Accept. (%)	Est. TFLOPs/prob.
Qwen2.5-Math (1.5B / 7B / 7B)	4	S-BoN (π_S)	55.9 ± 0.5	0.49 ± 0.01	–	252 ± 6
		RSD†	57.3 ± 1.2	0.58 ± 0.01	95.7 ± 0.2	257 ± 7
		S-BoN (π_B)‡	62.1 ± 0.4	1.06 ± 0.01	–	297 ± 6
		GSI	59.9 ± 0.6	0.87 ± 0.01	82.1 ± 0.2	370 ± 5
		ExpGSI (ours)	59.9 ± 0.8	$\mathbf{0.84 \pm 0.02}$	81.9 ± 0.4	$\mathbf{285 \pm 4}$
	16	S-BoN (π_S)	58.1 ± 0.2	1.05 ± 0.01	–	1182 ± 16
		RSD†	58.5 ± 1.0	1.17 ± 0.03	97.3 ± 0.2	1164 ± 37
		S-BoN (π_B)‡	62.5 ± 0.2	1.84 ± 0.05	–	1258 ± 13
		GSI	60.1 ± 0.5	1.71 ± 0.03	88.8 ± 0.3	1627 ± 23
		ExpGSI (ours)	59.7 ± 0.8	$\mathbf{1.40 \pm 0.01}$	88.7 ± 0.4	$\mathbf{999 \pm 24}$
Qwen3 (1.7B / 14B / 7B)	4	S-BoN (π_S)	54.5 ± 0.5	0.28 ± 0.00	–	521 ± 1
		RSD†	55.8 ± 0.2	0.31 ± 0.00	97.7 ± 0.0	541 ± 3
		S-BoN (π_B)‡	66.0 ± 0.5	0.78 ± 0.00	–	642 ± 8
		GSI	61.2 ± 0.3	0.60 ± 0.00	89.4 ± 0.1	1202 ± 9
		ExpGSI (ours)	60.4 ± 0.0	$\mathbf{0.55 \pm 0.00}$	89.5 ± 0.0	$\mathbf{891 \pm 2}$
	16	S-BoN (π_S)	54.8 ± 0.1	0.71 ± 0.00	–	2183 ± 4
		RSD†	56.9 ± 1.4	0.74 ± 0.00	98.5 ± 0.0	2203 ± 19
		S-BoN (π_B)‡	65.9 ± 0.0	1.34 ± 0.00	–	2565 ± 34
		GSI	61.0 ± 0.1	1.50 ± 0.01	92.7 ± 0.1	5769 ± 40
		ExpGSI (ours)	61.8 ± 0.6	$\mathbf{1.05 \pm 0.00}$	93.0 ± 0.3	$\mathbf{3169 \pm 6}$

Table 9: Complete results on MATH500 across candidate budgets. Reported values are means and standard deviations over three random seeds. Bold efficiency values indicate improvements of ExpGSI over GSI. †RSD uses the implementation and hyperparameter configuration adopted by GSI.

n	Method	Acc. (%)	Time/step (s)	Accept. (%)	Est. TFLOPs/prob.
2	S-BoN (π_S)	76.5 ± 0.4	0.40 ± 0.00	–	121 ± 5
	RSD†	79.3 ± 1.9	0.49 ± 0.02	96.7 ± 0.3	118 ± 3
	S-BoN (π_B)	83.2 ± 1.1	0.91 ± 0.00	–	148 ± 1
	GSI	80.3 ± 1.2	0.66 ± 0.02	89.6 ± 0.5	174 ± 5
	ExpGSI (ours)	80.2 ± 0.7	0.61 ± 0.02	89.5 ± 0.8	$\mathbf{147 \pm 3}$
4	S-BoN (π_S)	78.1 ± 0.8	0.52 ± 0.03	–	244 ± 22
	RSD†	80.0 ± 1.1	0.59 ± 0.02	97.6 ± 0.1	247 ± 5
	S-BoN (π_B)	83.2 ± 0.8	1.11 ± 0.02	–	322 ± 6
	GSI	81.8 ± 1.4	0.78 ± 0.02	92.4 ± 0.4	381 ± 8
	ExpGSI (ours)	82.3 ± 1.7	0.74 ± 0.02	92.3 ± 0.7	$\mathbf{288 \pm 5}$
8	S-BoN (π_S)	80.0 ± 0.7	0.71 ± 0.03	–	532 ± 40
	RSD†	80.6 ± 1.2	0.77 ± 0.01	98.1 ± 0.6	511 ± 29
	S-BoN (π_B)	84.5 ± 0.5	1.44 ± 0.03	–	647 ± 30
	GSI	82.3 ± 1.2	1.01 ± 0.01	94.1 ± 0.2	835 ± 14
	ExpGSI (ours)	82.1 ± 0.6	$\mathbf{0.91 \pm 0.01}$	94.3 ± 0.1	$\mathbf{524 \pm 22}$
16	S-BoN (π_S)	80.4 ± 0.4	1.08 ± 0.02	–	1130 ± 14
	RSD†	80.2 ± 0.9	1.15 ± 0.05	98.3 ± 0.4	1046 ± 17
	S-BoN (π_B)	84.2 ± 1.4	1.96 ± 0.02	–	1366 ± 16
	GSI	82.2 ± 1.4	1.52 ± 0.06	95.4 ± 0.5	1679 ± 10
	ExpGSI (ours)	81.8 ± 0.9	$\mathbf{1.17 \pm 0.02}$	95.6 ± 0.3	$\mathbf{900 \pm 24}$

Table 10: Complete results on MMLU-STEM across candidate budgets. Reported values are means and standard deviations over three random seeds. Bold efficiency values indicate improvements of ExpGSI over GSI. †RSD uses the implementation and hyperparameter configuration adopted by GSI.

n	Method	Acc. (%)	Time/step (s)	Accept. (%)	Est. TFLOPs/prob.
2	S-BoN (π_S)	59.2 ± 2.1	0.35 ± 0.00	–	107 ± 0
	RSD†	67.1 ± 0.8	0.48 ± 0.01	89.4 ± 0.4	92 ± 2
	S-BoN (π_B)	68.8 ± 2.3	0.79 ± 0.01	–	100 ± 1
	GSI	68.4 ± 1.5	0.81 ± 0.01	62.4 ± 1.0	133 ± 0
	ExpGSI (ours)	67.2 ± 1.2	$\mathbf{0.77 \pm 0.01}$	62.6 ± 1.0	$\mathbf{110 \pm 3}$
4	S-BoN (π_S)	64.0 ± 0.7	0.44 ± 0.01	–	212 ± 9
	RSD†	67.4 ± 2.0	0.56 ± 0.01	92.9 ± 0.3	199 ± 7
	S-BoN (π_B)	72.0 ± 2.0	0.94 ± 0.01	–	207 ± 3
	GSI	69.0 ± 0.4	0.94 ± 0.02	69.9 ± 0.3	283 ± 4
	ExpGSI (ours)	67.8 ± 1.8	$\mathbf{0.90 \pm 0.02}$	69.6 ± 0.9	$\mathbf{224 \pm 5}$
8	S-BoN (π_S)	63.6 ± 1.7	0.61 ± 0.01	–	517 ± 54
	RSD†	68.8 ± 2.1	0.76 ± 0.01	94.6 ± 0.1	468 ± 21
	S-BoN (π_B)	73.1 ± 1.0	1.18 ± 0.02	–	423 ± 12
	GSI	68.9 ± 1.3	1.23 ± 0.04	75.5 ± 0.6	581 ± 2
	ExpGSI (ours)	69.3 ± 3.0	$\mathbf{1.15 \pm 0.00}$	75.1 ± 0.4	$\mathbf{425 \pm 5}$
16	S-BoN (π_S)	67.5 ± 1.7	0.94 ± 0.02	–	1072 ± 54
	RSD†	69.2 ± 1.7	1.10 ± 0.03	95.7 ± 0.3	1020 ± 84
	S-BoN (π_B)	71.6 ± 2.3	1.56 ± 0.04	–	869 ± 8
	GSI	69.6 ± 1.9	1.81 ± 0.04	80.2 ± 0.7	1224 ± 21
	ExpGSI (ours)	67.8 ± 0.4	$\mathbf{1.53 \pm 0.04}$	79.9 ± 1.0	$\mathbf{822 \pm 20}$

Reward-model ablation (Skywork PRM). Table 13 reports the complete results of the reward-model ablation described in Section 5.3: all five methods on MATH500 at $n \in \{4, 16\}$ over three random seeds, with the 7B math-specialized PRM replaced by SKYWORK-O1-OPEN-PRM-QWEN-2.5-1.5B (Team, 2024) and all other settings kept identical to the main experiments. The table

Table 11: Complete results on Minerva Math across candidate budgets. Reported values are means and standard deviations over three random seeds. Bold efficiency values indicate improvements of ExpGSI over GSI. †RSD uses the implementation and hyperparameter configuration adopted by GSI.

n	Method	Acc. (%)	Time/step (s)	Accept. (%)	Est. TFLOPs/prob.
2	S-BoN (π_S)	25.0 ± 1.3	0.39 ± 0.01	–	149 ± 5
	RSD†	24.9 ± 1.7	0.50 ± 0.02	94.1 ± 0.2	150 ± 4
	S-BoN (π_B)	31.9 ± 1.5	0.90 ± 0.02	–	171 ± 5
	GSI	28.6 ± 1.1	0.77 ± 0.01	78.1 ± 0.3	198 ± 6
	ExpGSI (ours)	29.7 ± 0.6	$\mathbf{0.73 \pm 0.02}$	78.5 ± 0.8	$\mathbf{175 \pm 5}$
4	S-BoN (π_S)	25.7 ± 0.4	0.51 ± 0.01	–	301 ± 10
	RSD†	24.4 ± 1.1	0.60 ± 0.02	96.6 ± 0.5	325 ± 11
	S-BoN (π_B)	31.2 ± 0.6	1.13 ± 0.01	–	363 ± 10
	GSI	28.9 ± 0.8	0.90 ± 0.01	84.1 ± 0.0	445 ± 11
	ExpGSI (ours)	29.4 ± 1.1	$\mathbf{0.88 \pm 0.02}$	83.7 ± 0.5	$\mathbf{342 \pm 10}$
8	S-BoN (π_S)	25.2 ± 0.8	0.74 ± 0.00	–	650 ± 10
	RSD†	24.9 ± 1.3	0.81 ± 0.01	97.3 ± 0.2	654 ± 17
	S-BoN (π_B)	32.5 ± 2.1	1.44 ± 0.05	–	758 ± 11
	GSI	28.8 ± 1.5	1.18 ± 0.02	88.2 ± 0.4	949 ± 38
	ExpGSI (ours)	29.0 ± 0.7	$\mathbf{1.11 \pm 0.05}$	87.6 ± 0.9	$\mathbf{693 \pm 26}$
16	S-BoN (π_S)	26.5 ± 1.0	1.14 ± 0.02	–	1343 ± 19
	RSD†	26.2 ± 2.4	1.27 ± 0.02	98.0 ± 0.4	1427 ± 42
	S-BoN (π_B)	31.6 ± 1.0	1.99 ± 0.12	–	1538 ± 21
	GSI	28.7 ± 0.7	1.80 ± 0.02	90.6 ± 0.2	1977 ± 39
	ExpGSI (ours)	29.4 ± 1.9	$\mathbf{1.50 \pm 0.02}$	90.5 ± 0.4	$\mathbf{1274 \pm 50}$

follows the same format as the per-benchmark tables above, reporting accuracy, time per step, acceptance rate, and estimated TFLOPs per problem.

Sensitivity to the clipping level. We evaluate ExpGSI on MATH500 with $n = 16$ while varying

$$C/\widehat{C} \in \{0.5, 0.75, 1.0, 1.5, 2.0, \infty\},$$

where $\widehat{C} = 0.45$ is used in the main experiments. Each configuration is evaluated on the same 400 problems using two random seeds. We report final-answer accuracy, the average number of candidates scored per reasoning step, and estimated computation per problem. When $C = \infty$, clipping and early exit are disabled. The resulting selection rule replaces the Gumbel noise in GSI with exponential noise and scores all n candidates. Accuracy remains broadly stable over the tested clipping levels. Increasing C generally requires scoring more candidates. Removing clipping requires evaluating all 16 candidates and increases TFLOPs per problem. The finite clipping configurations therefore provide substantial computational savings without showing a consistent loss in accuracy.

Numerical example. We use the example of Verdun et al. (2025, Figure 1) to verify our theorems. Consider $P = (0.75, 0.20, 0.05)$ and $r = (0.016, 0.164, 0.820)$. At $\lambda = 0.5$, we obtain the parameter $\rho_\lambda = 0.745928$ and $P^\star_\lambda = (0.591233, 0.211972, 0.196795)$. For every n, we compute and compare the finite-n SBoN and ExpBoN marginal distributions. For example, at $n = 10$, we have $D_{\mathrm{KL}}(P^\star_\lambda \| \widetilde{P}_{10,\lambda}) = 3.58 \times 10^{-4}$ for ExpBoN versus $D_{\mathrm{KL}}(P^\star_\lambda \| P^{\mathrm{SBoN}}_{10,\lambda}) = 6.16 \times 10^{-3}$ for SBoN, and the relative centered reward gaps are 0.0423 and 0.1706, respectively. Figure 1b shows the exact KL curves; Figure 5 compares the exact quantities with the bounds of Theorem 3.2, Corollary 3.4, and Theorem 3.8. The same separation holds on empirical candidate pools generated by the draft model in the experiments of Section 5, both for reward-only scores and for the clipped score used by ExpGSI (Appendix A.10, Figure 6).

ExpBoN vs. SBoN on real candidate pools. Figure 6 repeats the finite-n comparison of Figure 5 on real candidate pools. Rolling out the calibration problems step by step (Appendix A.9) yields one pool per reasoning step: the 64 next-step candidates drawn from π_S at that prefix, together with

Table 12: Complete per-benchmark results for the Qwen3 configuration (QWEN3-1.7B draft, QWEN3-14B target, thinking mode disabled, same PRM and clipping level as the main experiments) at $n \in \{4, 16\}$. Reported values are means and standard deviations over two random seeds. Bold efficiency values indicate improvements of ExpGSI over GSI. †RSD uses the implementation and hyperparameter configuration adopted by GSI.

Benchmark	n	Method	Acc. (%)	Time/step (s)	Accept. (%)	Est. TFLOPs/prob.
MATH500	4	S-BoN (π_S)	67.6 ± 0.2	0.27 ± 0.00	–	714 ± 13
		RSD†	67.2 ± 0.0	0.28 ± 0.00	99.5 ± 0.0	715 ± 5
		S-BoN (π_B)	74.0 ± 0.4	0.75 ± 0.00	–	802 ± 1
		GSI	68.2 ± 0.7	0.46 ± 0.01	98.4 ± 0.2	1455 ± 1
		ExpGSI (ours)	67.9 ± 0.2	$\mathbf{0.40 \pm 0.01}$	98.2 ± 0.1	$\mathbf{1006 \pm 7}$
	16	S-BoN (π_S)	68.4 ± 0.5	0.70 ± 0.00	–	2874 ± 14
		RSD†	69.0 ± 1.4	0.71 ± 0.00	99.7 ± 0.0	2884 ± 50
		S-BoN (π_B)	73.9 ± 0.2	1.33 ± 0.00	–	3167 ± 71
		GSI	68.5 ± 0.4	1.34 ± 0.01	99.2 ± 0.2	6644 ± 37
		ExpGSI (ours)	69.0 ± 1.1	$\mathbf{0.80 \pm 0.01}$	99.3 ± 0.0	$\mathbf{3181 \pm 11}$
MMLU-STEM	4	S-BoN (π_S)	69.8 ± 0.7	0.28 ± 0.01	–	374 ± 4
		RSD†	71.5 ± 1.1	0.33 ± 0.00	96.5 ± 0.0	395 ± 10
		S-BoN (π_B)	86.0 ± 0.4	0.80 ± 0.00	–	510 ± 7
		GSI	82.1 ± 1.2	0.70 ± 0.02	80.5 ± 0.3	925 ± 1
		ExpGSI (ours)	80.9 ± 0.5	$\mathbf{0.67 \pm 0.00}$	80.3 ± 0.4	$\mathbf{736 \pm 14}$
	16	S-BoN (π_S)	69.8 ± 0.4	0.67 ± 0.00	–	1616 ± 41
		RSD†	74.2 ± 1.1	0.72 ± 0.01	97.7 ± 0.1	1629 ± 22
		S-BoN (π_B)	86.0 ± 1.8	1.30 ± 0.01	–	2002 ± 12
		GSI	81.5 ± 0.0	1.53 ± 0.00	85.9 ± 0.0	4558 ± 87
		ExpGSI (ours)	81.2 ± 0.4	$\mathbf{1.18 \pm 0.03}$	86.0 ± 0.4	$\mathbf{2831 \pm 41}$
Minerva Math	4	S-BoN (π_S)	26.1 ± 2.1	0.30 ± 0.00	–	475 ± 6
		RSD†	28.7 ± 0.5	0.34 ± 0.00	97.1 ± 0.2	513 ± 7
		S-BoN (π_B)	37.9 ± 1.6	0.78 ± 0.01	–	613 ± 14
		GSI	33.3 ± 0.3	0.63 ± 0.00	89.3 ± 0.2	1226 ± 26
		ExpGSI (ours)	32.5 ± 0.8	$\mathbf{0.57 \pm 0.00}$	89.9 ± 0.5	$\mathbf{932 \pm 13}$
	16	S-BoN (π_S)	26.3 ± 1.3	0.75 ± 0.00	–	2058 ± 65
		RSD†	27.4 ± 1.8	0.78 ± 0.00	98.0 ± 0.0	2095 ± 29
		S-BoN (π_B)	37.9 ± 1.6	1.39 ± 0.02	–	2527 ± 18
		GSI	32.9 ± 0.8	1.65 ± 0.00	93.0 ± 0.1	6106 ± 5
		ExpGSI (ours)	35.3 ± 3.1	$\mathbf{1.16 \pm 0.00}$	93.7 ± 0.4	$\mathbf{3495 \pm 48}$

Table 13: Results with the Skywork reward model on MATH500. Values are means and standard deviations over three seeds.

n	Method	Acc. (%)	Time (s)	Accept. (%)	TFLOPs
4	S-BoN (π_S)	76.5±0.0	.44±.00	–	41±1
	RSD†	79.2±1.4	.93±.02	65.5±.8	43±2
	S-BoN (π_B)	82.7±0.9	1.09±.02	–	76±0
	GSI	83.6±1.0	.98±.01	73.0±1.2	141±3
	ExpGSI (ours)	83.8±0.5	1.01±.03	72.2±.7	117±2
16	S-BoN (π_S)	75.1±1.7	.67±.03	–	148±6
	RSD†	80.4±1.0	1.24±.02	72.7±.0	147±4
	S-BoN (π_B)	84.4±1.3	1.51±.03	–	269±4
	GSI	83.2±1.2	1.45±.07	78.9±1.2	540±12
	ExpGSI (ours)	83.1±0.9	**1.41±.01**	79.5±2.3	**410±22**

their PRM rewards and log-likelihood ratios. For each pool we take the empirical distribution over its candidates and compute the *exact* finite-n output laws of ExpBoN and SBoN by one-dimensional numerical integration–no sampling is involved, and the integrator reproduces the exact values of the synthetic example in Figure 5.

Table 14: Sensitivity of ExpGSI to the clipping level on MATH500 with $n = 16$. Accuracy is reported as mean $\pm$ standard deviation over two seeds. When $C = \infty$, clipping and early exit are disabled. Separate runs with two seeds; the default row therefore differs from Table 9.

$C/\widehat{C}$	Acc. (%)	Scored cand./step	Est. TFLOPs/prob.
0.5	82.4 ± 1.9	8.9	892
0.75	83.0 ± 0.4	8.8	892
1.0 (default)	82.9 ± 0.9	9.2	941
1.5	82.3 ± 0.7	9.6	951
2.0	80.8 ± 1.4	10.2	993
∞ (no clipping)	81.9 ± 0.5	16.0	1592

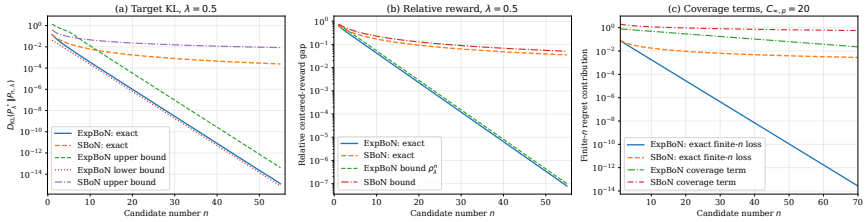


Figure 5: Finite-n comparison on the example $P = (0.75, 0.20, 0.05)$, $r = (0.016, 0.164, 0.820)$, $\lambda = 0.5$: (a) KL divergences with the bounds of Theorem 3.2; (b) relative centered-reward gaps with the bound of Corollary 3.4; (c) true-reward losses under the proxy reward $r_p = r + 0.1(1, 0, -1)$ with the bound of Theorem 3.8. In all panels the exact ExpBoN quantities and bounds decay geometrically, while the SBoN counterparts are polynomial.

The two mechanisms are compared on identical scores in four configurations, one per column of Figure 6. The first two columns use the plain reward score r/λ of Section 2: at the deployed temperature $\lambda = 1/\beta = 1/20$, and at $\lambda = 1/2$, the temperature of the synthetic example in Figure 5–so the second column is the direct real-data counterpart of that example. The last two columns use the GSI score of Section 4 without and with clipping ($\beta r + d$ versus $s_C = \beta r + \min(d, C)$; $\beta = 20$, $C = 0.45$), the latter being the score on which ExpGSI selects candidates (GSI uses the unclipped score). In all four configurations ExpBoN converges geometrically–within a handful of candidates for the reward scores, and by $n \approx 64$ under the GSI scores–whereas SBoN decays polynomially throughout, matching Theorem 3.2 and Corollary 3.4 on real data. Two further observations. First, the likelihood term slows finite-n convergence for both mechanisms–d spreads the scores within a pool, leaving less probability mass near the pool maximum–and clipping partially restores the speed by capping the right tail. Second, slower convergence does not conflict with the compute savings of ExpGSI: convergence is governed by the score dispersion within a pool, whereas early exit is triggered by proximity to the fixed ceiling U_C, and the exit test is exact at every finite n. Figure 3 in the main text complements this rate comparison with a matched-KL analysis of the reward–distribution-shift trade-off on the same pools.

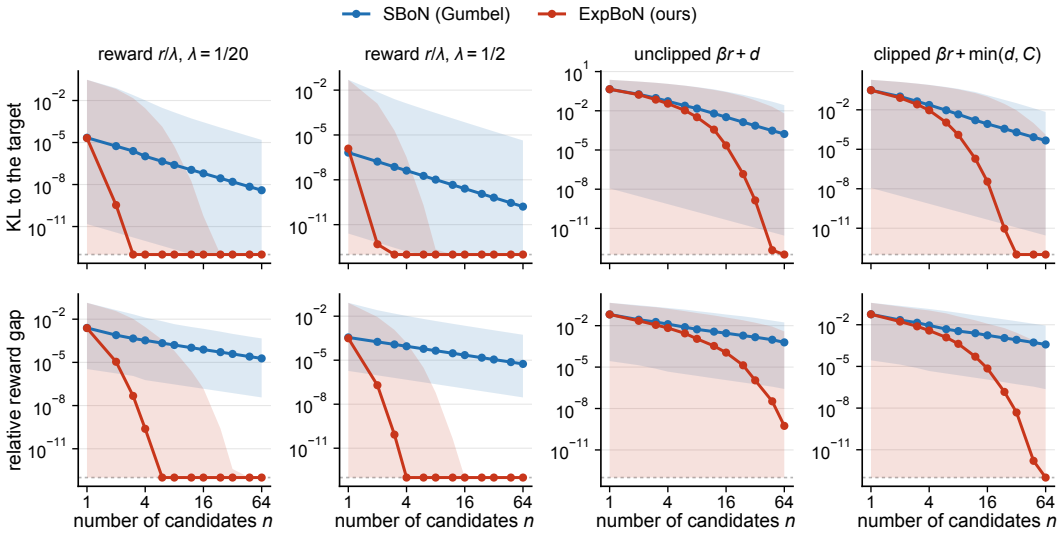


Figure 6: Exact finite-n convergence of ExpBoN vs. SBoN on real LLM candidate pools (medians with 10–90% bands). Columns: the reward score r/λ of Section 2 at the deployed temperature $\lambda = 1/20$ and at $\lambda = 1/2$, the temperature of the synthetic example in Figure 5; the GSI score of Section 4 without clipping ($\beta r + d$) and with clipping ($s_C = \beta r + \min(d, C)$; $\beta = 20$, $C = 0.45$). Rows: KL divergence to the tilted target and relative centered-reward gap. Output laws are computed exactly by numerical integration; the dotted line marks numerical precision (10^{-13}). ExpBoN converges geometrically in every configuration while SBoN decays polynomially; comparing the last two columns, clipping speeds up convergence by capping the right tail of the scores.