RANK-AWARE SPECULATIVE SAMPLING FOR DIFFUSION DRAFT TREES

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative sampling accelerates diffusion generation by verifying inexpensive draft states in parallel while preserving the target law. Recent tree-based methods allocate the parallel compute budget more effectively than single-chain drafts, as demonstrated by Diffusion Greedy Rejection Sampling (D-GRS). D-GRS generates K conditionally independent candidates per node, and sequentially tests them in their generation order. Yet the sampled candidates admit an informative ranking without additional target-model evaluations. To exploit this, we introduce Rank-Aware Speculative Sampling (RASS), a verification rule for speculative draft trees based on rank-aware list coupling. RASS orders draft candidates along the proposal–target mean displacement and samples a rank with weights optimized to minimize total variation between the selected-proposal and target laws. Finally, the selected candidate is maximally coupled with the target, with residual correction ensuring *exact* sampling for any choice of rank weights. We evaluate RASS on a Gaussian-mixture target, unconditional pixel-space generation on FFHQ, conditional generation on CIFAR-10, and latent diffusion with Stable Diffusion 3.5 using COCO2014 prompts. Measured by the ratio of standard to speculative sampling's target-model evaluation counts, RASS improves on D-GRS across the evaluated settings, with gains reaching approximately 20% on CIFAR-10 at matched compute budgets.

1 INTRODUCTION

Diffusion and flow-based models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b; Lipman et al., 2023) produce state-of-the-art samples across modalities (Yang et al., 2023); however, each sample requires tens to hundreds of sequential evaluations of a large network, incurring significant delay at inference. To circumvent this challenge, a body of literature trades exactness for speed through fewer-step solvers (Song et al., 2021a; Zhang & Chen, 2023; Lu et al., 2025) or distilled few-step generators (Salimans & Ho, 2022; Song et al., 2023; Sabour et al., 2025; Yin et al., 2024). Distinctly, De Bortoli et al. (2025) introduces *speculative diffusion sampling*, which uses a cheaper drafter to generate proposals that are subsequently corrected to yield *exact* target samples, thereby reducing the number of expensive target-model evaluations and accelerating inference. The central mechanism enabling speculative sampling is parallel verification of multiple drafted states. Its speed-up therefore hinges on how effectively the available parallel compute budget is allocated.

Reflection maximal coupling (RMC), introduced by De Bortoli et al. (2025), drafts a single chain in which an early rejection discards all states succeeding the inexact draft state. Bullo et al. (2026) show that *draft trees*, which spend the budget on K candidate children per node, use the same batch more effectively, and propose diffusion greedy rejection sampling (D-GRS), a verification rule adapted from relative entropy coding (Harsha et al., 2010; Flamich & Theis, 2023) that tests the K children sequentially in their drafting order. Each rejection updates the unassigned target mass and the next child is tested against it.

We elicit a fundamental limitation of D-GRS: proposals are verified in the order in which they are gleaned, imposing a sequence constraint that is not required by the underlying problem. Indeed, the K children of a node are conditionally independent and identically distributed (i.i.d.) draws from the same Gaussian proposal and are therefore exchangeable at verification. Enforcing an arbitrary order thus discards exploitable symmetry and can be strictly suboptimal relative to list coupling. Building

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.

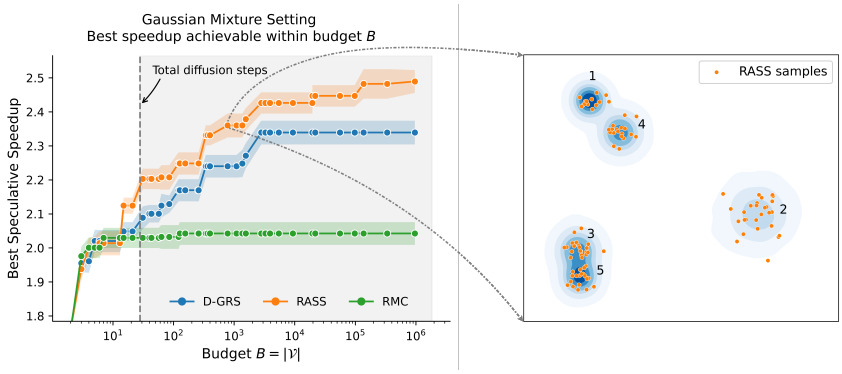


Figure 1: Target-call efficiency and sample visualization on the Gaussian-mixture target ($\varepsilon = 0.3$). **Left:** Best observed mean target-call speedup within each parallel budget $B = |\mathcal{V}|$, with shaded 95% confidence intervals. The vertical dashed line marks the number of diffusion steps. RASS achieves higher speedups than D-GRS and RMC throughout the gray region. **Right:** RASS samples overlaid on target-density contours, with numbers identifying the five mixture components. Experimental details are provided in Section 5.

on this observation, we introduce **rank-aware speculative sampling (RASS)**, a verification rule for diffusion draft trees that exploits the order statistics of the drafted proposals.

More specifically, RASS ranks proposals by their target-to-proposal likelihood ratios, prioritizing candidates that are most compatible with the target distribution. A naive strategy would always select the highest-ranked proposal, but this breaks exactness in an important regime: when the proposal and target distributions coincide, all drafts are distributionally equivalent and no rank should be preferred. RASS therefore places a discrepancy-dependent distribution over proposal ranks, interpolating between uniform selection when proposal and target are close and increasingly concentrated selection on the highest-likelihood proposal as their discrepancy grows.

Contributions. Our contributions are threefold:

(i) We identify a fundamental limitation of the state-of-the-art speculative diffusion sampler D-GRS (Bullo et al., 2026): its verification rule is based on sequence coupling, even though the children of a draft-tree node are exchangeable and list coupling can only improve upon an imposed ordering. Motivated by this gap, we introduce **rank-aware speculative sampling (RASS)**, which ranks i.i.d. proposals by their target-to-proposal likelihood ratios and uses this ranking to construct a stronger proposal distribution for verification (Section 3).

(ii) We derive a universal upper bound on the acceptance probability achievable by any list-coupling scheme in our setting (Section 4, Theorem 3). Existing bounds based on the hockey-stick divergence allow arbitrary dependence among the proposals (Phan et al., 2026); however, such coupling is unavailable in speculative sampling, where the drafts are generated i.i.d. from the proposal distribution. Exploiting this independence constraint, we obtain a strictly tighter characterization of the attainable acceptance probability.

(iii) We prove that RASS preserves the target law for every feasible choice of rank weights (Theorem 2). Numerical comparisons show up to approximately 13% higher per-node acceptance probability than D-GRS with the same number of drafts. Across a Gaussian-mixture target, CIFAR-10, FFHQ, and Stable Diffusion 3.5, RASS consistently improves mean target-call speedup, with gains reaching 20.9% at matched draft trees and parallel compute budgets. On CIFAR-10, it exceeds D-GRS's best observed mean speedup using 81.5% less parallel budget for every tested churn value.

2 Background

Notation. We use uppercase letters for random variables and corresponding lowercase letters for their realizations, and write $X_{1:n} = (X_1, \ldots, X_n)$. The state dimension is d, $\mathbb{I}_d$ denotes the identity

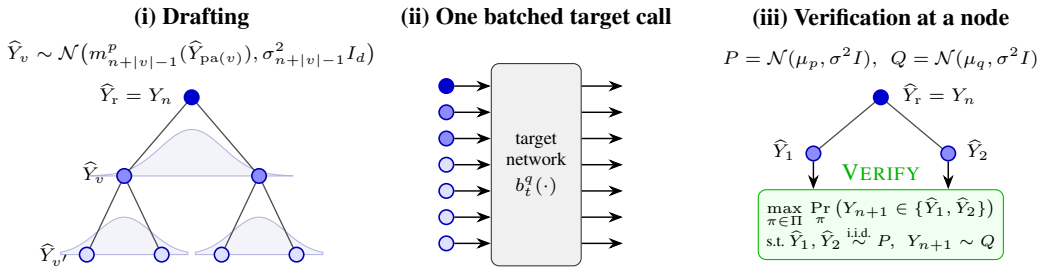


Figure 2: One speculative round on a $(K, L) = (2, 2)$ draft tree. (i) Candidates are drawn recursively from the proposal kernel. (ii) Target drifts required for verification are evaluated in one batch. (iii) Verification walks down the tree while candidates are accepted. The first rejection is corrected to an exact target draw and terminates the round.

matrix, and $\|\cdot\|$ is the Euclidean norm. We write $\mathcal{N}(\mu, \Sigma)$ for the Gaussian law with mean μ and covariance Σ, and $\mathcal{N}(x; \mu, \Sigma)$ for its density at x when Σ is positive definite. The standard Gaussian density and cumulative distribution function are denoted by ϕ and Φ, with $\bar{\Phi} = 1 - \Phi$. We write $\mathcal{U}(a, b)$ for the uniform law on (a, b). For probability laws P and Q, $\Pi(P, Q)$ denotes the set of joint laws with marginals P, Q, and $P^{\otimes K}$ is the joint law of K independent draws from P. For probability densities p, q, their total variation distance is $\mathrm{TV}(p, q) = \frac{1}{2} \int |p(x) - q(x)|\, dx$. Finally, $(x)_+ = \max\{x, 0\}$ denotes the positive part.

Standard Diffusion Sampling. Let $X_0 \sim q_{\mathrm{data}}$ and consider the forward diffusion $\mathrm{d}X_t = f_t(X_t)\,\mathrm{d}t + g_t\,\mathrm{d}B_t$, $t \in [0, 1]$ with marginals q_t and score $s_t(x) = \nabla_x \log q_t(x)$, where the dynamics are chosen such that $q_1 \approx \mathcal{N}(0, \mathbb{I}_d)$ (Song et al., 2021b). A family of reverse processes with the same marginals is

$$\mathrm{d}Y_t = b_t(Y_t)\,\mathrm{d}t + \varepsilon g_{1-t}\,\mathrm{d}W_t, \qquad b_t(x) = -f_{1-t}(x) + \frac{1+\varepsilon^2}{2} g_{1-t}^2 s_{1-t}(x), \tag{1}$$

where $\varepsilon \geq 0$ controls the stochasticity (Song et al., 2021b; Karras et al., 2022; Albergo et al., 2023). In practice, b_t (or, equivalently, s_{1-t}) is approximated by a neural network trained using denoising score matching or flow matching (Song et al., 2021b; Lipman et al., 2023). At inference time, we use the learned drift b_t^q to simulate the reverse process with Euler–Maruyama on the uniform grid $t_n = n\gamma$, $n = 0, \ldots, N$, where $\gamma = 1/N$:

$$Y_{n+1} = Y_n + \gamma b_{t_n}^q(Y_n) + \sqrt{\gamma}, \varepsilon g_{1-t_n} \xi_n, \qquad \xi_n \sim \mathcal{N}(0, \mathbb{I}_d). \tag{2}$$

Hence the target transition at step n is

$$Q_n(\cdot \mid y) = \mathcal{N}\big(m_n^q(y), \sigma_n^2 \mathbb{I}_d\big), \qquad m_n^q(y) = y + \gamma b_{t_n}^q(y), \quad \sigma_n = \sqrt{\gamma}\, \varepsilon g_{1-t_n}. \tag{3}$$

Evaluating m_n^q requires one neural-network evaluation, so generating an N-step trajectory requires N sequential evaluations.

Speculative diffusion sampling. Speculative diffusion sampling reduces this sequential dependence by generating candidate future states with an inexpensive proposal and evaluating the target model at these candidates in parallel (De Bortoli et al., 2025). A subsequent verification step accepts or corrects the candidates while preserving exactly the law of the target Markov chain. We call this three-stage procedure a *speculative round*, illustrated in Figure 2. In the remainder of this section, we summarize its three stages following Bullo et al. (2026), and we defer the full algorithm to Algorithm 2.

(i) Drafting: To generate candidate future states inexpensively, speculative diffusion sampling uses a proposal transition. Analogously to (3), we consider $P_n(\cdot \mid y) = \mathcal{N}\big(m_n^p(y), \sigma_n^2 \mathbb{I}_d\big)$, where $m_n^p(y) = y + \gamma b_{t_n}^p(y)$ and $b_{t_n}^p$ is an inexpensive approximation to the target drift $b_{t_n}^q$. The proposal thus shares the target covariance but uses a cheaper mean computation. In this work, following (De Bortoli et al., 2025), we use a *delayed-target proposal*, which holds a cached target-model evaluation fixed throughout drafting and updates it between rounds using the available parallel evaluations

(see Appendix E.1). The drafting procedure can be organized as a finite rooted tree $\mathcal{T} = (\mathcal{V}, \mathrm{pa})$ of candidate states (Bullo et al., 2026), where $\mathcal{V}$ is the node set, $\mathrm{r} \in \mathcal{V}$ is the root, and $\mathrm{pa} : \mathcal{V} \setminus \{\mathrm{r}\} \to \mathcal{V}$ maps each non-root node to its parent. For $u \in \mathcal{V}$, let $C(u) := \{v \in \mathcal{V} : \mathrm{pa}(v) = u\}$ denote its set of children, and let $|u|$ denote its depth, with $|\mathrm{r}| = 0$. The chain construction of (De Bortoli et al., 2025) is recovered as the degenerate case with branching factor one. The root r contains the current state Y_n, and each node $v \neq \mathrm{r}$ at depth $|v| = \ell$ represents a candidate state for step $n + \ell$. Conditional on its parent, this candidate is drawn from the proposal transition at step $n + \ell - 1$. Writing $\mathbf{y}_A := (y_v)_{v \in A}$ for the collection of states indexed by $A \subseteq \mathcal{V}$, the class of draft models we consider is of the form

$$p_{\mathcal{T}}\big(\mathbf{y}_{\mathcal{V}\setminus\{\mathrm{r}\}} \mid y_{\mathrm{r}}\big) = \prod_{v \in \mathcal{V}\setminus\{\mathrm{r}\}} \mathcal{N}\Big(y_v;\ m^p_{n+|v|-1}\big(y_{\mathrm{pa}(v)}\big),\ \sigma^2_{n+|v|-1}\mathbb{I}_d\Big). \tag{4}$$

In particular, children of the same parent are conditionally i.i.d. proposals for the same target transition. For a full tree with branching factor K and lookahead depth L, every non-leaf node has K children, and depth ℓ contains K^ℓ candidate states. Hence drafting produces $|\mathcal{V}| - 1 = \sum_{\ell=1}^{L} K^\ell$ candidate states.

(ii) Parallel evaluation: Verification requires target-drift evaluations only at the internal nodes $\mathcal{I}(\mathcal{T}) := \{u : C(u) \neq \emptyset\}$, and these evaluations can be batched into a single target-network call. One may therefore define the parallel compute budget using only the internal nodes, $B = |\mathcal{I}|$. This reduces the verification budget, but introduces a caveat: if all drafted nodes are rejected and the algorithm returns a residual sample, the target drift at that new root state is unavailable and must be evaluated before the next draft tree can be generated. An alternative is to verify the entire tree, including the leaves, corresponding to the larger budget $B = |\mathcal{V}|$. Even though this spends additional parallel compute, it provides target-drift evaluations at the leaves in advance. Consequently, if the current tree is fully rejected, the score evaluated at a leaf can be reused as the delayed proposal drift for the next tree, whose root is the residual sample. Full-tree verification also makes same-timestep sibling evaluations available for reuse after a leaf rejection. Since the relative benefit of these two strategies is problem dependent, we follow Bullo et al. (2026) and adopt the full-tree budget $B = |\mathcal{V}|$ in the main experiments, deferring the internal-node alternative $B = |\mathcal{I}|$ to Appendix E.2. Speculative sampling therefore trades a larger parallel batch for fewer *sequential* target-model evaluations.

(iii) Verification: Verification proceeds from the root down the tree, one level at a time, along the accepted branch. We denote the nodewise verification rule by VERIFY. Consider a visited node with proposal $P = \mathcal{N}(\mu_p, \sigma^2\mathbb{I}_d)$, target $Q = \mathcal{N}(\mu_q, \sigma^2\mathbb{I}_d)$, and drafted children $\widehat{Y}_{1:K} \overset{\text{i.i.d.}}{\sim} P$. Given $(\mu_p, \mu_q, \sigma, \widehat{Y}_{1:K})$, VERIFY returns a state Y satisfying $Y \sim Q$. If a child is accepted, then $Y = \widehat{Y}_j$ for some $j \in \{1, \dots, K\}$, and verification continues from that child. Otherwise, VERIFY returns a correction drawn from the appropriate residual distribution, which is committed as the next state and terminates the speculative round. The performance of VERIFY is determined by the probability of accepting one of the drafted children. Formally, this gives the *maximal list-coupling* objective

$$\max_{\pi \in \Pi(P^{\otimes K}, Q)} \Pr_{\pi}\Big(Y \in \{\widehat{Y}_1, \dots, \widehat{Y}_K\}\Big). \tag{5}$$

D-GRS instantiates VERIFY by greedily testing candidates in drafting order against the remaining target mass (Bullo et al., 2026). For $K = 1$, the problem reduces to classical maximal coupling, as used by De Bortoli et al. (2025). For P and Q being isotropic Gaussians with the same covariance, the verification problem reduces to one dimension. Define

$$\Delta := (\mu_q - \mu_p)/\sigma, \qquad \delta := \|\Delta\|, \qquad e := \Delta/\delta, \quad \delta > 0, \tag{6}$$

and, for every node, decompose each standardized child $(\widehat{Y}_k - \mu_p)/\sigma$ into its projection onto e and its orthogonal component as

$$\widehat{S}_k := e^\top(\widehat{Y}_k - \mu_p)/\sigma, \qquad Z_{\perp,k} := (\widehat{Y}_k - \mu_p)/\sigma - e\widehat{S}_k, \qquad k = 1, \dots, K. \tag{7}$$

Under P, $\widehat{S}_{1:K} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, whereas the target projection has density $q_\delta(s) := \phi(s - \delta)$. In both cases, the orthogonal component follows $\mathcal{N}(0, \mathbb{I}_d - ee^\top)$ independently of the projection. Hence, VERIFY needs to generate a scalar S with target density $\mathcal{N}(\delta, 1)$, after which the output is reconstructed as $Y = \mu_p + \sigma(Se + Z_\perp)$. Henceforth, we take this reduction as implicit and regard VERIFY as operating directly on the scalar projections $\widehat{S}_{1:K}$. Our contribution, presented in Section 3, is a new instantiation of VERIFY.

3 RANK-AWARE SPECULATIVE SAMPLING

We now introduce a rank-aware instantiation of VERIFY. RASS ranks the drafted children projections and selects one according to a rank distribution. Then, it maximally couples the resulting selected-child distribution with the target scalar distribution.

Rank-based selection. Define $\widehat{U}_k := \Phi(\widehat{S}_k)$ for $k = 1, \dots, K$. Since $\widehat{S}_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, the probability integral transform gives $\widehat{U}_k \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0,1)$. Let $\widehat{U}_{(1)} < \cdots < \widehat{U}_{(K)}$ denote the corresponding order statistics, and let $j(r)$ be the index of the child with rank r, such that $\widehat{U}_{(r)} = \widehat{U}_{j(r)}$. Since Φ is strictly increasing, this ordering is also the ordering of the projections $\widehat{S}_{1:K}$. Thus, $j(r)$ identifies the child with rank r along e, and rank K corresponds to the child with the largest projection toward the target mean. Let $\mathcal{W}_K$ denote the K-dimensional probability simplex. Given rank weights $\omega \in \mathcal{W}_K$ (which depend only on (δ, K)), RASS samples

$$R \sim \text{Categorical}(\omega), \qquad J := j(R), \tag{8}$$

with the selected proposal being $\widehat{Y}_J$. While R is independent of the drafted children, the selected index J depends on their projections through j. The selected scalar projection is then $\widehat{S}_J = \Phi^{-1}(\widehat{U}_{(R)})$, with Φ^{-1} denoting the inverse CDF of the standard normal distribution. The next proposition characterizes the distribution induced by this rank-based selection.

Proposition 1 (Law of selected child $\widehat{S}_J$)**.** *For any $\omega \in \mathcal{W}_K$, the selected projection $\widehat{S}_J$ has density*

$$p_\omega(s) = \phi(s)\, b_\omega(\Phi(s))\,, \qquad \textit{with} \quad b_\omega(a) := \sum_{r=1}^{K} \omega_r f_r(a), \tag{9}$$

where f_r denotes the density of a $\text{Beta}(r, K+1-r)$ random variable. Furthermore, conditional on $\widehat{S}_{1:K}$ and R, the selected orthogonal component satisfies $Z_{\perp,J} \sim \mathcal{N}(0, \mathbb{I}_d - ee^\top)$.

Two limiting choices of ω illustrate the effect of rank selection. First, uniform weights, $\omega_r = 1/K$, satisfy $b_\omega(a) = 1$. Consequently, $p_\omega = \phi$, so uniform rank selection recovers the original proposal distribution. Second, the top-rank choice $\omega_K = 1$ selects the child with the largest projection $\widehat{S}_{(K)} = \max(\widehat{S}_1, \dots, \widehat{S}_K)$, whose density is $p_\omega(s) = K\phi(s)\Phi(s)^{K-1}$. This choice biases the selected proposal toward larger projections along the target direction. For small δ, this bias can overshoot the target, reducing the overlap with the target relative to the original proposal ϕ. As delineated in what follows, we adopt a weight selection scheme that *adaptively* interpolates between uniform and top-rank selection.

Verification by maximal coupling. Rank-based selection converts the K drafted children into a single rank-selected candidate $\widehat{Y}_J$. For fixed rank weights ω, its scalar projection $\widehat{S}_J$ follows the induced density p_ω, which in general differs from the target density q_δ. Following the verification procedure of Section 2, RASS maximally couples p_ω with the target q_δ, maximizing the probability of accepting the selected child. Equivalently, RASS chooses the rank mixture so that the induced selected-child measure is as close as possible to the target, thereby maximizing the maximal-coupling acceptance probability. Specifically, let $Y \sim Q$ denote the target-distributed output of the verifier. For a given selected child $\widehat{Y}_J = y$ with scalar projection $\widehat{S}_J = s$, the acceptance probability is

$$a_\omega(s) := \Pr\big(Y = \widehat{Y}_J \mid \widehat{S}_J = s\big) = \min\big\{1,\, q_\delta(s)/p_\omega(s)\big\}, \qquad s \in \mathbb{R}. \tag{10}$$

Therefore, the acceptance rate is

$$\alpha_K(\delta;\omega) := \Pr(\widehat{Y}_J = Y) = \int_{\mathbb{R}} a_\omega(s) p_\omega(s)\, \mathrm{d}s = \int_{\mathbb{R}} \min\{q_\delta(s), p_\omega(s)\}\, \mathrm{d}s = 1 - \text{TV}(q_\delta, p_\omega). \tag{11}$$

On rejection, the remaining mass is $q_\delta(s) - \min\{q_\delta(s), p_\omega(s)\} = (q_\delta(s) - p_\omega(s))_+$. Hence, to guarantee exactness, RASS draws $S \sim (q_\delta(s) - p_\omega(s))_+ / \text{TV}(q_\delta, p_\omega)$, and reconstructs $Y = \mu_p + \sigma(Se + Z_{\perp,1})$, with the orthogonal component of the first child in drafting order. The rank weights ω

Algorithm 1 VERIFY ← RASS

Require: $\mu_p, \mu_q, \sigma > 0$; children $\widehat{Y}_{1:K} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_p, \sigma^2 I_d)$
Ensure: $(Y, \texttt{accepted}, j)$

Compute $\widehat{S}_{1:K}$ and $Z_{\perp,1}$ as in (7), $\delta \leftarrow \|\mu_q - \mu_p\|/\sigma$
if $\delta = 0$ **then**
  return $(\widehat{Y}_1, \texttt{true}, 1)$
end if
Compute $\omega^\star$ by solving (12) ▷ (See Appendix D.1)
$R \sim \text{Categorical}(\omega^\star), \quad J \leftarrow j(R)$
$A \sim \text{Bernoulli}\Big(\min\{1, q_\delta(\widehat{S}_J)/p_\omega(\widehat{S}_J)\}\Big)$
if $A = 1$ **then**
  return $(\widehat{Y}_J, \texttt{true}, J)$
end if
Draw $S \sim (q_\delta(s) - p_\omega(s))_+/\text{TV}(q_\delta, p_\omega)$ ▷ (See Appendix D.2)
return $\big(\mu_p + \sigma(Se + Z_{\perp,1}), \texttt{false}, K+1\big)$

determine the selected-proposal density p_ω and hence the acceptance probability $\alpha_K(\delta;\omega)$ in (11). For fixed (δ, K), we choose

$$\omega^\star(\delta, K) \in \arg\max_{\omega \in \mathcal{W}_K} \alpha_K(\delta;\omega). \tag{12}$$

The RASS verification rule is reported in Algorithm 1, and implementation details for sampling the residual distribution, and the computational complexity for solving (12) are deferred to Appendix D.2 and D.1, respectively.

4 THEORETICAL ANALYSIS

In this section, we first establish that RASS preserves the target law for every feasible rank distribution. We then characterize the maximum acceptance probability attainable from K i.i.d. drafts, providing a common benchmark for RASS and D-GRS. Proofs are deferred to Appendix C.

Theorem 2 (Exactness)**.** *For every $K \geq 1$ and every $\delta \geq 0$, the RASS verifier is exact for any feasible rank weights $\omega \in \mathcal{W}_K$ used by the verifier.*

Exactness holds for any choice of rank weights, allowing us to optimize acceptance without changing the target law. To assess how effectively RASS uses the draft list, we first determine the best acceptance probability achievable by any exact verifier under the same drafting distribution.

Proposition 3 (Optimal list acceptance)**.** *In the Gaussian setting of Section 2, the maximum probability of accepting a draft while preserving the target distribution is*

$$\alpha_K^{\text{up}}(\delta) = 1 - \sup_{s \in \mathbb{R}} \left\{ \Phi(s)^K - \Phi(s-\delta) \right\},$$

where Φ denotes the standard Gaussian CDF.

Note that Proposition 3 characterizes the maximal acceptance probability achievable by any list-coupling scheme when the K proposals are sampled *independently*. This yields a tighter bound than Phan et al. (2026, Theorem 2.2), which allows arbitrary coupling among the draft samples—an assumption that is not available in our setting. Proposition 3 reflects a simple limitation: a verifier can return a draft above a threshold s only if the list contains one. The target assigns probability $1 - \Phi(s-\delta)$ to this region, whereas the list contains such a draft with probability $1 - \Phi(s)^K$. Any shortfall must therefore be supplied through rejection and correction. Maximizing this shortfall over s gives a lower bound on rejection.Thus, $\alpha_K^{\text{up}}(\delta)$ provides a common benchmark for RASS and D-GRS, measuring how much room remains to improve acceptance with the same draft budget. Figure 3 compares the acceptance probabilities of RASS and D-GRS with the optimal i.i.d. list-coupling bound $\alpha_K^{\text{up}}(\delta)$, numerically demonstrating higher acceptance for RASS across the tested positive mean displacements, with the largest gains at intermediate values of δ.

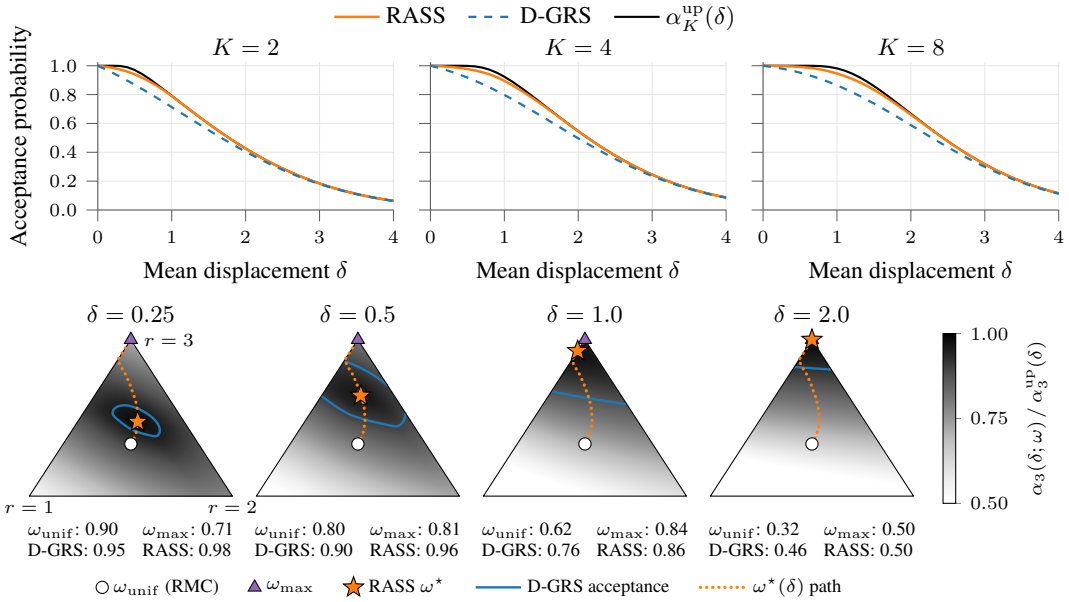


Figure 3: RASS and D-GRS acceptance compared with $\alpha_K^{\mathrm{up}}(\delta)$. *Top:* acceptance versus δ for $K \in \{2, 4, 8\}$, with optimized RASS weights from (12). *Bottom:* relative acceptance $\alpha_3(\delta;\omega)/\alpha_3^{\mathrm{up}}(\delta)$ on $\mathcal{W}_3$ at four displacements. The solid blue contour encloses weights outperforming D-GRS; the dotted curve traces $\omega^\star(\delta)$ for $\delta \in (0, 2.5]$. Acceptance values are reported below each panel.

5 Experiments

We compare RASS with D-GRS (Bullo et al., 2026) and RMC (De Bortoli et al., 2025) on a Gaussian-mixture target, class-conditional CIFAR-10 and unconditional FFHQ generation in pixel space, and latent text-to-image generation with Stable Diffusion 3.5 (SD3.5) on COCO 2014 captions. Within each experiment, all methods use the same target discretization and delayed-velocity proposal, reusing target evaluations between speculative rounds. All three methods preserve the discretized target law, with exactness established for RASS in Theorem 2. Consistently with this guarantee, Table 1 reports comparable FID and CLIP scores across methods. RASS and D-GRS use identical tree configurations, while RMC uses a chain matched to the same budget and truncated at the sampling horizon. Following Section 2, parallel target evaluation includes the leaves, with budget $B = |\mathcal{V}|$. We report mean per-trajectory target-call speedup over 100 trajectories per configuration for the Gaussian-mixture and SD3.5 experiments and 500 for CIFAR-10 and FFHQ. Shaded bands indicate 95% confidence intervals. Implementation details and results with verification restricted to internal nodes are reported in Appendix E.

Gaussian-mixture target. The target distribution is an equally weighted mixture of five isotropic Gaussians in $\mathbb{R}^{512}$. We sample uniformly the component means from $[-2, 2]^{512}$ and the standard deviations from $[0.1, 0.25]$. We use a 30-step schedule, churn value $\varepsilon = 0.3$, and the full grid $K, L \in \{2, \dots, 7\}$. Figure 1 shows that RASS achieves higher mean speedup than D-GRS at every evaluated budget, reaching a 6.4% higher maximum. Within the tested grid, RASS exceeds D-GRS's highest observed mean speedup using only 28% of its budget. In Appendix E.2, we report results for $\varepsilon \in \{0.06, 0.1, 0.6\}$.

Pixel-space diffusion. We use the pretrained denoisers of Karras et al. (2022), with the velocity parameterization described in Appendix E.1. CIFAR-10 generates 32×32 images conditioned on uniformly sampled class labels, while FFHQ generates unconditional 64×64 images. We use a 100-step schedule and evaluate $(K, L) \in \{(2, 3), (2, 5), (3, 4), (4, 4)\}$ at $\varepsilon \in \{0.1, 0.3, 0.6\}$. These configurations correspond to full-tree budgets $B \in \{15, 63, 121, 341\}$. Figure 4 shows higher mean

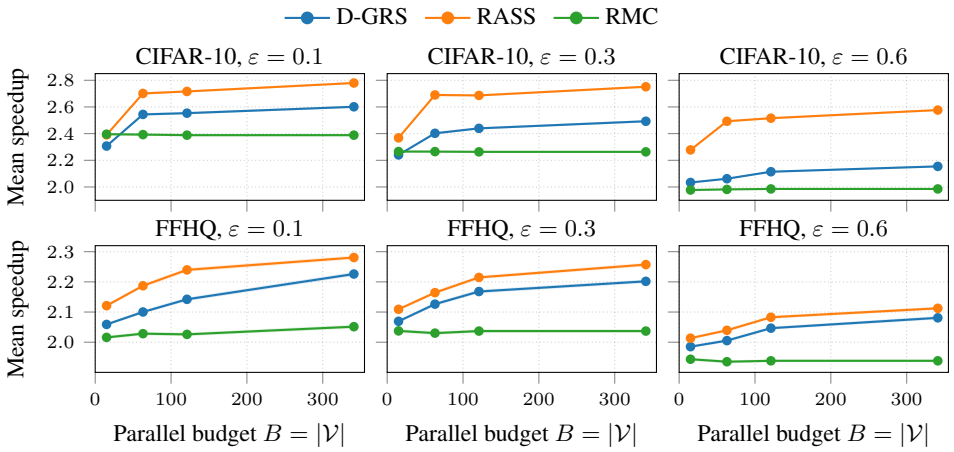


Figure 4: Mean target-call speedup on CIFAR-10 (top) and FFHQ (bottom) across parallel budgets and churn values. Curves average over 500 images per configuration, and shading shows 95% confidence intervals.

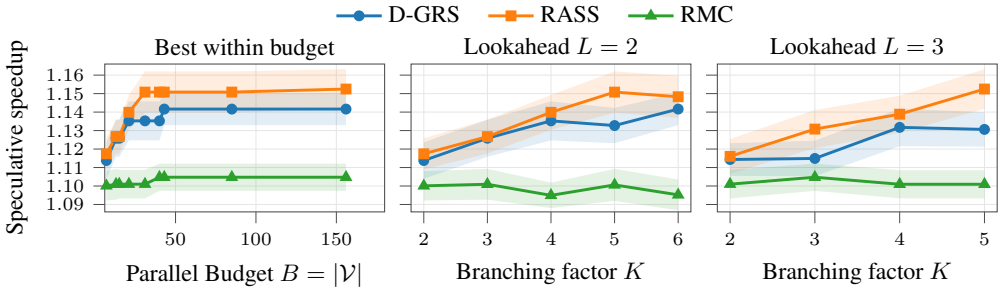


Figure 5: Mean target-call speedup for SD3.5-medium. Left: best observed mean speedup within parallel budget $B = |\mathcal{V}|$. Middle and right: speedup versus branching factor K at lookahead $L = 2$ and $L = 3$, respectively. Shading indicates 95% confidence intervals over 100 paired caption–noise inputs.

speedups for RASS than D-GRS in every evaluated configuration. On CIFAR-10, the relative improvement increases with churn, reaching 20.9% at $\varepsilon = 0.6$ and $B = 63$. On FFHQ, whose state dimension is 4× larger, absolute speedups are lower than CIFAR-10, but RASS consistently improves on D-GRS by 1.4% to 4.6%. RASS also uses the parallel budget more efficiently. In fact, on CIFAR-10, its mean speedup at $B = 63$ exceeds D-GRS's at $B = 341$ for all tested churn values. On FFHQ, the same comparison holds with RASS at $B = 121$ for $\varepsilon \leq 0.3$.

High-dimensional Latent-space Text-to-image. We use Stable Diffusion 3.5 (SD3.5)[1] with the linear-interpolant SDE parameterization of Bullo et al. (2026). We generate 512×512 images in a $16 \times 64 \times 64$ latent space ($d = 65{,}536$). We use $N = 50$ steps, classifier-free guidance 7.0, timestep shift 3.0, and churn $\varepsilon = 0.8$, evaluating $K \in \{2, 3, 4, 5, 6\}$ and $L \in \{2, 3\}$ on 100 shared COCO 2014 validation captions with paired initial noise. Figure 5 shows higher mean speedups for RASS than both baselines in every evaluated configuration, reaching 1.152× versus 1.142× for D-GRS and 1.105× for RMC. RASS exceeds D-GRS's best observed mean speedup at $B = 31$ using 28% less parallel budget. At matched configurations, its relative improvement over D-GRS reaches 1.93%, at $(K, L) = (5, 3)$, i.e., $B = 156$.

[1]The model is available in Huggingface as `stabilityai/stable-diffusion-3.5-medium`.

	FID ↓		SD3.5 CLIP ↑				
Sampler	CIFAR-10	FFHQ	$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$
Target	1.838	2.929			26.68 ± 0.33		
RMC	1.871	2.898	26.53 ± 0.33	26.74 ± 0.32	26.76 ± 0.35	26.48 ± 0.33	26.80 ± 0.33
D-GRS	1.854	2.902	26.50 ± 0.29	26.69 ± 0.37	26.48 ± 0.33	26.71 ± 0.33	26.82 ± 0.34
RASS	1.844	2.936	26.53 ± 0.33	26.49 ± 0.35	26.31 ± 0.33	26.73 ± 0.33	26.82 ± 0.33

Table 1: Sample quality. FID (↓) compares the empirical means and covariances of Inception-v3 features from 50k generated images and real reference images on CIFAR-10 and FFHQ, with $(K, L) = (2, 3)$ and $\varepsilon = 0.3$. For SD3.5, CLIP score (↑) embeddings are obtained from `openai/clip-vit-large-patch14`. We report mean ± standard error over 100 shared COCO 2014 prompts at $L = 2$ and $\varepsilon = 0.8$. The target reports the standard sampler without speculation.

6 Limitations

Following the convention in the growing literature on speculative diffusion sampling (De Bortoli et al., 2025; Bullo et al., 2026), we report speed-up in terms of reductions in sequential target-model evaluations at matched parallel compute budgets. This metric captures the algorithmic benefit of speculation, but translating it into wall-clock acceleration is inherently hardware dependent. Speculative sampling trades fewer sequential target calls for larger parallel batches, additional drafting, and verification overhead; latency gains therefore materialize only when sufficiently capable accelerators can execute these operations efficiently, a requirement which is challenging in an academic setting. Our reported speed-ups should therefore be interpreted primarily as improvements in target-call efficiency rather than guaranteed wall-clock acceleration. Appendix D.3 separately quantifies the scalar verification overhead, while optimizing the full systems pipeline remains an important direction for future work. Beyond these systems constraints, RASS also relies on specific distributional assumptions. In particular, the drafts must be conditionally independent Gaussian samples sharing the isotropic covariance of the target transition. This restricts its direct applicability to stochastic samplers satisfying this structure and motivates extensions to more general proposal and covariance families.

7 Conclusion and Future Work

We introduced RASS, a rank-aware verification rule for speculative diffusion draft trees. By optimizing a distribution over candidate ranks and maximally coupling the selected proposal with the target, RASS improves acceptance while preserving the target law for every feasible choice of rank weights. We also characterized the optimal acceptance probability attainable from independent Gaussian drafts, providing a common benchmark for exact list-based verifiers. Across Gaussian-mixture, pixel-space, and latent text-to image experiments, RASS consistently improves mean target call speedup over D-GRS and RMC at matched parallel budgets.

Several directions remain open. First, the gap between optimized rank mixtures and the optimal list-coupling bound motivates richer selection rules that exploit more information from the realized draft list. Developing computationally efficient verifiers that close this gap would further improve the use of parallel proposals. Second, integrating RASS with efficient batching, rank-weight precomputation, and adaptive tree construction may help translate target-call savings into lower generation latency. Finally, continuous diffusion language models offer a promising application: RePlaid's stochastic sampler without self-conditioning has Gaussian transitions compatible with RASS when proposals retain the target covariance (Yang et al., 2026). Evaluation and handling self-conditioning dependencies remain future work.

8 Disclosure of using AI

We used generative AI to assist with writing, literature searches, numerical checks, and interpreting results and limitations. The authors reviewed all AI-assisted material and take full responsibility for the manuscript's accuracy and integrity.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Marcello Bullo, Yanxiao Liu, Öykü Sıla Güner, Arpan Mukherjee, and Deniz Gündüz. Accelerating diffusion sampling via speculative draft trees. *arXiv preprint arXiv:2609.17691*, 2026. URL https://arxiv.org/abs/2609.17691.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Haoxuan Chen, Yinuo Ren, Lexing Ying, and Grant M Rotskoff. Accelerating diffusion models with parallel sampling: Inference at sub-linear time complexity. In *Advances in Neural Information Processing Systems*, volume 37, pp. 133661–133709, 2024.

Majid Daliri, Christopher Musco, and Ananda Theertha Suresh. Coupling without communication and drafter-invariant speculative decoding. In *2025 IEEE International Symposium on Information Theory (ISIT)*, pp. 1–6. IEEE, 2025.

Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling. In *International Conference on Machine Learning (ICML)*, pp. 12590–12631. PMLR, 2025.

Sander Dieleman. Learning the integral of a diffusion model, 2026. URL https://sander.ai/2026/05/06/flow-maps.html.

Gergely Flamich. Greedy poisson rejection sampling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 37089–37127, 2023.

Gergely Flamich and Deniz Gündüz. Data compression with stochastic codes. *IEEE BITS the Information Theory Magazine*, 2026. doi: 10.1109/MBITS.2026.3702602.

Gergely Flamich and Lucas Theis. Adaptive greedy rejection sampling. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pp. 454–459. IEEE, 2023.

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.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30467–30476, 2026.

Prahladh Harsha, Rahul Jain, David McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. *IEEE Transactions on Information Theory*, 56(1):438–449, 2010.

Spencer Hill, Fady Alajaji, Tamás Linder, and Gergely Flamich. Rejection sampling is optimal for relative entropy coding. *To appear at the 2026 IEEE Information Theory Workshop (ITW)*, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Zhengmian Hu, Tong Zheng, Vignesh Viswanathan, Ziyi Chen, Ryan Rossi, Yihan Wu, Dinesh Manocha, and Heng Huang. Towards optimal multi-draft speculative decoding. In *International Conference on Learning Representations (ICLR)*, pp. 3181–3203, 2025.

Qi Huangfu and J. A. Julian Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022.

Ashish Khisti, M Reza Ebrahimi, Hassan Dbouk, Arash Behboodi, Roland Memisevic, and Christos Louizos. Multi-draft speculative sampling: Canonical decomposition and theoretical limits. In *International Conference on Learning Representations (ICLR)*, 2025.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning (ICML)*, pp. 19274–19286. PMLR, 2023.

Cheuk Ting Li. Channel simulation: Theory and applications to lossy compression and differential privacy. *Foundations and Trends in Communications and Information Theory*, 21(6):847–1106, 2024.

Cheuk Ting Li and Venkat Anantharam. A unified framework for one-shot achievability via the Poisson matching lemma. *IEEE Transactions on Information Theory*, 67(5):2624–2651, 2021.

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.

Yijun Lin, Jinhao Sheng, Qingyue Cai, and Feng Zhou. Spectr-gbv: Multi-draft block verification accelerating speculative decoding. *arXiv preprint arXiv:2604.25925*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Yanxiao Liu and Cheuk Ting Li. One-shot coding over general noisy networks. *IEEE Transactions on Information Theory*, 71(11):8346–8357, 2025.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 932–949, 2024.

Buu Phan, Gergely Flamich, Ashish Khisti, and Shahab Asoodeh. Multi-marginal couplings for metropolis-hastings. *arXiv preprint arXiv:2605.12807*, 2026.

Joseph Rowan, Truong Buu Phan, and Ashish Khisti. List-level distribution coupling with applications to speculative decoding and lossy compression. In *Advances in Neural Information Processing Systems*, volume 38, pp. 5576–5619, 2025.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 146459–146512, 2025.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 4263–4276, 2023.

Alexander Soen, Hisham Husain, Valentin De Bortoli, and Arnaud Doucet. Accelerating speculative diffusions via block verification. *arXiv preprint arXiv:2606.13426*, 2026.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pp. 32211–32252. PMLR, 2023.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. Spectr: Fast speculative decoding via optimal transport. In *Advances in Neural Information Processing Systems*, volume 36, pp. 30222–30242, 2023.

Zhiwei Tang, Jiasheng Tang, Hao Luo, Fan Wang, and Tsung-Hui Chang. Accelerating parallel sampling of diffusion models. In *International Conference on Machine Learning (ICML)*, 2024.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

Zhihan Yang, Wei Guo, Shuibai Zhang, Subham Sekhar Sahoo, Yongxin Chen, Arash Vahdat, Morteza Mardani, and John Thickstun. Continuous diffusion scales competitively with discrete diffusion for language. *arXiv preprint arXiv:2605.18530*, 2026.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, pp. 47455–47487, 2024.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *International Conference on Learning Representations*, 2023.

Chuqin Zhou, Xiaoyue Ling, Yunuo Chen, Jincheng Dai, Guo Lu, and Wenjun Zhang. Dual-representation image compression at ultra-low bitrates via explicit semantics and implicit textures. *arXiv preprint arXiv:2602.05213*, 2026a.

Huanjian Zhou and Masashi Sugiyama. Parallel simulation for log-concave sampling and score-based diffusion models. In *International Conference on Machine Learning (ICML)*, pp. 79192–79225. PMLR, 2025.

Linqi Zhou, Mathias Parger, Ayaan Haque, and Jiaming Song. Terminal velocity matching. In *The Fourteenth International Conference on Learning Representations*, 2026b.

A Related Work

Accelerating diffusion sampling. Training-free solvers reduce the number of steps (Song et al., 2021a; Zhang & Chen, 2023; Lu et al., 2025); distillation and flow-map methods learn few-step generators (Salimans & Ho, 2022; Song et al., 2023; Sabour et al., 2025; Geng et al., 2026; Zhou et al., 2026b; Yin et al., 2024; Sauer et al., 2024; Dieleman, 2026); parallel-in-time methods trade compute for latency through iterative refinement (Shih et al., 2023; Tang et al., 2024; Chen et al., 2024; Zhou & Sugiyama, 2025). All of these change the sampling distribution or require training. Speculative diffusion sampling (De Bortoli et al., 2025) is exact; Soen et al. (2026) accelerate it with block verification and Bullo et al. (2026) with draft trees and greedy rejection sampling. RASS is a verification rule for the latter framework and is orthogonal to the proposal used to draft.

Speculative decoding and multi-draft coupling. Speculative decoding for language models (Leviathan et al., 2023; Chen et al., 2023) has been extended to tree-structured drafts (Miao et al., 2024) and to multi-draft verification, where several candidate tokens are coupled with the target through optimal transport (Sun et al., 2023), canonical decompositions with theoretical limits (Khisti et al., 2025; Hu et al., 2025), block verification (Lin et al., 2026), and list-level couplings (Rowan et al., 2025; Daliri et al., 2025). Those constructions operate on discrete distributions over a vocabulary, where the optimal list coupling is a transport problem of vocabulary size. In the continuous, high-dimensional Gaussian setting of diffusion, the rank-1 structure reduces the problem to a scalar, and RASS's rank family turns it into a K-dimensional LP that does not depend on d; to our knowledge, rank selection with optimized weights has not been used as a coupling device before.

Relative entropy coding. GRS (Harsha et al., 2010; Flamich & Theis, 2023; Flamich, 2023) develops relative entropy coding, also known as channel simulation, schemes that exactly communicate samples from a target probability distribution to a remote party and are based on rejection sampling. Another widely used relative entropy coding scheme is based on the Poisson functional representation (Li & El Gamal, 2018), which has recently found a wide range of applications in information theory (Li & Anantharam, 2021; Liu & Li, 2025), machine learning (Zhou et al., 2026a; Flamich et al., 2026), and differential privacy. Its optimality (Hill et al., 2026) underlies D-GRS; see also the general references by Li (2024); Flamich & Gündüz (2026). RASS replaces the sequential residual-measure recursion with a single maximal coupling of a selected law.

B Speculative Sampling Algorithm

Algorithm 2 implements the draft-tree sampler described in Section 2, following Bullo et al. (2026, Algorithm 1). It accommodates any VERIFY rule with the interface of Algorithm 1, including RASS, D-GRS, and, when $K = 1$, the maximal coupling of De Bortoli et al. (2025). We use the notation of Section 2 and write $\mathcal{T}|_{L_n}$ for the tree truncated at depth $L_n = \min\{L, N - n\}$, ensuring that each round stays within the sampling horizon. At each node u, we enumerate its children in drafting order as $(v_1, \dots, v_{K_u})$, where $K_u = |\mathcal{C}(u)|$.

The evaluation set $\mathcal{E}_n$ determines the parallel budget. We evaluate the full tree, $\mathcal{E}_n = \mathcal{V}(\mathcal{T}_n)$, in the main experiments, and report the internal-node alternative, $\mathcal{E}_n = \mathcal{I}(\mathcal{T}_n)$, in Appendix E.2. Verification requires only the target means at internal nodes, while evaluations at leaves provide additional values for the prefetch cache described in Appendix E.1. Each round commits $1 \le N_\alpha \le L_n$ transitions, including the correction step when a proposal is rejected. The template also supports unequal branching factors and leaves at different depths.

To preserve the target trajectory law, a plug-in rule must satisfy the exactness requirement of Section 2 at each node. Its decision may depend only on the current transition parameters, the node's children, and auxiliary randomness. Consequently, conditional on an accepted child, its descendants retain their proposal-tree law, allowing verification to continue recursively. Deterministic steps ($\sigma_n = 0$) are handled separately by applying the target transition directly.

Algorithm 2 Speculative diffusion sampler with a pluggable VERIFY rule.
Adapted from Bullo et al. (2026, Algorithm 1).

Require: Initial law q_0; means m^p, m^q; scales $\sigma_{0:N-1} > 0$; horizon N; tree $\mathcal{T}$; rule VERIFY
Ensure: Target trajectory $Y_{1:N}$

$n \leftarrow 0$; draw $Y_0 \sim q_0$
while $n < N$ **do**
  $L_n \leftarrow \min\{L, N-n\}$, $\mathcal{T}_n \leftarrow \mathcal{T}|_{L_n}$, $\widehat{Y}_{\mathrm{r}} \leftarrow Y_n$
  (i) *Drafting: expand each layer in parallel.*
  for $\ell = 1, \dots, L_n$ **do**
    for all nodes u at depth $\ell - 1$ in $\mathcal{T}_n$, in parallel **do**
      Draw $\{\widehat{Y}_v : v \in \mathcal{C}(u)\} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(m^p_{n+|u|}(\widehat{Y}_u), \sigma^2_{n+|u|} I_d)$
    end for
  end for
  (ii) *Parallel Evaluation: one batched target-model call.*
  Compute $m^q_{n+|u|}(\widehat{Y}_u)$ for all $u \in \mathcal{E}_n$ in parallel ▷ $\mathcal{E}_n = \mathcal{V}(\mathcal{T}_n)$ or $\mathcal{I}(\mathcal{T}_n)$
  (iii) *Verification: walk down the tree until rejection or a leaf.*
  $u \leftarrow \mathrm{r}$; $N_\alpha \leftarrow 0$
  while $u \in \mathcal{I}(\mathcal{T}_n)$ **do**
    $k \leftarrow n + |u|$
    $(Y_{k+1}, \texttt{accepted}, j) \leftarrow \text{VERIFY}\big(m^p_k(\widehat{Y}_u), m^q_k(\widehat{Y}_u), \sigma_k; \widehat{Y}_{v_1}, \dots, \widehat{Y}_{v_{K_u}}\big)$
    $N_\alpha \leftarrow N_\alpha + 1$
    if $\neg$ `accepted` **then**
      break
    end if
    $u \leftarrow v_j$
  end while
  $n \leftarrow n + N_\alpha$
end while
return $Y_{1:N}$

C THEORETICAL ANALYSIS

C.1 PROOF OF PROPOSITION 1

Proposition 1 (Law of selected child $\widehat{S}_J$)**.** *For any $\omega \in \mathcal{W}_K$, the selected projection $\widehat{S}_J$ has density*

$$p_\omega(s) = \phi(s)\, b_\omega(\Phi(s))\,, \qquad \textit{with} \quad b_\omega(a) := \sum_{r=1}^{K} \omega_r f_r(a), \tag{9}$$

where f_r denotes the density of a $\mathrm{Beta}(r, K+1-r)$ random variable. Furthermore, conditional on $\widehat{S}_{1:K}$ and R, the selected orthogonal component satisfies $Z_{\perp,J} \sim \mathcal{N}(0, \mathbb{I}_d - ee^\top)$.

Proof. We first characterize the CDF of the r-th order statistic.

Fix a rank r and define

$$M_a := \sum_{k=1}^{K} \mathbb{1}\{\widehat{U}_k \le a\}.$$

Since $\widehat{U}_k \sim \mathcal{U}(0,1)$ independently, $M_a \sim \mathrm{Binomial}(K, a)$. Moreover,

$$\{\widehat{U}_{(r)} \le a\} = \{M_a \ge r\}.$$

Therefore,

$$\Pr(\widehat{U}_{(r)} \le a) = \sum_{j=r}^{K} \binom{K}{j} a^j (1-a)^{K-j},$$

which is the CDF of a $\mathrm{Beta}(r, K+1-r)$ random variable. that is the CDF of a $\mathrm{Beta}(r, K+1-r)$ random variable, whose density is

$$f_r(a) := \frac{K!}{(r-1)!\,(K-r)!} a^{r-1}(1-a)^{K-r}, \qquad a \in (0,1). \tag{13}$$

Since R is independent of the children, $\widehat{U}_{(R)}$ has density

$$b_\omega(a) = \sum_{r=1}^{K} \omega_r f_r(a).$$

Since $\widehat{S}_J = \Phi^{-1}(\widehat{U}_{(R)})$, and equivalently $\widehat{U}_{(R)} = \Phi(\widehat{S}_J)$, the monotone change-of-variables formula with $a = \Phi(s)$ yields

$$p_\omega(s) = b_\omega(\Phi(s))\,\Phi'(s) = \phi(s) b_\omega(\Phi(s))\,.$$

Finally, the variables $Z_{\perp,1:K}$ are i.i.d. $\mathcal{N}(0, \mathbb{I}_d - ee^\top)$ and are independent of the projections $\widehat{S}_{1:K}$. Since J is determined by $(\widehat{S}_{1:K}, R)$, selecting an index according to J does not change the conditional distribution of its orthogonal component. Hence $Z_{\perp,J} \sim \mathcal{N}(0, \mathbb{I}_d - ee^\top)$. □

C.2 Proof of Theorem 2

Theorem 2 (Exactness)**.** *For every $K \geq 1$ and every $\delta \geq 0$, the RASS verifier is exact for any feasible rank weights $\omega \in \mathcal{W}_K$ used by the verifier.*

Proof. If $\delta = 0$, the shared preparation in Algorithm 1 returns the first child, and $P = Q$, so the claim is immediate. Assume $\delta > 0$. Let S^{out} and $Z_\perp^{\mathrm{out}}$ denote the scalar and orthogonal components of the returned state, so that

$$Y = \mu_p + \sigma\big(S^{\mathrm{out}} e + Z_\perp^{\mathrm{out}}\big).$$

On the acceptance branch, $S^{\mathrm{out}} = \widehat{S}_J$ has density p_ω by Proposition 1 and is retained with probability $a_\omega(s) = 1 \wedge q_\delta(s)/p_\omega(s)$. Hence the accepted scalar submeasure has density

$$p_\omega(s) a_\omega(s) = \min\{q_\delta(s), p_\omega(s)\}.$$

Moreover, conditional on $(\widehat{S}_{1:K}, R)$, the selected orthogonal component $Z_{\perp,J}$ has law $\mathcal{N}(0, \mathbb{I}_d - ee^\top)$ and is independent of the acceptance coin. Thus the accepted joint submeasure factors as the above scalar submeasure times the orthogonal Gaussian law.

On rejection, the rejection probability is $\eta_\omega = 1 - \alpha_K(\delta; \omega)$. If $\eta_\omega > 0$, the residual scalar has density r_ω, so its scalar contribution is

$$\eta_\omega r_\omega(s) = \big(q_\delta(s) - p_\omega(s)\big)_+.$$

The orthogonal component $Z_{\perp,1}$ is independent of all projections, the rank draw, the acceptance coin, and the independent residual draw, and has law $\mathcal{N}(0, \mathbb{I}_d - ee^\top)$. If $\eta_\omega = 0$, this branch has zero probability. Therefore the total joint law of $(S^{\mathrm{out}}, Z_\perp^{\mathrm{out}})$ is the product of $\mathcal{N}(\delta, 1)$ and $\mathcal{N}(0, \mathbb{I}_d - ee^\top)$, since

$$\min\{q_\delta(s), p_\omega(s)\} + \big(q_\delta(s) - p_\omega(s)\big)_+ = q_\delta(s).$$

It follows that

$$Y \sim \mathcal{N}(\mu_p + \sigma\delta e, \sigma^2 \mathbb{I}_d) = \mathcal{N}(\mu_q, \sigma^2 \mathbb{I}_d).$$

At every visited node, the preceding argument shows that the next committed state has exactly the corresponding target transition law. If a child is accepted, the verifier returns that drafted child itself and its decision uses only the children of the current node. Hence, conditional on the accepted child, its already drafted descendants retain their original proposal-tree conditional law. Verification may therefore continue one level deeper with the same procedure. If a proposal is rejected, the exact residual draw is committed and the speculative round terminates. Induction over committed steps therefore gives the target Markov-chain law. □

C.3 Proof of Proposition 3

Proposition 3 (Optimal list acceptance)**.** *In the Gaussian setting of Section 2, the maximum probability of accepting a draft while preserving the target distribution is*

$$\alpha_K^{\mathrm{up}}(\delta) = 1 - \sup_{s\in\mathbb{R}} \left\{\Phi(s)^K - \Phi(s-\delta)\right\},$$

where Φ *denotes the standard Gaussian CDF.*

Proof. We first reduce the problem to one dimension. Recall that

$$P = \mathcal{N}(\mu_p, \sigma^2 I_d), \qquad Q = \mathcal{N}(\mu_q, \sigma^2 I_d), \qquad \delta = \frac{\|\mu_q - \mu_p\|}{\sigma}.$$

The case $\delta = 0$ is immediate, since then $P = Q$ and one may always return, say, the first draft. Hence suppose $\delta > 0$, and let

$$e = \frac{\mu_q - \mu_p}{\sigma\delta} = \frac{\mu_q - \mu_p}{\|\mu_q - \mu_p\|}.$$

For i.i.d. drafts $X_1, \dots, X_K \sim P$ and a target variable $Y \sim Q$, define the scalar coordinates

$$\widehat{S}_i := \frac{\langle X_i - \mu_p, e\rangle}{\sigma}, \qquad S := \frac{\langle Y - \mu_p, e\rangle}{\sigma}.$$

Then

$$\widehat{S}_1, \dots, \widehat{S}_K \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \qquad S \sim \mathcal{N}(\delta, 1).$$

Apply the probability integral transform

$$\widehat{U}_i := \Phi(\widehat{S}_i), \qquad V := \Phi(S).$$

Then $\widehat{U}_1, \dots, \widehat{U}_K$ are i.i.d. $\mathcal{U}(0,1)$, while V has CDF

$$F_\delta(a) := \Pr(V \le a) = \Pr(\Phi(S) \le a) \tag{14}$$

$$= \Pr(S \le \Phi^{-1}(a)) = \Pr\left(\frac{S-\delta}{\sigma} \le \frac{\Phi^{-1}(a) - \delta}{\sigma}\right) \tag{15}$$

$$= \Phi\left(\Phi^{-1}(a) - \delta\right), \qquad a \in (0,1), \tag{16}$$

Its density can be obtained directly by a likelihood-ratio argument. Indeed, for any Borel set $A \subseteq (0,1)$,

$$\begin{aligned}
\Pr(V \in A) &= \int_{\{s:\, \Phi(s)\in A\}} \phi(s-\delta)\, ds \\
&= \int_{\{s:\, \Phi(s)\in A\}} \frac{\phi(s-\delta)}{\phi(s)} \phi(s)\, ds \\
&= \int_{\{s:\, \Phi(s)\in A\}} \exp\left\{\delta s - \frac{\delta^2}{2}\right\} \phi(s)\, ds.
\end{aligned}$$

Since Φ pushes the standard Gaussian measure $\phi(s)\, ds$ forward to the uniform measure on $(0,1)$, this becomes

$$\Pr(V \in A) = \int_A \exp\left\{\delta\Phi^{-1}(a) - \frac{\delta^2}{2}\right\} da.$$

Hence V has density

$$h_\delta(a) = \exp\left\{\delta\Phi^{-1}(a) - \frac{\delta^2}{2}\right\}.$$

For $\delta > 0$, this density is strictly increasing because Φ^{-1} is strictly increasing. Let

$$E := \{V \in \{U_1, \dots, U_K\}\}$$

be the acceptance event. For any $a \in [0,1]$, acceptance with $V > a$ requires at least one draft above a. Hence

$$\Pr(E, V > a) \le \Pr\left(\max_{1\le i\le K} \widehat{U}_i > a\right) = 1 - a^K.$$

It follows that

$$\begin{aligned}1-\Pr(E) &\geq \Pr(E^c, V > a)\\ &= \Pr(V > a) - \Pr(E, V > a)\\ &\geq \big(1 - F_\delta(a)\big) - \big(1 - a^K\big)\\ &= a^K - F_\delta(a).\end{aligned}$$

Therefore every valid coupling satisfies

$$\Pr(E) \leq 1 - \max_{a\in[0,1]} \left\{a^K - F_\delta(a)\right\}.$$

Substituting $a = \Phi(s)$ gives

$$\Pr(E) \leq 1 - \sup_{s\in\mathbb{R}} \left\{\Phi(s)^K - \Phi(s-\delta)\right\}.$$

It remains to prove that this bound is attainable.

More generally, let ν be any probability measure on $[0,1]$ and consider K i.i.d. uniform drafts. For a Borel set $A \subseteq [0,1]$, write $|A|$ for its Lebesgue measure and define

$$d(\nu) := \sup_{A\subseteq[0,1] \text{ Borel}} \left\{\nu(A) - \left[1 - (1-|A|)^K\right]\right\}.$$

The term

$$1 - (1-|A|)^K$$

is precisely the probability that at least one of the K drafts lies in A. Thus, for any coupling of the list and a target $V \sim \nu$,

$$\nu(A) \leq \Pr(E^c) + \Pr(\{U_1, \ldots, U_K\} \cap A \neq \varnothing),$$

and consequently

$$\Pr(E^c) \geq d(\nu).$$

Hence the acceptance probability is at most $1 - d(\nu)$.

We now show that this general bound is sharp. Partition $[0,1]$ into n intervals $I_{n,1}, \ldots, I_{n,n}$ of length $1/n$. Consider the finite coupling problem between the vector of draft-bin labels

$$(J_1, \ldots, J_K), \qquad J_i = j \iff U_i \in I_{n,j},$$

and the target-bin label J, where $V \sim \nu$. Connect a draft-label vector $(j_1, \ldots, j_K)$ to a target bin j exactly when $j \in \{j_1, \ldots, j_K\}$. The finite max-flow/min-cut theorem implies that the largest mass that can be coupled along such admissible edges is

$$1 - d_n,$$

where

$$d_n = \max_{\substack{A\subseteq[0,1]\\ A \text{ a union of partition bins}}} \left\{\nu(A) - \left[1 - (1-|A|)^K\right]\right\}.$$

Since the maximization defining d_n is over a subclass of the Borel sets,

$$d_n \leq d(\nu).$$

Thus there exists a coupling of the bin labels under which, with probability at least $1 - d(\nu)$, the target bin appears among the draft bins. Complete any unmatched source and target mass arbitrarily, for example by coupling the normalized residual marginals independently, so that the full source and target marginals remain unchanged.

Lift this finite coupling from bin labels to points as follows. Conditionally on their labels, draw each U_i uniformly inside its assigned bin, independently across i, and draw V according to ν conditioned on its assigned target bin. The resulting coupling π_n has exactly the marginal $\mathrm{Unif}[0,1]^{\otimes K}$ for the draft list and marginal ν for V. Moreover, whenever the target bin is one of the draft bins,

$$\min_{1\leq i\leq K} |V - U_i| \leq \frac{1}{n}.$$

Hence

$$\pi_n\left(\min_{1\le i\le K}|V-U_i|\le\frac{1}{n}\right)\ge 1-d(\nu).$$

Since $[0,1]^{K+1}$ is compact, the sequence (π_n) has a weakly convergent subsequence, say $\pi_{n_j}\Rightarrow\pi$. The limiting coupling π preserves both marginals. For $\varepsilon>0$, let

$$R_\varepsilon:=\left\{(u_1,\dots,u_K,v):\min_i|v-u_i|\le\varepsilon\right\}.$$

This set is closed. For all sufficiently large j, $1/n_j\le\varepsilon$, so

$$\pi_{n_j}(R_\varepsilon)\ge 1-d(\nu).$$

By the closed-set part of the Portmanteau theorem,

$$\pi(R_\varepsilon)\ge\limsup_{j\to\infty}\pi_{n_j}(R_\varepsilon)\ge 1-d(\nu).$$

Since

$$R_0:=\{v=u_i\text{ for some }i\}=\bigcap_{m=1}^{\infty}R_{1/m},$$

continuity from above yields

$$\pi(R_0)\ge 1-d(\nu).$$

Together with the preceding upper bound, this proves

$$\sup_\pi\Pr_\pi\{V\in\{U_1,\dots,U_K\}\}=1-d(\nu),$$

and the supremum is attained.

It remains to specialize this characterization to the Gaussian target law ν_δ. Since h_δ is increasing, among all measurable sets $A\subseteq[0,1]$ of a fixed Lebesgue measure m, the target mass $\nu_\delta(A)$ is maximized by the upper interval

$$T_m=[1-m,1].$$

Indeed, $A\setminus T_m$ and $T_m\setminus A$ have equal Lebesgue measure, while every point of $A\setminus T_m$ lies to the left of every point of $T_m\setminus A$. Therefore

$$\nu_\delta(A)=\int_A h_\delta(a)\,da\le\int_{T_m}h_\delta(a)\,da=1-F_\delta(1-m).$$

The draft-availability term depends on A only through $m=|A|$. Consequently,

$$\begin{aligned}d(\nu_\delta)&=\max_{m\in[0,1]}\left\{1-F_\delta(1-m)-\left[1-(1-m)^K\right]\right\}\\&=\max_{a\in[0,1]}\left\{a^K-F_\delta(a)\right\}\\&=\sup_{s\in\mathbb{R}}\left\{\Phi(s)^K-\Phi(s-\delta)\right\},\end{aligned}$$

where in the second line we set $a=1-m$, and in the last line $a=\Phi(s)$.

Therefore the maximal achievable acceptance probability is

$$\alpha_K^{\mathrm{up}}(\delta)=1-\sup_{s\in\mathbb{R}}\left\{\Phi(s)^K-\Phi(s-\delta)\right\}.$$

The weak-limit construction above supplies a coupling attaining this value, completing the proof. □

C.4 Proof of Theorem 4

Theorem 4 (Acceptance probability). *Fix $\delta > 0$ and $\omega \in \mathcal{W}_K$. We take*

$$\mathcal{A}_\omega := \{s \in \mathbb{R} : q_\delta(s) > p_\omega(s)\} = \{s : \rho(s) > b_\omega(\Phi(s))\},$$

and let $\{(a_j, b_j)\}_j$ be its disjoint connected components. Define $C_\omega(a) := \sum_{r=1}^K \omega_r I_a(r, K+1-r)$. Then the acceptance probability satisfies

$$\alpha_K(\delta; \omega) = 1 - \sum_j \big[H_\omega(b_j) - H_\omega(a_j)\big] \tag{17}$$

where $H_\omega(s) := \Phi(s-\delta) - C_\omega(\Phi(s))$ satisfies $H'_\omega = q_\delta - p_\omega$, so each term of the sum is the residual mass $\int_{a_j}^{b_j} (q_\delta - p_\omega)\, ds$ on one component of A_ω. Moreover, we have $\alpha_K(\delta; \omega^{\mathrm{unif}}) = 2\overline{\Phi}(\delta/2)$.

Proof. Since p_ω and q_δ are probability densities,

$$\begin{aligned}\alpha_K(\delta; \omega) &= \int_{\mathbb{R}} \min\{q_\delta(s), p_\omega(s)\}\, \mathrm{d}s \\ &= 1 - \int_{\mathbb{R}} \big(q_\delta(s) - p_\omega(s)\big)_+\, \mathrm{d}s.\end{aligned}$$

By definition,

$$\mathcal{A}_\omega = \{s \in \mathbb{R} : q_\delta(s) > p_\omega(s)\},$$

and hence

$$\alpha_K(\delta; \omega) = 1 - \int_{\mathcal{A}_\omega} \big(q_\delta(s) - p_\omega(s)\big)\, \mathrm{d}s.$$

Recall that

$$p_\omega(s) = \phi(s) b_\omega(\Phi(s)), \qquad b_\omega(a) = \sum_{r=1}^K \omega_r f_r(a),$$

and

$$C_\omega(a) = \sum_{r=1}^K \omega_r I_a(r, K+1-r).$$

Since $I_a(r, K+1-r)$ is the CDF corresponding to the density $f_r(a)$,

$$\frac{\mathrm{d}}{\mathrm{d}a} C_\omega(a) = b_\omega(a).$$

Therefore

$$\frac{\mathrm{d}}{\mathrm{d}s} C_\omega(\Phi(s)) = b_\omega(\Phi(s))\phi(s) = p_\omega(s).$$

Moreover,

$$\frac{\mathrm{d}}{\mathrm{d}s} \Phi(s-\delta) = \phi(s-\delta) = q_\delta(s).$$

Thus, for

$$H_\omega(s) := \Phi(s-\delta) - C_\omega(\Phi(s)),$$

we have

$$H'_\omega(s) = q_\delta(s) - p_\omega(s).$$

The set $\mathcal{A}_\omega$ is open, since $q_\delta - p_\omega$ is continuous, and therefore it can be written as a disjoint union of open intervals,

$$\mathcal{A}_\omega = \bigcup_j (a_j, b_j).$$

Here an endpoint may be infinite, in which case $H_\omega(\pm\infty)$ is understood as the corresponding limit. Consequently,

$$\begin{aligned}\int_{\mathcal{A}_\omega} \big(q_\delta(s) - p_\omega(s)\big)\,\mathrm{d}s &= \sum_j \int_{a_j}^{b_j} \big(q_\delta(s) - p_\omega(s)\big)\,\mathrm{d}s \\ &= \sum_j \big[H_\omega(b_j) - H_\omega(a_j)\big].\end{aligned}$$

Every term in the sum is nonnegative, since $q_\delta > p_\omega$ on (a_j, b_j), and the sum is bounded by one. Hence

$$\alpha_K(\delta;\omega) = 1 - \sum_j \big[H_\omega(b_j) - H_\omega(a_j)\big],$$

which proves (17). We now consider the two special choices of rank weights. For uniform weights,

$$\omega^{\mathrm{unif}} = \left(\frac{1}{K}, \ldots, \frac{1}{K}\right).$$

The equally weighted mixture of the K order-statistic densities is uniform on $(0,1)$, so

$$b_{\omega^{\mathrm{unif}}}(a) = 1, \qquad C_{\omega^{\mathrm{unif}}}(a) = a.$$

Thus

$$p_{\omega^{\mathrm{unif}}}(s) = \phi(s).$$

Since

$$\rho(s) := \frac{q_\delta(s)}{\phi(s)} = \exp\left(\delta s - \frac{\delta^2}{2}\right)$$

is strictly increasing for $\delta > 0$, we have

$$q_\delta(s) > \phi(s) \quad \Longleftrightarrow \quad s > \frac{\delta}{2}.$$

Therefore

$$\mathcal{A}_{\omega^{\mathrm{unif}}} = \left(\frac{\delta}{2}, \infty\right).$$

Furthermore,

$$H_{\omega^{\mathrm{unif}}}(s) = \Phi(s-\delta) - \Phi(s),$$

and hence

$$H_{\omega^{\mathrm{unif}}}(\infty) = 0,$$

while

$$\begin{aligned}H_{\omega^{\mathrm{unif}}}(\delta/2) &= \Phi(-\delta/2) - \Phi(\delta/2) \\ &= 1 - 2\Phi(\delta/2).\end{aligned}$$

It follows that the rejection probability is

$$H_{\omega^{\mathrm{unif}}}(\infty) - H_{\omega^{\mathrm{unif}}}(\delta/2) = 2\Phi(\delta/2) - 1,$$

and therefore

$$\begin{aligned}\alpha_K(\delta;\omega^{\mathrm{unif}}) &= 1 - \big(2\Phi(\delta/2) - 1\big) \\ &= 2\big(1 - \Phi(\delta/2)\big) \\ &= 2\overline{\Phi}(\delta/2).\end{aligned}$$

Finally, consider the top-rank weights

$$\omega^{\max} = (0, \ldots, 0, 1).$$

The largest of K i.i.d. uniform random variables has CDF a^K and density Ka^{K-1}, so

$$C_{\omega^{\max}}(a) = a^K, \qquad b_{\omega^{\max}}(a) = Ka^{K-1}.$$

Consequently,

$$p_{\omega^{\max}}(s) = K\phi(s)\Phi(s)^{K-1}.$$

Every finite boundary point of $\mathcal{A}_{\omega^{\max}}$ satisfies $q_\delta(s) = p_{\omega^{\max}}(s)$. Dividing by $\phi(s) > 0$ gives

$$\frac{\phi(s-\delta)}{\phi(s)} = K\Phi(s)^{K-1},$$

or equivalently

$$\exp\left(\delta s - \frac{\delta^2}{2}\right) = K\Phi(s)^{K-1}.$$

Suppose now that $K > 1$. To characterize the solutions of this equation, define

$$L_K(s) := \log K + (K-1)\log\Phi(s) - \delta s + \frac{\delta^2}{2}.$$

Then

$$L_K(s) = \log\frac{p_{\omega^{\max}}(s)}{q_\delta(s)},$$

so the boundary equation is precisely

$$L_K(s) = 0.$$

Differentiating,

$$L_K'(s) = (K-1)\frac{\phi(s)}{\Phi(s)} - \delta.$$

The inverse Mills ratio $\phi(s)/\Phi(s)$ is strictly decreasing, with

$$\frac{\phi(s)}{\Phi(s)} \longrightarrow \infty \quad \text{as } s \to -\infty, \qquad \frac{\phi(s)}{\Phi(s)} \longrightarrow 0 \quad \text{as } s \to \infty.$$

Hence L_K' vanishes at exactly one point. Thus L_K is strictly increasing up to its unique maximizer and strictly decreasing afterwards.

Moreover,

$$\lim_{s\to\infty} L_K(s) = -\infty.$$

Also, using the standard Gaussian left-tail asymptotics $\log\Phi(s) = -s^2/2 + o(s^2)$ as $s \to -\infty$, and since $K > 1$,

$$\lim_{s\to-\infty} L_K(s) = -\infty.$$

The maximum of L_K must be strictly positive. Indeed, if $L_K(s) \le 0$ for every s, then

$$p_{\omega^{\max}}(s) \le q_\delta(s) \qquad \text{for all } s.$$

Since both sides are probability densities with total mass one, this would force equality almost everywhere. But $p_{\omega^{\max}}$ and q_δ are distinct for $K > 1$ and $\delta > 0$, a contradiction.

Therefore L_K crosses zero exactly twice. Denote the two roots by

$$s_- < s_+.$$

Since L_K is positive between its roots and negative outside them,

$$p_{\omega^{\max}}(s) > q_\delta(s) \quad \Longleftrightarrow \quad s \in (s_-, s_+).$$

Equivalently,

$$\mathcal{A}_{\omega^{\max}} = (-\infty, s_-) \cup (s_+, \infty).$$

This completes the proof. □

The acceptance guarantee provided by Theorem 4 has the similar form as the D-GRS guarantee by Bullo et al. (2026, Theorem 2), i.e.,

$$1 - G_{K+1} = \Phi(\ln\lambda_K - \delta/2) + \lambda_K\overline{\Phi}(\ln\lambda_K + \delta/2),$$

with $\{\lambda_j\}_{j=0}^K$ a sequence of scalar levels defined as in (Bullo et al., 2026), and neither requires quantities dependent on d. The two guarantees are explicitly compared in Figure 3.

D IMPLEMENTING RASS

We describe how to compute the rank weights and sample the residual distribution in lines 5 and 11 of Algorithm 1. We use the scalar densities q_δ and p_ω defined in Sections 2 and 3.

D.1 RANK WEIGHTS

For fixed (δ, K), the optimization in (12) chooses weights $\omega \in \mathcal{W}_K$ to maximize the overlap between the selected-proposal density p_ω and the target density q_δ. The following property allows us to formulate its quadrature approximation as a linear program.

Proposition 5. *For fixed (δ, K), the acceptance probability $\alpha_K(\delta;\omega)$ is concave in ω on $\mathcal{W}_K$.*

Proof. By Proposition 1, $p_\omega(s)$ is linear in ω for each s. Thus, $\min\{q_\delta(s), p_\omega(s)\}$ is concave in ω. Integrating over s preserves concavity. □

Evaluating the overlap through its closed-form expression requires locating the boundaries of $\{s : q_\delta(s) > p_\omega(s)\}$, which depend on ω. Quadrature instead evaluates both densities at each grid point, so the optimization does not require solving $q_\delta(s) = p_\omega(s)$ as the weights change.

Linear program. Let $s_1 < \cdots < s_M$ be the quadrature grid points and $\nu_1, \ldots, \nu_M > 0$ the corresponding quadrature weights. Using the order-statistic densities f_r from Proposition 1, define

$$\mathsf{B}_r(s_i) := \phi(s_i) f_r(\Phi(s_i)), \qquad p_\omega(s_i) = \sum_{r=1}^{K} \omega_r \mathsf{B}_r(s_i).$$

The quadrature approximation to the acceptance probability is

$$\widetilde{\alpha}_K(\delta;\omega) := \sum_{i=1}^{M} \nu_i \min\left\{ q_\delta(s_i), \sum_{r=1}^{K} \omega_r \mathsf{B}_r(s_i) \right\}.$$

Maximizing this objective over $\mathcal{W}_K$ is equivalent to solving

$$\begin{aligned} \underset{\omega \in \mathcal{W}_K,\, z \in \mathbb{R}^M}{\text{maximize}} \quad & \sum_{i=1}^{M} \nu_i z_i \\ \text{subject to} \quad & 0 \le z_i \le q_\delta(s_i), \\ & z_i \le \sum_{r=1}^{K} \omega_r \mathsf{B}_r(s_i), \qquad i = 1, \ldots, M. \end{aligned} \tag{18}$$

Since every ν_i is positive, each z_i equals the smaller of its two upper bounds at an optimum. Hence (18) represents the quadrature objective exactly. Only the replacement of the integral by quadrature introduces an approximation.

The program has $K + M$ variables and $2M$ upper-bound inequalities, in addition to nonnegativity and the simplex constraint. Its size is independent of the state dimension d, and its coefficients depend only on (δ, K) and the quadrature rule.

Approximation quality. Let $\widetilde{\omega}$ and $\omega^\star$ solve (18) and (12), respectively. Define the uniform quadrature error

$$\epsilon_M := \sup_{\omega \in \mathcal{W}_K} \left| \widetilde{\alpha}_K(\delta;\omega) - \alpha_K(\delta;\omega) \right|.$$

Optimality of $\widetilde{\omega}$ for the quadrature objective gives

$$\alpha_K(\delta;\widetilde{\omega}) \ge \widetilde{\alpha}_K(\delta;\widetilde{\omega}) - \epsilon_M \ge \widetilde{\alpha}_K(\delta;\omega^\star) - \epsilon_M \ge \alpha_K(\delta;\omega^\star) - 2\epsilon_M.$$

Thus, the loss in acceptance probability is at most $2\epsilon_M$. This bound accounts for both integration error within the grid and any omitted tail mass.

Algorithm 3 RANKWEIGHTS(δ, K): finite-grid rank optimization

Require: Normalized mean gap $\delta \geq 0$; integer $K \geq 1$
Ensure: Rank weights $\omega \in \mathcal{W}_K$

$\widetilde{\delta} \leftarrow \mathrm{round}_3(\delta)$ ▷ Three significant digits.
$\omega^{\mathrm{unif}} \leftarrow (1/K, \ldots, 1/K)$; $\omega^{\max} \leftarrow (0, \ldots, 0, 1)$
if $K = 1$ **or** $\widetilde{\delta} < 10^{-7}$ **then**
  return ω^{unif}
end if
if $\widetilde{\delta} > 12$ **then**
  return $\omega^{\max}$ ▷ Optimal for large δ; here $\alpha_K < 10^{-8}$.
end if
$(s_1, \ldots, s_M) \leftarrow \mathrm{sort_unique}\big(\mathrm{linspace}(-9, 9, 361) \cup \mathrm{linspace}(\widetilde{\delta} - 9, \widetilde{\delta} + 9, 361)\big)$
$\nu_1 \leftarrow \frac{1}{2}(s_2 - s_1)$; $\nu_M \leftarrow \frac{1}{2}(s_M - s_{M-1})$; $\nu_i \leftarrow \frac{1}{2}(s_{i+1} - s_{i-1})$, $i = 2, \ldots, M-1$
$q_i \leftarrow \phi(s_i - \widetilde{\delta})$; $\mathsf{B}_r(s_i) \leftarrow \phi(s_i) f_r(\Phi(s_i))$, $i = 1, \ldots, M$, $r = 1, \ldots, K$
$\mathcal{W}_{\mathrm{cand}} \leftarrow \{\omega^{\mathrm{unif}}, \omega^{\max}\}$
if the program (18) is solved with weights $\widehat{\omega}$ **then**
  $v_r \leftarrow \max\{\widehat{\omega}_r, 0\}$, $r = 1, \ldots, K$ ▷ Remove solver round-off.
  $\mathcal{W}_{\mathrm{cand}} \leftarrow \mathcal{W}_{\mathrm{cand}} \cup \{v / \sum_r v_r\}$
end if
return $\arg \max_{w \in \mathcal{W}_{\mathrm{cand}}} \sum_{i=1}^{M} \nu_i \min\Big\{q_i, \sum_{r=1}^{K} \mathsf{B}_r(s_i) w_r\Big\}$

We assess the numerical loss by comparing the weights returned by Algorithm 3 with those obtained using a finer quadrature grid of 6001 points per interval. We evaluate the acceptance probabilities of both sets of weights using Theorem 4. For $K \in \{2, 3, 4, 6, 8\}$ and 14 values of $\delta \in [0.01, 6]$, the largest observed loss is 2.1×10^{-5}. By comparison, the gap between optimized rank mixtures and the optimal list-coupling acceptance probability α_K^{up} of Proposition 3 reaches 3.7×10^{-2} (Figure 3). These results indicate that quadrature error has a negligible effect on acceptance in the tested settings.

Implementation. Algorithm 3 implements the routine RANKWEIGHTS(δ, K) used in line 5 of Algorithm 1. To reuse weights across transitions with similar mean displacements, we round δ to three *significant* digits, obtaining $\widetilde{\delta}$, and cache the weights by $(\widetilde{\delta}, K)$. For $\delta > 0$, this gives a relative rounding error of at most 5×10^{-3}, including for small displacements.

The quadrature grid is the sorted union, with duplicates removed, of two equally spaced grids, each containing 361 points: one on $[-9, 9]$ and the other on $[\widetilde{\delta} - 9, \widetilde{\delta} + 9]$. These intervals extend nine standard deviations around the proposal and target means used in the optimization. On the merged grid, the composite trapezoidal weights are

$$\nu_1 = \frac{s_2 - s_1}{2}, \qquad \nu_i = \frac{s_{i+1} - s_{i-1}}{2} \quad (2 \leq i \leq M-1), \qquad \nu_M = \frac{s_M - s_{M-1}}{2}.$$

To improve numerical stability in the tails, we compute $\mathsf{B}_r(s_i)$ through its logarithm, using $\log \Phi(s_i)$ and $\log \Phi(-s_i)$ for the beta-density factors. We solve (18) using the dual simplex method of HiGHS (Huangfu & Hall, 2018).

Optimized weights. Figure 6 shows the numerically optimized weights returned by Algorithm 3 for $K = 4$ and $K = 8$. As $\delta \to 0$, the weights approach the uniform distribution, recovering the proposal density ϕ. Increasing δ shifts weight toward higher ranks, with nearly all mass assigned to the highest rank for $\delta \gtrsim 2$ when $K = 4$ and $\delta \gtrsim 2.5$ when $K = 8$.

The irregular variation of intermediate-rank weights at small δ can arise from nonuniqueness of the linear-program solution. When several solutions are optimal, the simplex solver selects a vertex of the optimal face, and this choice may change as δ varies. Such changes need not indicate a change in acceptance: for a fixed (δ, K), all solutions on the optimal face yield the same quadrature objective.

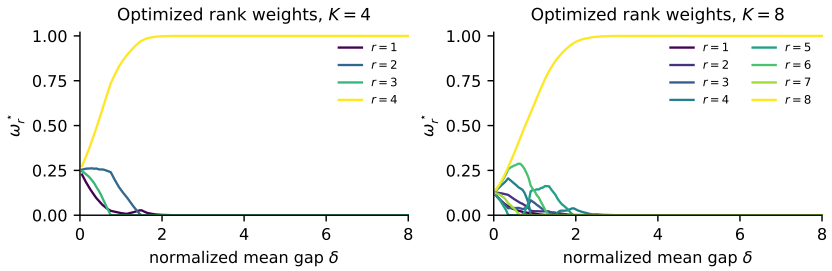


Figure 6: Numerically optimized rank weights returned by Algorithm 3 for $K = 4$ and $K = 8$. The weights approach uniformity as $\delta \to 0$ and concentrate on the highest rank for $\delta \gtrsim 2$ ($K = 4$) and $\delta \gtrsim 2.5$ ($K = 8$). Irregular changes in intermediate-rank weights at small δ can reflect transitions between distinct optimal solutions, which yield the same quadrature objective at a given δ.

D.2 Residual Sampling

On rejection, line 11 of Algorithm 1 samples the scalar projection from

$$r_\omega(s) = \frac{(q_\delta(s) - p_\omega(s))_+}{\eta_\omega}, \qquad \eta_\omega = 1 - \alpha_K(\delta; \omega).$$

This branch is reached only when $\eta_\omega > 0$. Recall from Theorem 4 that the residual is positive on

$$A_\omega = \{s : q_\delta(s) > p_\omega(s)\} = \bigcup_j (a_j, b_j),$$

where the disjoint intervals are ordered from left to right. The function

$$H_\omega(s) = \Phi(s - \delta) - C_\omega(\Phi(s))$$

satisfies $H'_\omega = q_\delta - p_\omega$ and $H_\omega(\pm\infty) = 0$. Hence, the unnormalized residual mass on interval (a_j, b_j) is

$$m_j = H_\omega(b_j) - H_\omega(a_j), \qquad \sum_j m_j = \eta_\omega.$$

The beta CDFs defining C_ω have integer parameters, giving the polynomial expression

$$C_\omega(a) = \sum_{r=1}^{K} \omega_r \sum_{\ell=r}^{K} \binom{K}{\ell} a^\ell (1-a)^{K-\ell}.$$

We can therefore compute interval masses without numerical integration. In the implementation, we evaluate H_ω as

$$H_\omega(s) = -[\Phi(s) - \Phi(s - \delta)] - [C_\omega(\Phi(s)) - \Phi(s)].$$

Locating the residual intervals. Every finite endpoint satisfies

$$q_\delta(s) = p_\omega(s), \qquad \text{equivalently} \qquad \exp(\delta s - \delta^2/2) = b_\omega(\Phi(s)).$$

The residual need not be confined to a single right tail. For example, with $K = 2$, $\omega = (0, 1)$, and $\delta = 1$,

$$p_\omega(s) = 2\phi(s)\Phi(s), \qquad H_\omega(s) = \Phi(s - 1) - \Phi(s)^2,$$

and

$$A_\omega \approx (-\infty, -1.486) \cup (1.028, \infty).$$

The left interval contains approximately 0.8% of the residual mass, so sampling only from the right interval would not preserve the target law. To locate the intervals numerically, we evaluate $\log q_\delta - \log p_\omega$ on a grid of 513 points and refine each detected sign change using Brent's method. This is a numerical search: a finite grid alone does not guarantee that every crossing is detected. At a simple crossing, an endpoint error of size ϵ changes the corresponding interval mass by $O(\epsilon^2)$, since $q_\delta - p_\omega$ vanishes at the endpoint. We cache the interval endpoints and masses using the unrounded pair (δ, ω).

Inverse-CDF sampler. Draw $U \sim \mathcal{U}(0,1)$ and set $v = U\eta_\omega$. First, select the interval j satisfying

$$\sum_{i<j} m_i < v < \sum_{i\le j} m_i.$$

Equality at an interval boundary has probability zero and may be resolved arbitrarily. Then solve

$$H_\omega(S) - H_\omega(a_j) = v - \sum_{i<j} m_i, \qquad S \in (a_j, b_j),$$

using Brent's method. Since $H'_\omega > 0$ on this interval, the solution is unique. With exact interval masses and inversion, this construction samples from r_ω. Each draw uses one uniform random variable and one scalar root solve, with no numerical integration or additional target-model evaluation.

Alternative rejection sampler. One can instead repeatedly draw $S \sim \mathcal{N}(\delta, 1)$ and accept with probability

$$\left(1 - \frac{p_\omega(S)}{q_\delta(S)}\right)_+.$$

The acceptance probability of each trial is η_ω, so this procedure requires $1/\eta_\omega$ trials on average. Conditional on needing a correction, this can be expensive when $\alpha_K(\delta;\omega)$ is close to one. The inverse-CDF method avoids these repeated trials: after computing or retrieving the residual intervals, it performs one root solve per draw.

Orthogonal component. We reconstruct the corrected state as

$$Y = \mu_p + \sigma(Se + Z_{\perp,1}),$$

reusing the orthogonal component of the first child in drafting order. This component retains its Gaussian law independently of the scalar verification and residual draw. An independent sample from $\mathcal{N}(0, I_d - ee^\top)$ can be used instead, with the same output law.

D.3 RASS Computational Overhead

RASS introduces two scalar computations: selecting the rank weights and sampling the residual after rejection. The weights can be cached, while the residual admits either inverse-CDF or rejection sampling. We benchmark these computations separately and measure their combined contribution to verification. With cached weights and rejection sampling, the mean verification time of RASS is within 1.5% of D-GRS for every tested branching factor.

Benchmark setup. We measure scalar verification costs on synthetic inputs, using one core of an Intel Core i9-10980XE and no diffusion model. We consider $K \in \{2, 3, 4, 6, 8\}$ and normalized mean displacements $\delta \in \{0.01, 0.05, 0.1, 0.25, 0.5, 1, 2, 4, 8\}$. For each pair, we set $\mu_p = 0$, $\mu_q = \delta e_1$, and $\sigma = 1$ in dimension $d = 16$, and generate proposal children using a fixed random seed. Timings exclude the d-dimensional projections, which are common to RASS, D-GRS and RMC. Each reported value averages over the tested displacements and repetitions, with the half-width of a 95% confidence interval.

Rank-weight computation. Table 2 reports the cost of individual subroutines, using $1{,}000$ repetitions per (K, δ). Solving RankWeights takes 5.8–8.0 ms. As described above, this computation depends only on $(\widetilde{\delta}, K)$: once computed, the weights are reused whenever the same pair occurs. This cost is therefore incurred on a cache miss rather than at every verified node. Its contribution during sampling depends on how frequently new pairs are encountered.

Residual sampling. An inverse-CDF residual draw takes 0.30–0.66 ms. Its cost grows with K, consistent with the increasing number of terms in b_ω and C_ω. Rejection sampling has a different cost profile. Each trial succeeds with probability η_ω, so a residual draw requires $1/\eta_\omega$ trials on average. The measured cost is below $14\,\mu$s for $\delta \ge 2$, but reaches 4–44 ms at $\delta = 0.01$, where the residual mass is small. These conditional costs do not directly determine the average verification overhead.

	RANKWEIGHTS	Residual draw (μs)	
K	(ms)	inverse CDF	rejection sampler
2	5.78 $\pm$0.04	295 $\pm$3	645 $\pm$42
3	6.94 $\pm$0.04	314 $\pm$1	682 $\pm$43
4	7.22 $\pm$0.05	400 $\pm$3	1820 $\pm$117
6	7.42 $\pm$0.04	513 $\pm$2	3654 $\pm$224
8	7.95 $\pm$0.04	657 $\pm$4	6617 $\pm$426

Table 2: Costs of individual RASS subroutines on one CPU core. Entries average $1{,}000$ repetitions per (K, δ) over the nine displacements listed in the benchmark setup and report mean $\pm$ the half-width of a 95% confidence interval. RANKWEIGHTS measures one uncached linear-program solve. Residual timings measure one correction draw, conditional on requiring a correction; they are not weighted by the probability of rejection.

			RASS	
K	RMC	D-GRS	inverse CDF	rejection sampler
2	12.55 $\pm$0.02	29.90 $\pm$0.20	117 $\pm$2	29.53 $\pm$0.30
3	12.60 $\pm$0.02	33.90 $\pm$0.21	121 $\pm$2	33.84 $\pm$0.42
4	12.51 $\pm$0.02	37.72 $\pm$0.21	129 $\pm$2	38.21 $\pm$0.55
6	12.73 $\pm$0.02	47.05 $\pm$0.24	159 $\pm$3	47.71 $\pm$0.67
8	12.58 $\pm$0.02	54.69 $\pm$0.25	168 $\pm$3	55.49 $\pm$1.36

Table 3: Scalar verification time per node (μs) on one CPU core, with rank weights already cached. Entries average $3{,}000$ calls with fresh children per (K, δ) over the nine tested displacements and report mean $\pm$ the half-width of a 95% confidence interval. Residual sampling is included only when verification rejects. Rank-weight computation and d-dimensional projections are excluded. RMC uses one child, whereas RASS and D-GRS use K.

A residual draw is needed only with probability η_ω. Thus, for $\eta_\omega > 0$, the expected number of residual rejection-sampling trials per verified node is

$$\Pr(\text{rejection})\,\mathbb{E}[\text{trials} \mid \text{rejection}] = \frac{1}{\eta_\omega}\eta_\omega = 1.$$

Consequently, the large number of trials required by an individual correction at small δ is offset, in expectation, by the rarity of such corrections. The time per trial can still vary with (δ, K).

Verification cost. Table 3 measures scalar verification with cached rank weights, using $3{,}000$ calls per (K, δ) and fresh children at each call. The timings include residual sampling whenever verification rejects the selected proposal. Averaged over the tested displacements, RMC takes approximately $12.6\,\mu$s per call and D-GRS takes 30–55 μs. RASS with rejection sampling also takes 30–55 μs, with mean times within 1.5% of D-GRS for every tested K. RASS with inverse-CDF sampling takes 117–168 μs.

These results show that the choice of residual sampler materially affects verification overhead. Both constructions sample the same residual law in exact arithmetic and therefore preserve the same acceptance probabilities and target law.

With rejection sampling and cached weights, RASS therefore adds little average scalar verification cost relative to D-GRS in this benchmark. Computing previously unseen rank weights remains an additional cost. The measured scalar computations are independent of the state dimension and run entirely on the CPU.

D.4 EXACTNESS UNDER NUMERICAL APPROXIMATION

Theorem 2 guarantees exactness for every feasible $\omega \in \mathcal{W}_K$, regardless of how the weights are obtained. Thus, quadrature and rounding in RANKWEIGHTS affect acceptance but do not invalidate the exactness guarantee. Neither does the choice among optimal solutions. The acceptance test and

residual sampler use the actual displacement δ and the returned weights ω, rather than the rounded displacement $\widetilde{\delta}$ used during optimization. Numerical errors may nevertheless arise when implementing these steps, through floating-point arithmetic, the finite-grid search for residual intervals, and root-solver tolerances. The exactness theorem applies to the ideal verifier. However, its numerical implementation is subject to these errors. Table 1 compares sample quality across methods using FID and CLIP scores, while Appendix E.3 provides visual comparisons. These results complement the theoretical guarantee but do not establish distributional exactness.

E Experimental details and additional results

E.1 Delayed-drift proposals and prefetching

We specify the proposal construction and reuse policy used within the draft-tree sampler of Section 2 and Algorithm 2. Delayed drift determines how a cached target evaluation is used to draft a tree. Prefetching determines which evaluation is retained for the next round. For simplicity, conditioning information, such as class labels or text prompts, is held fixed and suppressed below.

A general proposal template. Write the target mean in Eq. (3) as

$$m_k^q(x) = \mathcal{M}_k\big(x, h_k(x)\big), \tag{19}$$

where $h_k(x)$ is the expensive model field and $\mathcal{M}_k$ is an inexpensive reconstruction map incorporating the state and sampling schedule. The field may be a drift, score, denoiser output, or velocity. At the start of a round at Y_n, let $\bar{h} = h_{\tilde{n}}(\tilde{y})$ be an available evaluation, with $\tilde{n} \le n$. We hold this value fixed throughout drafting and use

$$m_k^p(x; \bar{h}) = \mathcal{M}_k(x, \bar{h}), \qquad P_k(\cdot \mid x, \bar{h}) = \mathcal{N}\big(m_k^p(x; \bar{h}), \sigma_k^2 I_d\big)\,. \tag{20}$$

Thus, at a parent u of depth $|u|$, drafting uses $k = n + |u|$ and $x = \widehat{Y}_u$. Each child is sampled according to (4), with the model field frozen: the state and schedule arguments of $\mathcal{M}_k$ remain those of the drafted node. Conditional on the cache, this construction needs no further model evaluation during drafting. If no evaluation is available initially, we obtain $\bar{h} = h_n(Y_n)$ with one target evaluation before drafting.

For the Euler-Maruyama parameterization in (3), take $h_k = b_{t_k}^q$ and $\mathcal{M}_k(x, z) = x + \gamma z$, giving $m_k^p(x; \bar{h}) = x + \gamma \bar{h}$. More generally, an affine velocity parameterization,

$$m_k^q(x) = a_k x + b_k v_k(x), \qquad m_k^p(x; \bar{v}) = a_k x + b_k \bar{v}, \tag{21}$$

uses known scalar coefficients a_k, b_k and a cached velocity $\bar{v}$. This retains the current coefficients and state-dependent term even when the velocity was evaluated at a different state or timestep.

Prefetching between rounds. Let $\mathcal{E} \subseteq V$ contain the nodes whose target evaluations are available from the round's parallel evaluation, including any exact evaluations reused from earlier rounds. For $v \in \mathcal{E}$, write

$$H_v = h_{n+|v|}(\widehat{Y}_v).$$

These values can be retained when the target means are evaluated, or recovered from those means when the reconstruction map is invertible in its second argument. After the round commits N_α transitions, using the convention of Algorithm 2, its next root is $y^+ = Y_{n^+}$ with $n^+ = n + N_\alpha$. No update is needed if $n^+ = N$.

Let u be the parent of the final verified transition and set $\mathcal{A} = C(u) \cap \mathcal{E}$. The nearest-node policy used in our experiments selects the next cache as follows:

1. If the round ends by accepting a leaf $v_\star \in \mathcal{E}$, retain its own evaluation:
$$\bar{h}^+ = H_{v_\star} = h_{n^+}(y^+).$$
2. If the round ends by rejection and $\mathcal{A} \neq \varnothing$, retain the evaluation at the closest evaluated candidate from that final sibling list:
$$v^\dagger \in \arg\min_{v \in \mathcal{A}} \|\widehat{Y}_v - y^+\|_2, \qquad \bar{h}^+ = H_{v^\dagger}.$$

These candidates are at timestep n^+, although their states generally differ from the residual output y^+.

3. Otherwise, retain the final parent's evaluation: $\bar{h}^+ = H_u = h_{n^+-1}(\widehat{Y}_u)$.

Ties may be resolved in a fixed order. A parent-only policy always uses the third choice. Disabling prefetching instead evaluates $h_{n^+}(y^+)$ before drafting each new round.

With internal-node-only evaluation, $\mathcal{E} = I(\mathcal{T})$ away from the terminal horizon, so accepted leaves and candidates at the last level are unavailable for reuse. Evaluating the leaves, $\mathcal{E} = I(\mathcal{V})$, makes their fields available in the same parallel phase, supporting the first two choices also at the last level. Before horizon truncation, this adds K^L rows for a full (K, L) tree and gives the full-tree budget $B = |V|$ used in the main experiments. The cache-selection operation itself requires no additional target evaluation.

Specialization to the experimental samplers. All the experiments use the linear interpolation

$$X_\varsigma = (1-\varsigma)X_0 + \varsigma Z, \qquad X_0 \sim q_{\mathrm{data}}, \quad Z \sim \mathcal{N}(0, I_d),$$

with independent X_0 and Z. This path connects data at $\varsigma = 0$ to Gaussian noise at $\varsigma = 1$, and its velocity is

$$v_\varsigma(x) = \mathbb{E}[Z - X_0 \mid X_\varsigma = x].$$

Sampling proceeds in decreasing ς. The Gaussian-mixture experiment evaluates this velocity analytically, the pixel-space experiments approximate it through a pretrained denoiser (Karras et al., 2022), and SD3.5 supplies a learned flow velocity.

Let ς_k denote the interpolation parameter at step k, different from the transition standard deviation σ_k, and set $\Delta\varsigma_k = \varsigma_{k+1} - \varsigma_k < 0$. On stochastic steps, the target mean has the affine form

$$m_k^q(x) = a_k x + b_k v_{\varsigma_k}(x),$$

where

$$a_k = 1 + \Delta\varsigma_k \frac{\epsilon^2}{1-\varsigma_k}, \qquad b_k = \Delta\varsigma_k(1+\epsilon^2), \qquad \sigma_k = \epsilon\sqrt{\frac{2\varsigma_k}{1-\varsigma_k}(-\Delta\varsigma_k)}.$$

The cached quantity is a previous velocity evaluation, $\bar{v} = v_{\tilde{\varsigma}}(\tilde{y})$. The delayed-drift proposal therefore uses

$$m_k^p(x; \bar{v}) = a_k x + b_k \bar{v},$$

holding $\bar{v}$ fixed during drafting while retaining the current state x and schedule coefficients a_k, b_k. All experiments use the prefetch policy described above to select $\bar{v}$ for the next round, with their respective schedules and conditioning.

E.2 Results with internal-node evaluation

Section 2 presesnts two options for the parallel evaluation phase. The main experiments evaluate the full tree, with budget $B = |\mathcal{V}|$. This section gives the results when the sampler evaluates only the internal nodes $\mathcal{I}(\mathcal{T})$, with budget $B = |\mathcal{I}|$.

Setup. Before horizon truncation, a full (K, L) tree has

$$|\mathcal{V}| = \sum_{\ell=0}^{L} K^\ell, \qquad |\mathcal{I}| = \sum_{\ell=0}^{L-1} K^\ell = |\mathcal{V}| - K^L.$$

Internal-node evaluation thus removes the K^L leaves from the batch. It also removes their target fields from the prefetch cache (Appendix E.1). If a round accepts a leaf or rejects at the last level, no evaluated candidate is available at the next root. The next round then drafts from the evaluation at the final parent. This evaluation is one step older and is at a different state. Internal-node evaluation thus decreases the batch size, but it can decrease the accuracy of the next proposal.

All other settings are the same as in Section 5. These settings include the target discretization, the delayed-drift proposal, the prefetch policy, the (K, L) configurations, and the number of samples. RASS and D-GRS use identical trees. RMC uses a chain matched to the same internal-node budget. We compare the two budget definitions at a fixed configuration (K, L) and at a fixed budget B.

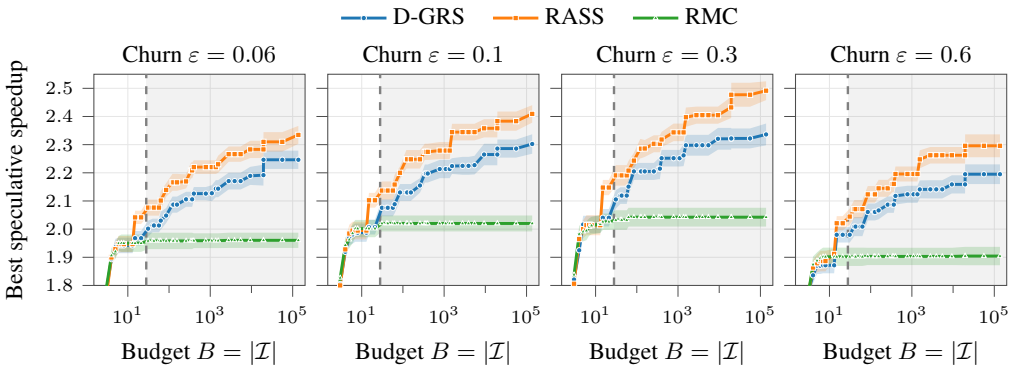


Figure 7: Best mean target-call speedup achievable within parallel budget $B = |\mathcal{I}|$ on the Gaussian-mixture target, for churn $\varepsilon \in \{0.06, 0.1, 0.3, 0.6\}$. Each point shows the best configuration (K, L) whose budget does not exceed B. Shading indicates 95% confidence intervals over 100 trajectories. The dashed line marks the number of diffusion steps, and the shaded region is budgets at or above it. For every churn value, RASS matches or exceeds D-GRS at every budget in the shaded region.

Gaussian mixture. Figures 7 and 8 give the results with $B = |\mathcal{I}|$. Figures 9 and 10 give the corresponding full-tree results. Both sets use churn $\varepsilon \in \{0.06, 0.1, 0.3, 0.6\}$ and the grid $K \in \{1, \dots, 7\}$, $L \in \{2, \dots, 7\}$. At a fixed configuration, internal-node evaluation decreases the speedup of the two tree methods, mainly for shallow trees. For RASS, the median decrease is 14.2% at $L = 2$, 7.5% at $L = 3$, and 3.0% at $L = 4$. For $L \geq 5$, the median decrease is less than 1%. A probable cause is that a shallow tree ends more rounds at its last level, where the leaf evaluations are necessary for an accurate cache. At a fixed budget, internal-node evaluation gives a higher speedup at moderate budgets. Within $B \leq 100$, the best mean speedup of RASS is 1.1% to 2.1% higher with $B = |\mathcal{I}|$ than with $B = |\mathcal{V}|$. Over the full grid, the highest speedups for the two definitions differ by at most 0.6%. With the internal-node budget, RASS gives a higher mean speedup than D-GRS in all 144 configurations with $K \geq 2$, by 0.6% to 6.7%. In 115 of these configurations, the 95% confidence intervals do not overlap. For every churn value, RASS matches or exceeds D-GRS at every budget at or above the number of diffusion steps. Its highest mean speedup is 3.9% to 6.6% above that of D-GRS. At $K = 1$, each node has one child, so no ranking is possible. The two rules then give the same speedup within the confidence intervals. RMC exceeds both tree methods at $L = 2$ for every K, and in some configurations with $L = 3$ and $K \leq 3$. In these configurations, the matched chain is longer than the tree and can commit more transitions per round.

Pixel-space diffusion. Figures 11 and 12 show the results for CIFAR-10 and FFHQ. The four configurations give $B = |\mathcal{I}| \in \{7, 31, 40, 85\}$, instead of $B = |\mathcal{V}| \in \{15, 63, 121, 341\}$. At a fixed configuration, internal-node evaluation decreases the speedup of the two tree methods. For RASS, the decrease is between 1.9% and 8.5% on CIFAR-10 and at most 7.1% on FFHQ. The largest decrease occurs at $(K, L) = (2, 3)$. A probable cause is that a shallow tree ends more rounds at its last level, where the leaf evaluations are necessary for an accurate cache. The speedup of RMC changes by less than 2%. Its matched chains have a length of at least seven, and few rounds reach the end of the chain. At a fixed budget, internal-node evaluation uses the parallel budget more efficiently. On CIFAR-10, RASS with $B = |\mathcal{I}| = 31$ is within 2.6% of RASS with $B = |\mathcal{V}| = 63$. On FFHQ, the difference between these two values is at most 0.5%. Thus, internal-node evaluation gives almost the same speedup with half the budget. On CIFAR-10, RASS with $B = |\mathcal{I}| = 31$ also exceeds the highest full-tree speedup of D-GRS, which requires $B = |\mathcal{V}| = 341$. This result holds for all three churn values. With the internal-node budget, RASS gives a higher mean speedup than D-GRS in every configuration. The relative gain is between 3.9% and 18.7% on CIFAR-10 and between 1.4% and 2.4% on FFHQ. On CIFAR-10, the gain increases with churn, as with the full-tree budget. It is between 3.9% and 6.6% at $\varepsilon = 0.1$ and between 10.3% and 18.7% at $\varepsilon = 0.6$. At the smallest budget, $B = |\mathcal{I}| = 7$, RMC gives a higher speedup than both tree methods in five of the six panels. A $(2, 3)$ tree can commit at most three transitions per round, but the matched chain can commit up to seven. For $B \geq 31$, both tree methods exceed RMC in every seting we tested.

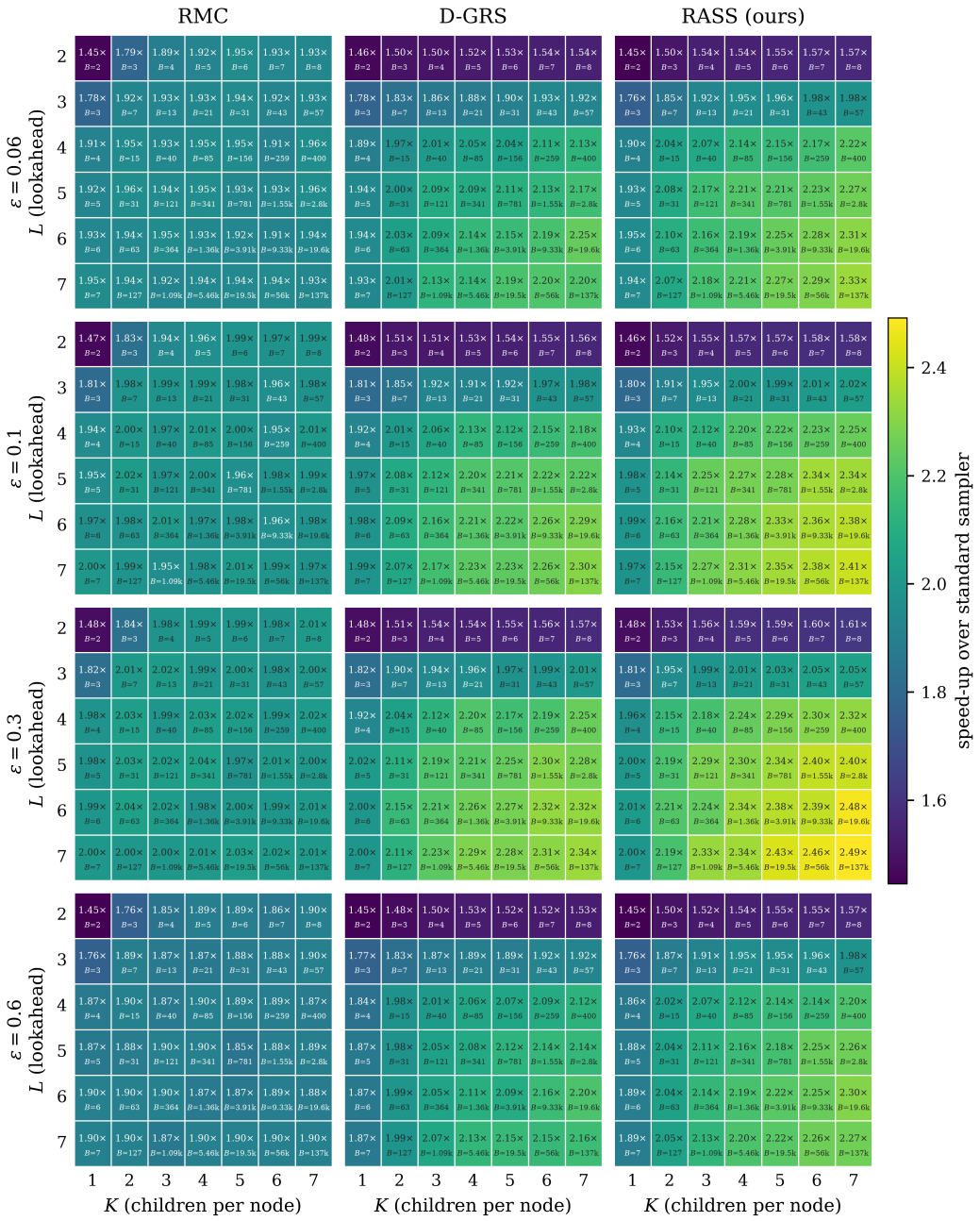


Figure 8: Mean target-call speedup against branching factor K and lookahead L, achieved using parallel budget $B = |\mathcal{I}|$.

Latent-space text-to-image. Figure 14 compares the best speedup within budget for SD3.5 under the two budget definitions. Figure 13 shows all configurations with $B = |\mathcal{I}|$. The grid gives $B = |\mathcal{I}| = K+1 \in \{3, \dots, 7\}$ for $L = 2$ and $B \in \{7, 13, 21, 31, 43\}$ for $L = 3$. At a fixed configuration, the effect of internal-node evaluation depends on the lookahead. For $L = 2$, it decreases the speedup of RASS and D-GRS by 2.1% to 3.7%. For $L = 3$, the change is at most 1.2% in magnitude. The mean speedups on SD3.5 are near $1.1\times$, so few rounds probably reach the third level. The leaf evaluations then seldom change the next proposal. The speedup of RMC changes by less than 1%. At a fixed budget, the two definitions give similar results. Within $B \leq 43$, the best mean speedup of RASS is $1.152\times$ with $B = |\mathcal{I}|$ and $1.151\times$ with $B = |\mathcal{V}|$. With the internal-node budget,

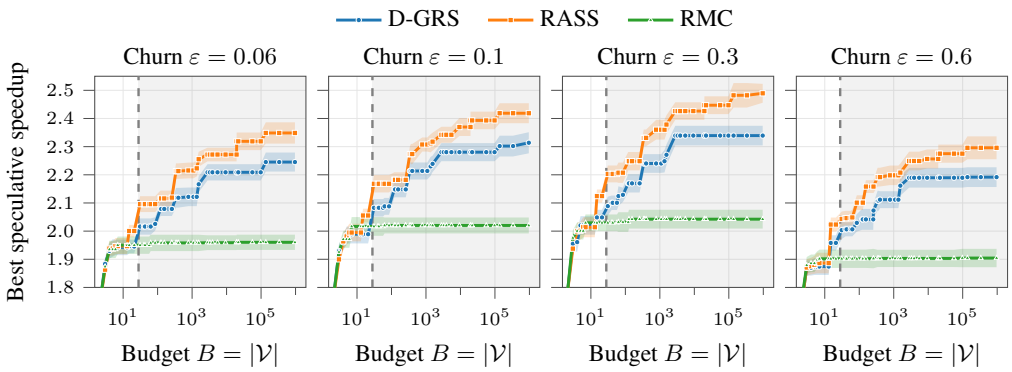


Figure 9: Best mean target-call speedup achievable within parallel budget $B = |\mathcal{V}|$ on the Gaussian-mixture target, for churn $\varepsilon \in \{0.06, 0.1, 0.3, 0.6\}$. Each point shows the best configuration (K, L) whose budget does not exceed B. Shading indicates 95% confidence intervals over 100 trajectories. The dashed line marks the number of diffusion steps, and the shaded region is budgets at or above it. For every churn value, RASS matches or exceeds D-GRS at every budget in the shaded region. Its highest speedup is 4.5% to 6.4% above D-GRS's, while RMC stays at about $1.9\times$–$2.0\times$.

RASS gives a higher mean speedup than D-GRS in all ten configurations, by 0.2% to 1.2%. The 95% confidence intervals overlap in all ten configurations, so no single difference is significant. RASS reaches $1.144\times$ at $B = 31$. This value exceeds the highest speedup of D-GRS, $1.141\times$ at $B = 43$, with 28% less budget. RMC stays between $1.093\times$ and $1.099\times$. It exceeds RASS only at $(K, L) = (2, 2)$, where $B = 3$ ($1.098\times$ against $1.094\times$, with overlapping confidence intervals).

Summary. Neither budget definition is better in all settings. Internal-node evaluation uses a smaller batch per round. Full-tree evaluation gives a more accurate cache for the next round. The better choice depends on the model, the lookahead, and the hardware cost of a larger batch. The order of the methods does not change. Under both definitions, RASS gives a higher mean speedup than D-GRS in every evaluated configuration.

E.3 Qualitatively Assessment of Exactness

Figures 15, 16, 17 show samples from the standard target sampler alongside RMC, D-GRS and RASS. Rows are chosen as follows: we draw four sample indices uniformly at random without replacement from the 500 images of each pixel-space run and from the 100 prompts of the SD3.5 run. We use the main-paper protocol with $\varepsilon = 0.3$, $K = 4$, $L = 4$ for CIFAR-10 and FFHQ, and $\varepsilon = 0.8$, $K = 6$, $L = 3$ for SD3.5.

All three speculative samplers are exact, so each produces a sample from the same distribution as the target sampler. On CIFAR-10 the class label is shared across methods, but the four columns are independent draws. On SD3.5 the prompt and the initial noise are shared, so images often share a composition but differ in detail, because each sampler consumes its randomness differently along the trajectory. Across all three datasets the speculative samples are indistinguishable in quality and content from those of the target sampler, and the text-conditioned samples are consistent with their prompts.

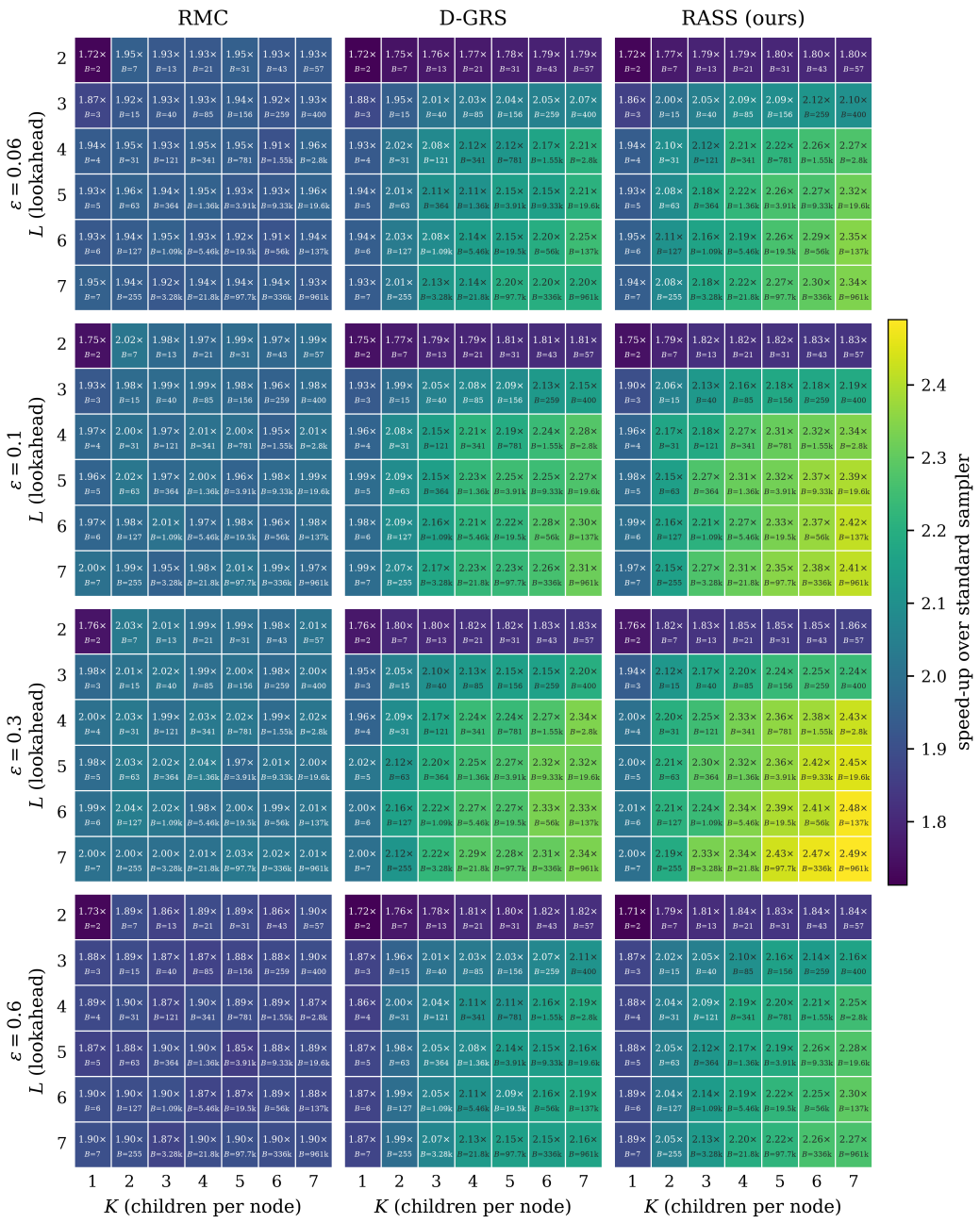


Figure 10: Mean target-call speedup against branching factor K and lookahead L, achieved using parallel budget $B = |\mathcal{V}|$.

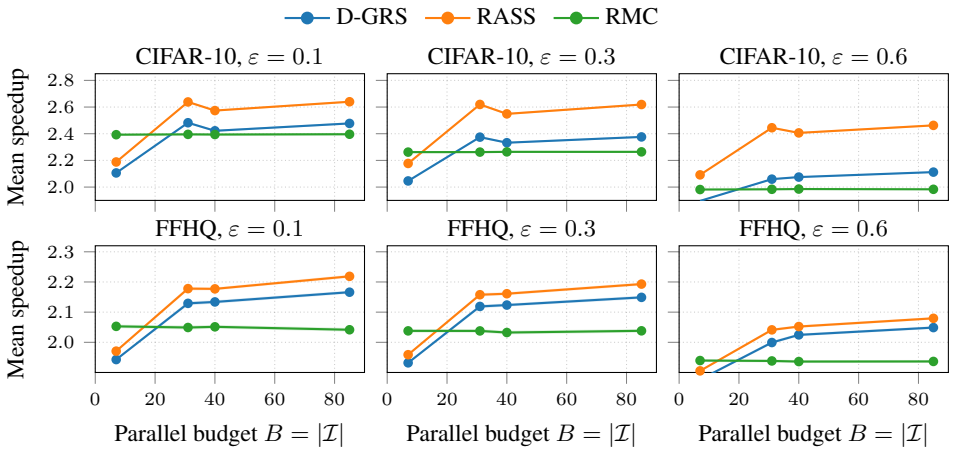


Figure 11: Mean target-call speedup on CIFAR-10 (top) and FFHQ (bottom) across parallel budgets and churn values. Curves average over 500 images per configuration, and shading shows 95% confidence intervals.

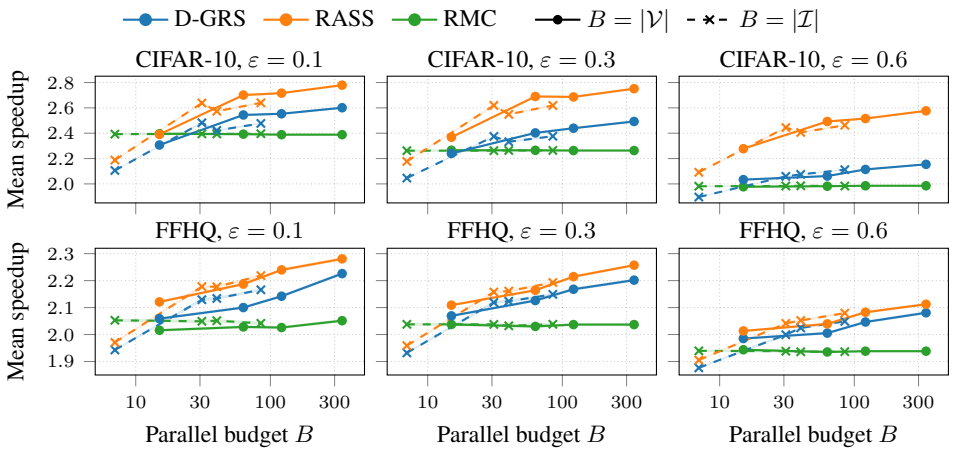


Figure 12: Pixel-space speedup under the two parallel-budget definitions. Mean target-call speedup on CIFAR-10 (top) and FFHQ (bottom) for churn $\varepsilon \in \{0.1, 0.3, 0.6\}$ and $(K, L) \in \{(2,3), (2,5), (3,4), (4,4)\}$. Solid lines evaluate the full tree, $B = |\mathcal{V}| \in \{15, 63, 121, 341\}$, as in Figure 4. Dashed lines evaluate only the internal nodes, $B = |\mathcal{I}| \in \{7, 31, 40, 85\}$, as in Figure11. RMC uses a chain matched to each budget. Curves average over 500 images per configuration, and shading shows 95% confidence intervals. RASS exceeds D-GRS in every configuration under both budget definitions.

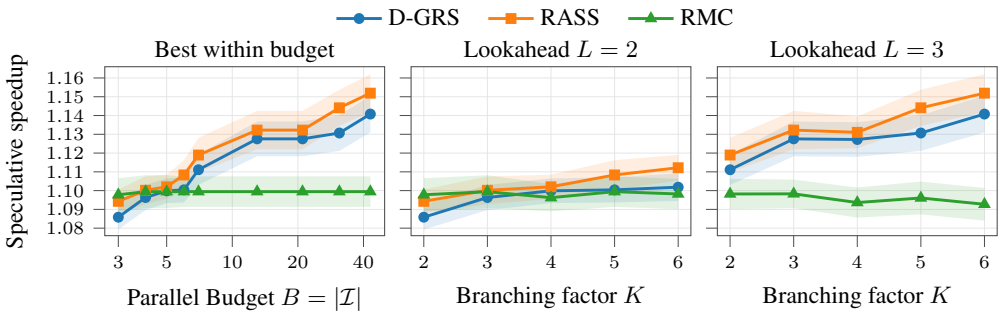


Figure 13: Mean target-call speedup for SD3.5-medium when verification evaluates only the internal nodes, $B = |\mathcal{I}|$. Left: best observed mean speedup within parallel budget B (logarithmic axis). Middle and right: speedup versus branching factor K at lookahead $L = 2$ and $L = 3$, respectively. Shading indicates 95% confidence intervals over 100 paired caption–noise inputs. RASS attains a higher mean speedup than D-GRS in every configuration.

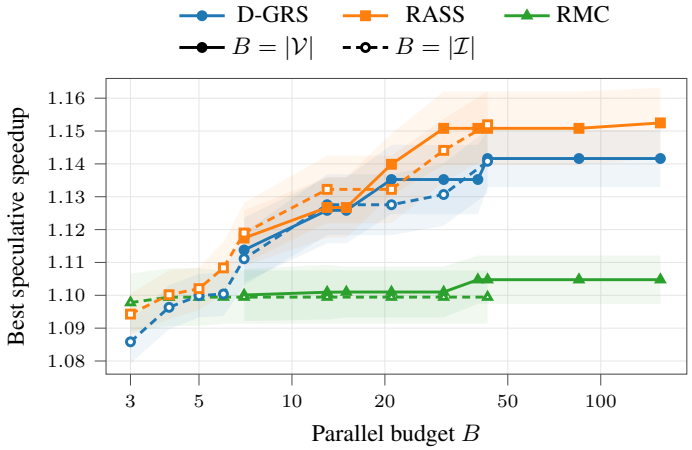


Figure 14: Best mean target-call speedup within parallel budget B for SD3.5-medium under two budget definitions. Solid lines with filled markers evaluate the full tree, $B = |\mathcal{V}|$, as in Figure **??**; dashed lines with open markers evaluate only the internal nodes, $B = |\mathcal{I}|$. Each point shows the best configuration (K, L) whose budget does not exceed B; the budget axis is logarithmic. Shading indicates 95% confidence intervals over 100 paired caption–noise inputs. Evaluating only internal nodes, RASS reaches its highest mean speedup, $1.152\times$, at $B = 43$ rather than $B = 156$.

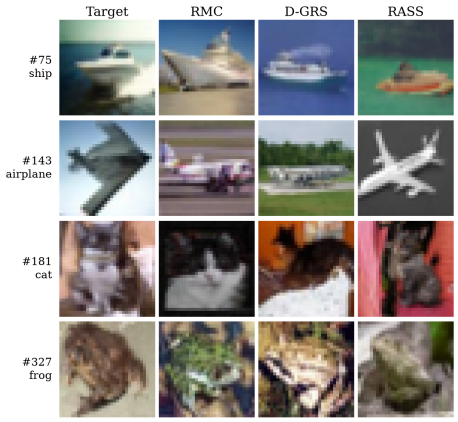


Figure 15: CIFAR-10 (32×32, class-conditional), $\varepsilon = 0.3$, $K = 4$, $L = 4$.



Figure 16: FFHQ (64×64, unconditional), $\varepsilon = 0.3$, $K = 4$, $L = 4$.

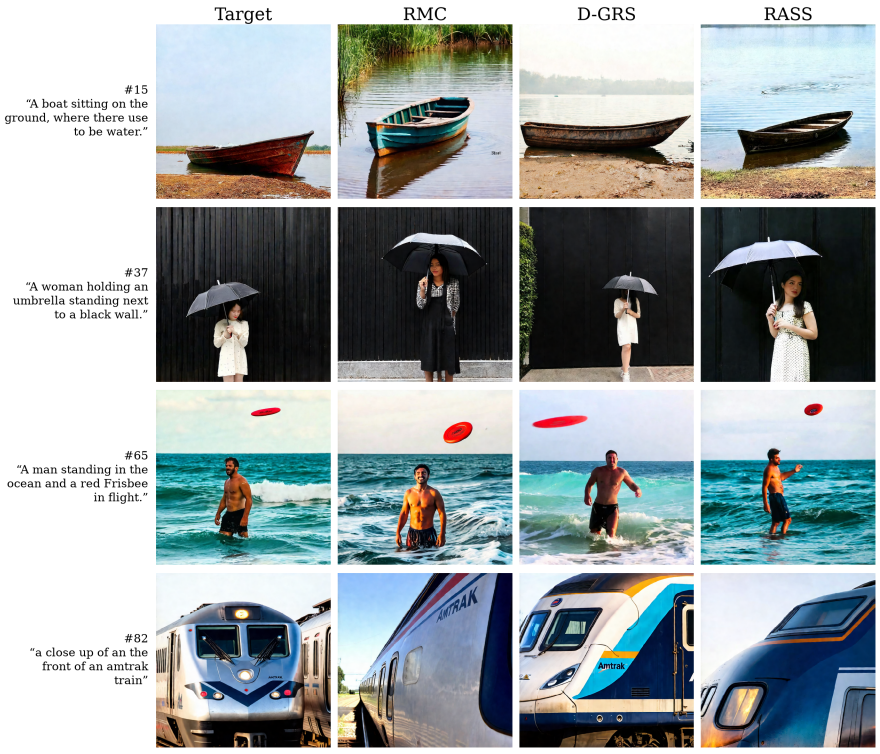


Figure 17: Stable Diffusion 3.5 Medium (512×512), $\varepsilon = 0.8$, $K = 6$, $L = 3$.