ACCELERATING SPECULATIVE DIFFUSIONS VIA BLOCK VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative decoding speeds up LLM inference by using a draft model to generate tokens, with an acceptance-rejection scheme that ensures that the output matches the target distribution. Adapting this to continuous diffusions is difficult because speculative sampling requires drawing from a residual distribution. While straightforward in discrete spaces, efficiently sampling this residual in continuous space is non-trivial. Consequently, existing diffusion adaptations either use computationally inefficient sampling techniques or rely on an alternative scheme. In this work, we introduce a novel scheme that efficiently implements the original speculative sampling mechanism for diffusion models. Our approach offers a critical advantage over current methods: it enables us to adapt block verification from LLMs to diffusions—which provably improves the acceptance rate of drafts. Furthermore, we formalize and analyze the Free Drafter, a heuristic self-speculative drafter for diffusions that requires no training. By enabling block verification, our Free Drafter yields up to a 6.3% speedup over existing speculative methods with no additional training and negligible overhead beyond the existing parallel verification pass.

1 INTRODUCTION

Diffusion models have become ubiquitous in generative modeling (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b). These models excel at generating high-fidelity samples by reversing a noising process that transforms a data distribution into Gaussian noise. However, sampling from a diffusion model requires many evaluations of large neural networks to generate data.

Among the many approaches proposed for acceleration, one promising direction is speculative diffusions (Wang et al., 2024; De Bortoli et al., 2025; Hu et al., 2025; Li et al., 2025; Wen et al., 2026)—the application of speculative sampling to diffusion models. Originally introduced for LLMs, the core idea of speculative decoding is to leverage a small, faster drafter model to propose a sequence of candidate tokens. These tokens are then verified in parallel by the larger, more accurate target model. A subsequence of the proposed draft is accepted and taken as part of the output, and the first rejected draft token is replaced by a draw from a residual distribution (Leviathan et al., 2023; Chen et al., 2023). A critical property of speculative sampling is that the output distribution of the algorithm exactly matches the target model's distribution—there is no loss in sample quality. For LLMs, block verification is a state-of-the-art extension that further improves speculative decoding without impacting sample quality. Block verification provides an optimal change in the verification strategy: instead of verifying the proposed sequence token-by-token (sample verification, a.k.a. token verification for LLMs), the entire block is jointly verified (Sun et al., 2025).

As first observed by Sun et al. (2023), speculative sampling can be viewed as a sequential implementation of the Γ-maximal coupling (Lindvall, 1992). Given a pair of distributions p, q and a draft state $\mathsf{X} \sim p$, a maximal coupling can be thought of as an algorithm that outputs $\tilde{\mathsf{X}} \sim q$ while maximizing the probability that $\tilde{\mathsf{X}} = \mathsf{X}$. That is, it transforms a draft sample into a target sample while maximizing the probability of accepting the draft sample. However, implementing Γ-maximal coupling requires efficient sampling from a residual distribution r_Γ. While trivial in discrete settings, continuous settings typically rely on standard rejection sampling procedures (Jacob, 2021; Wang et al., 2024; Subbaraman et al., 2025; Zou et al., 2025). Unfortunately, this incurs a high computational cost (Jacob, 2021; De Bortoli et al., 2025). While recent work by Anari et al. (2026) leverages parallelism to improve efficiency, the resulting procedure is complex, relies on a fallback tree, and suffers from

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.

Table 1: Summary of the speculative diffusions variants. Verification and residual methods in Algo. 1 can be swapped to give the following. † was proposed in De Bortoli et al. (2025) and Hu et al. (2025).

Approach	Verification (Line 11)	Residual (Line 16)	Stochastic
Reflection† (R)	Sample (Algo. 2)	Reflection (Algo. 4)	✗
Decomposition (D)	Sample (Algo. 2)	Decomposition (Algo. 5, $\alpha = 1$)	✓
Block (B)	Block (Algo. 6)	Decomposition (Algo. 5)	✓

stochastic execution times. To bypass these bottleneck, speculative diffusion methods proposed in De Bortoli et al. (2025); Hu et al. (2025) employ the reflection-maximal coupling (Bou-Rabee et al., 2020) instead of the Γ-maximal coupling, as the former is significantly simpler and cheaper.

In this paper, we make the following contributions:

- **Efficient Γ-maximal coupling for diffusions**: We implement Γ-maximal coupling for diffusion models using a new, one-step algorithm to sample the residual distribution r_Γ. By requiring only a single target model evaluation, our implementation significantly simplifies existing techniques (Wang et al., 2024; Anari et al., 2026) (see Algos. 5 and 7).
- **Block verification for continuous models**: Leveraging our sampling procedure, we adapt the LLM block verification of Sun et al. (2025) to the diffusion setting (see Algo. 6).
- **Theoretical analysis of coupling limitations**: We prove that a broad class of reflection-style deterministic corrections cannot be used with block verification (see Prop. 3.2).
- **The Free Drafter**: We introduce a heuristic self-speculative mechanism—the Free Drafter—which empirically outperforms previously considered drafting strategies in speculative diffusion.
- **Empirical speedups**: Experimentally, block verification scheme reduces wall-clock latency by up to 6.3% versus standard speculative sampling, with no additional training or computational cost.

Notation Let $[\![\texttt{pred}]\!]$ denote Iverson brackets (Knuth, 1992) which evaluate to 1 when predicate $\texttt{pred}$ is true and evaluate to 0 when false. Id denotes the identity matrix. Random variables are denoted via sans serif font, *i.e.*, $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$. When discussing speculative diffusions, we denote the drafter by p, the target model by q, and the residual function by r. The draft size will be denoted by γ. Draft samples are denoted with a hat $\hat{y}$ and we denote sequences via $y_{i:j} = (y_k)_{k=i}^{j}$ for $i \leq j$.

2 Speculative Diffusions

Diffusion models. Let q_{data} be a data distribution on $\mathbb{R}^d$. As in Song et al. (2021b), the forward noising process $(\mathsf{X}_t)_{t\in[0,1]}$ of the model is defined by

$$\mathrm{d}\mathsf{X}_t = f_t \mathsf{X}_t \,\mathrm{d}t + g_t \,\mathrm{d}\mathsf{B}_t, \qquad \mathsf{X}_0 \sim q_{\text{data}}, \tag{1}$$

where $(\mathsf{B}_t)_{t\in[0,1]}$ corresponds to d-dimensional Brownian motion. Let q_t denote the distribution of X_t and set f_t, g_t such that $q_1 = \mathcal{N}(0, \mathrm{Id})$. The generative process $(\mathsf{Y}_t)_{t\in[0,1]}$ is defined by

$$\mathrm{d}\mathsf{Y}_t = b_t(\mathsf{Y}_t) \,\mathrm{d}t + \varepsilon g_{1-t} \,\mathrm{d}\mathsf{W}_t, \qquad \mathsf{Y}_0 \sim q_1, \tag{2}$$

where $b_t(x) = -f_{1-t}x + \frac{1+\varepsilon^2}{2} g_{1-t}^2 s_{1-t}(x)$ with $s_t(x) = \nabla \log q_t(x)$ denoting the Stein score; and $(\mathsf{W}_t)_{t\in[0,1]}$ corresponds to another d-dimensional Brownian motion. The churn parameter $\varepsilon \geq 0$ (Karras et al., 2022) controls the stochasticity of $(\mathsf{Y}_t)_{t\in[0,1]}$ (Albergo et al., 2025). One can show that $\mathsf{Y}_{1-t} \sim q_t$, hence in particular $\mathsf{Y}_1 \sim q_{\text{data}}$. We always set here $\varepsilon > 0$.

In practice, b_t is approximated via a neural network to obtain $\hat{b}_t$. At inference, a timestep discretized version of $(\mathsf{Y}_t)_{t\in[0,1]}$ is evaluated with $q_1 = \mathcal{N}(0, \mathrm{Id})$. Denote a $K+1$ step discretization of $[0,1]$ as $(t_k)_{k=0}^{K}$, with $t_k = k\delta$ and $\delta = 1/K$, and denote the corresponding generative process as DDPM.

Drafter and Target Models. When using an Euler–Maruyama discretization of the generative process defined by time discretization $(t_k)_{k=0}^{K}$, the resulting Markov chain has transition densities

$$q(y_{k+1} \mid y_k) = \mathcal{N}(y_{k+1}; m_k^q(y_k), \sigma_k^2 \mathrm{Id}), \qquad \text{for} \quad m_k^q(y) = y + \delta \hat{b}_{t_k}^q(y), \ \sigma_k = \sqrt{\delta}\varepsilon g_{1-t_k}. \tag{3}$$

Speculative diffusions use a drafter model defined by transition densities

$$p(y_{k+1} \mid y_k) = \mathcal{N}(y_{k+1}; m_k^p(y_{j:k}), \sigma_k^2 \mathrm{Id}), \tag{4}$$

Algorithm 1: Abstract Speculative Diffusions

Require: Target model q, drafter model p, draft size γ, number of sampling steps K.

1: Sample $\mathsf{Y}_0 \sim \mathcal{N}(0, \mathrm{Id})$ and set $k = 0$.
2: **while** $k < K$ **do**
3: Set $\gamma_k = \min\{\gamma, K - k\}$.
4: ▷ Construct Draft.
5: Initialize draft $\hat{\mathsf{Y}}_k = \mathsf{Y}_k$.
6: **for** $j \in \{1, \dots, \gamma_k\}$ **do**
7: Sample $\hat{\mathsf{Y}}_{k+j} \sim p(\cdot \mid \hat{\mathsf{Y}}_{k:k+j-1})$. ▷ Cache draft mean m^p_{k+j-1} computed in sampling.
8: **end for**
9: ▷ Verify Draft.
10: Compute $m^q_{k+j-1} = m^q_{k+j-1}(\hat{\mathsf{Y}}_{k+j-1})$ for all $j \in \{1, \dots, \min\{\gamma_k + 1, K - k\}\}$ in parallel.
11: Compute $\tau, \Delta, \mathsf{Z}, m^q, \sigma, \alpha_{\text{res}} = \texttt{Verification}((\hat{\mathsf{Y}}_{k+j}, m^p_{k+j-1}, m^q_{k+j-1}, \sigma_{k+j-1})_{j=1}^{\gamma_k})$.
12: ▷ Residual Calculation.
13: **if** $\tau = \gamma_k + 1$ and $k + \tau \leq K$ **then** ▷ Latter condition handles end-of-sampling case.
14: Sample $\mathsf{Y}_{k+\gamma_k+1} \sim q(\cdot \mid \hat{\mathsf{Y}}_{k+\gamma_k})$.
15: **else if** $k + \tau \leq K$ **then**
16: Compute $\mathsf{Y}_{k+\tau} = \texttt{Residual}(\Delta, \mathsf{Z}, m^q, \sigma; \alpha_{\text{res}})$. ▷ Ensures $\mathsf{Y}_{k+\tau} \sim q(\cdot \mid \hat{\mathsf{Y}}_{k+\tau-1})$.
17: **end if**
18: Set $\mathsf{Y}_{k+j} = \hat{\mathsf{Y}}_{k+j}$ for $j \in \{1, \dots, \tau - 1\}$; and set $k = \min\{k + \tau, K\}$.
19: **end while**
20: **return** $\mathsf{Y}_{0:K}$.

where the mean is calculated with $\max\{0, k - \gamma + 1\} \leq j < k$ and γ is the draft length we will consider. The drafter is selected such that $b^p_{t_k} \approx b^q_{t_k}$ with $b^p_{t_k}$ being cheaper to evaluate than $b^q_{t_k}$. When the index k is clear from context, we drop the subscript to simplify notation. In the sequel, we assume that $m^p \neq m^q$ as otherwise we could just sample from the drafter with no quality drop.

Speculative Sampling. Speculative sampling uses the fast draft model p to generate a sequence of length γ, which the target model q verifies in parallel. Ideally, this verification adds minimal wall-clock latency over a single target model evaluation. Algo. 1 outlines speculative diffusions: a drafted sequence is validated and corrected via `Verification` and `Residual` algorithms. We refer to one iteration of this process as a *round*. The standard instantiations of speculative sampling for autoregressive LLMs use a sample verification algorithm (Leviathan et al., 2023; Chen et al., 2023), where its diffusion equivalent procedure `SampleVerification`, is depicted in Algo. 2. Alongside the sample verification procedure, the corresponding LLM `Residual` algorithm involves sampling from the following distribution (Algo. 3):

$$r_\Gamma(y) \propto \max\left\{0, q(y \mid \hat{y}_{k+\tau-1}) - p(y \mid \hat{y}_{k:k+\tau-1})\right\}, \tag{5}$$

where τ corresponds to the (relative) index which we need to correct via the residual distribution. For sample verification, τ is the index of the first rejected draft sample (as per Line 5, Algo. 2).

For LLMs, implementing Algo. 3 is straightforward because the state space is finite. However, for diffusion models on continuous spaces, sampling is significantly more complex. Previous methods to address this have relied on standard rejection sampling (Jacob, 2021; Subbaraman et al., 2025; Wang et al., 2024; Zou et al., 2025), which is computationally inefficient, exhibits random runtimes and multiple target model evaluations; see (Jacob, 2021) and (De Bortoli et al., 2025, Section 3.2). While a more sophisticated rejection variant has recently been developed (Anari et al., 2026), it is complex and still suffers from random execution times and multiple target model calls. To bypass these limitations, De Bortoli et al. (2025); Hu et al. (2025) do not sample from r_Γ. Instead, they utilize a deterministic correction provided by the reflection coupling of diffusion models (Bou-Rabee et al., 2020), as per Algo. 4—their proposed speculative diffusion procedures can be summarized as an instantiation of speculative sampling with `SampleVerification` and `Reflection`.

Despite an efficient algorithm using r_Γ remaining elusive, it has be shown that using either the residual sampling (Algo. 3) or the reflection coupling (Algo. 4) guarantees that the output of Algo. 1 follows the target distribution q (Leviathan et al., 2023; De Bortoli et al., 2025; Hu et al., 2025).

Algorithm 2: `SampleVerification`$((\hat{y}_{k+i}, m^p_{k+i-1}, m^q_{k+i-1}, \sigma_{k+i-1})_{i=1}^{\gamma})$

Require: Draft $\hat{y}_{k+1:k+\gamma}$, draft means $m^p_{k:k+\gamma-1}$, target means $m^q_{k:k+\gamma-1}$, variances $\sigma^2_{k:k+\gamma-1}$.

Set $\boldsymbol{c}_0 = 1$.
for $j \in \{1, \dots, \gamma\}$ **do**
  Set $\Delta_j = (m^p_{k+j-1} - m^q_{k+j-1})/\sigma_{k+j-1}$ and $\mathsf{Z}_j = (\hat{y}_{k+j} - m^p_{k+j-1})/\sigma_{k+j-1}$.
  Calculate $\alpha_j = \min\{1, \mathcal{N}(\mathsf{Z}_j + \Delta_j; 0, \mathrm{Id})/\mathcal{N}(\mathsf{Z}_j; 0, \mathrm{Id})\}$.
  Flip coin $\boldsymbol{c}_j = [\![\eta < \alpha_j]\!]$, where $\eta \sim \mathrm{Unif}[0, 1]$.
end for
Set $\tau = \min\left(\{j \in \{1, \dots, \gamma\} \mid \boldsymbol{c}_j = 0\} \cup \{\gamma + 1\}\right)$.
return τ, $\Delta = \Delta_\tau$, $\mathsf{Z} = \mathsf{Z}_\tau$, $m^q = m^q_{k+\tau-1}$, $\sigma = \sigma_{k+\tau-1}$, $\alpha_{\text{res}} = 1$.[1]

Algorithm 3: Γ-`Coupling`(p, q)

Require: Denoising difference Δ, draft noise Z, target mean m^q, variance $\sigma^2 > 0$ (ignore α_{res}).

Define residual distribution:
  $r_\Gamma(y) \propto \max\{0, q(y) - p(y)\}$.
return $\mathsf{Y} \sim r_\Gamma$.

Algorithm 4: `Reflection`$(\Delta, \mathsf{Z}, m^q, \sigma)$

Compute normalized $e = \Delta/\|\Delta\|$.
Compute noise $\mathsf{Z}_{\text{r}} = (\mathrm{Id} - 2ee^\top)\mathsf{Z}$.
return $\mathsf{Y} = m^q + \sigma\mathsf{Z}_{\text{r}}$.

3 Alternative Residuals and Block Verification

In this section, we first introduce a method to sample from a generalized version of the residual distribution r_Γ for diffusion models. Second, we leverage this sampling procedure to derive a block verification scheme for diffusions. Finally, we introduce a self-speculative drafter and analyze its impact on inference acceleration.

3.1 Sampling the Residual Distribution for Diffusions

We show that sampling from the residual distribution r_Γ can be achieved in deterministic time and a single target model evaluation, drastically simplifying existing methods. Sampling from (5) for diffusions boils down to sampling from $r(y) \propto \max\{0, q(y) - p(y)\}$, where $p = \mathcal{N}(m^p, \sigma^2\mathrm{Id})$ and $q = \mathcal{N}(m^q, \sigma^2\mathrm{Id})$. We propose a method leveraging an orthogonal decomposition of Gaussian random variables: ① a 1D sampling task corresponding to the mean difference projection; and ② a Gaussian component. More precisely, we define:

$$\Delta \doteq (m^p - m^q)/\sigma \quad \text{and} \quad e \doteq \Delta/\|\Delta\|. \tag{6}$$

The vector Δ represents the discrepancy in denoising directions between the draft and target models, while e is its normalized counterpart. We assume throughout that $\|\Delta\| \neq 0$. Note that these quantities are also central to the definition of the reflection coupling residual in Algo. 4.

Proposition 3.1. *Let $p = \mathcal{N}(m^p, \sigma^2\mathrm{Id})$, $q = \mathcal{N}(m^q, \sigma^2\mathrm{Id})$, $\alpha \in (0, 1]$, and $\|\Delta\| \neq 0$. The distribution $r(y; \alpha) \propto \max\{0, \alpha q(y) - p(y)\}$ can be sampled as follows:*

$$\mathsf{U} \sim \psi_\alpha; \qquad \mathsf{Z}_\perp \sim \mathcal{N}(0, \mathrm{Id} - ee^\top); \qquad \mathsf{Y} = m^q + \sigma(\mathsf{U}e + \mathsf{Z}_\perp), \tag{7}$$

where $\psi_\alpha(u) \propto \max\{0, \alpha\mathcal{N}(u; 0, 1) - \mathcal{N}(u - \|\Delta\|; 0, 1)\}$ is a univariate distribution; with c.d.f.

$$\Psi_\alpha(u) = \frac{\alpha\Phi(u) - \Phi(u - \|\Delta\|)}{\alpha\Phi\left(\frac{\log\alpha}{\|\Delta\|} + \frac{\|\Delta\|}{2}\right) - \Phi\left(\frac{\log\alpha}{\|\Delta\|} - \frac{\|\Delta\|}{2}\right)} \quad \text{if} \quad u < \frac{\log\alpha}{\|\Delta\|} + \frac{\|\Delta\|}{2} \tag{8}$$

and $\Psi_\alpha(u) = 1$ otherwise, with Φ being the c.d.f. of the standard normal distribution.

The parameter α in Prop. 3.1 is essential for the block verification framework discussed in the sequel. For standard speculative decoding, setting $\alpha = 1$ recovers the original residual r_Γ. This proposition effectively reduces a high-dimensional sampling problem to a tractable 1D task. Once $\mathsf{U} \sim \psi_\alpha$ is obtained, the subsequent sampling of $\mathsf{Z}_\perp$ and Y in (7) is straightforward to implement. Sampling

[1] When $\tau = \gamma + 1$, the values Δ, Z, m^q, and σ can be arbitrary as they will not be used in any subsequent calculation (see the 'if-condition' in Algo. 1). A more precise pseudo-code could combine the verification and residual calculation procedures to avoid these undefined variables, see for instance Algo. 8.

U is also simple as its c.d.f. Ψ_α can be easily computed pointwise as per (8) (*e.g.*, via the normal c.d.f. Φ), so the inverse sampling method can be used. In practice, we employ a bisection search (see Appendix A.1 for details). The complete sampling procedure is summarized in Algo. 5. By replacing the reflection coupling in Algo. 1 (at Line 16) with $\texttt{Decomposition}(\Delta, m^q, \sigma; \alpha = 1)$, we recover standard speculative decoding (Leviathan et al., 2023) adapted to diffusion models.

Algorithm 5: $\texttt{Decomposition}(\Delta, m^q, \sigma; \alpha)$

Require: Denoising difference Δ, target mean m^q, variance $\sigma^2 > 0$, weight $\alpha \in (0, 1]$ (ignore Z).
1: Sample $\mathsf{U} \sim \psi_\alpha$ via (8)
2: Sample $\mathsf{G} \sim \mathcal{N}(0, \mathrm{Id})$, set $\mathsf{Z}_\perp = \mathsf{G} - e(e^\top \mathsf{G})$, where $e = \Delta/\|\Delta\|$ so that $\mathsf{Z}_\perp \sim \mathcal{N}(0, \mathrm{Id} - ee^\top)$.
3: **return** $\mathsf{Y} = m^q + \sigma(\mathsf{U}e + \mathsf{Z}_\perp)$.

3.2 Block Verification

Standard speculative decoding is derived by maximizing $\min\{1, q(\hat{y})/p(\hat{y})\}$, the *per-sample* acceptance rate, which corresponds to the probabilities α calculated in $\texttt{SampleVerification}$ (Algo. 2, Line 4). One way to improve speculative decoding is to alter the verification function (Algo. 1, Line 11). Block verification provides such an alternative: rather than maximizing the acceptance rate for each sample independently, it maximizes the joint acceptance rate of the entire draft block $\hat{y}_{k+1:k+\gamma}$ which reduces the number of target model evaluations (Sun et al., 2025).

We adapt block verification to diffusion models by replacing $\texttt{SampleVerification}$ with the procedure $\texttt{BlockVerification}$, as defined in Algo. 6. To ensure that we sample exactly from the target distribution q, one must employ a generalized variant of the residual distribution r_Γ:

$$r_{\text{BLOCK}}(y) \propto \max\left\{0, \alpha_{\tau-1} q(y \mid \hat{y}_{k+\tau-1}) - p(y \mid \hat{y}_{k:k+\tau-1})\right\}, \tag{9}$$

where $\alpha_{\tau-1}$ is defined recursively in Algo. 6. Note that in block verification τ corresponds to the index after the last accepted sample (as per Line 12, Algo. 6). A derivation of the acceptance probabilities h_j used in Algo. 6 is provided in Appendix E.

Our previous derivation of sample-wise decomposition residual enables sampling from (9). Thus, r_{BLOCK} can be sampled by applying Prop. 3.1 and Algo. 5 with $\alpha = \alpha_{\tau-1}$. As $\texttt{Decomposition}$ samples directly from r_{BLOCK}, instantiating Algo. 1 with $\texttt{BlockVerification}$ and $\texttt{Decomposition}$ yields a block verification algorithm whose output is consistent with the target model q—thereby recovering the theoretical guarantees of Sun et al. (2025, Theorem 1).

Algorithm 6: $\texttt{BlockVerification}((\hat{y}_{k+i}, m^p_{k+i-1}, m^q_{k+i-1}, \sigma_{k+i-1})_{i=1}^{\gamma})$

Require: Draft $\hat{y}_{k+1:k+\gamma}$, draft means $m^p_{k:k+\gamma-1}$, target means $m^q_{k:k+\gamma-1}$, variances $\sigma^2_{k:k+\gamma-1}$.
1: Initialize $\boldsymbol{c}_0 = 1$ and $\alpha_0 = 1$.
2: **for** $j \in \{1, \ldots, \gamma\}$ **do**
3: Set $\Delta_j = (m^p_{k+j-1} - m^q_{k+j-1})/\sigma_{k+j-1}$ and $\mathsf{Z}_j = (\hat{y}_{k+j} - m^p_{k+j-1})/\sigma_{k+j-1}$.
4: Calculate $\alpha_j = \min\{1, \alpha_{j-1}\mathcal{N}(\mathsf{Z}_j + \Delta_j; 0, \mathrm{Id})/\mathcal{N}(\mathsf{Z}_j; 0, \mathrm{Id})\}$.
5: **end for**
6: **for** $j \in \{1, \ldots, \gamma\}$ **do**
7: **if** $j \neq \gamma$ **then**
8: Calculate $h_j = \frac{v_j}{v_j + 1 - \alpha_j}$, where $v_j = \alpha_j \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} + \frac{\|\Delta_{j+1}\|}{2}\right) - \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} - \frac{\|\Delta_{j+1}\|}{2}\right)$.
9: **else**
10: Set $h_\gamma = \alpha_\gamma$.
11: **end if**
12: Flip coin $\boldsymbol{c}_j = [\![\eta < h_j]\!]$, where $\eta \sim \mathrm{Unif}[0, 1]$.
13: **end for**
14: Set $\tau = 1 + \max\{j \in \{0, 1, \ldots, \gamma\} \mid \boldsymbol{c}_j = 1\}$.
15: **return** τ, $\Delta = \Delta_\tau$, $\mathsf{Z} = \mathsf{Z}_\tau$, $m^q = m^q_{k+\tau-1}$, $\sigma = \sigma_{k+\tau-1}$, $\alpha_{\text{res}} = \alpha_{\tau-1}$.[1]

A natural question is if a deterministic residual calculation exists for the block verification case, *e.g.*, reflection coupling's residual. The following rules out a broad class of reflection-style corrections, including that of $\texttt{Reflection}$, and thus motives the stochastic residual used in block verification.

Proposition 3.2 (Informal). *For draft sequences of length $\gamma \geq 2$, there exists no valid deterministic residual correction of the form $t(\mathsf{Z})$ for block verification that is simultaneously invertible and has a constant Jacobian factor under change of variables.*

3.3 Self-Speculative Drafter

In the context of speculative diffusion, self-speculative approaches have proven highly effective (De Bortoli et al., 2025). One strategy involves extrapolating a single evaluation of the target model q to generate a draft sequence over $i \in \{1, \dots, \gamma\}$ steps, a method we refer to as the *Frozen Drafter*. This approach is motivated by the exchangeability property of Ornstein–Uhlenbeck diffusions (Hu et al., 2025), which can be derived via stochastic localization (Eldan, 2013; El Alaoui and Montanari, 2022). Given a time discretization $(t_k)_{k=0}^K$ and an initial denoising state y_k, the drafting strategy is defined by setting the drift terms $b^p_{t_{k+i-1}}$ for $i \in \{1, \dots, \gamma\}$ as:

$$b^p_{t_{k+i-1}}(\hat{y}) = -f_{1-t_{k+i-1}}\hat{y} + \frac{1+\varepsilon^2}{2} g^2_{1-t_{k+i-1}} s^q_{1-t_k}(y_k), \tag{10}$$

where s^q denotes the target model's neural network approximation of the score function s. Crucially, this expression differs from the standard drift $b^q_{t_{k+i-1}}(y)$ by utilizing a time-delayed score $s^q_{1-t_k}(y_k)$, evaluated only once at the beginning of the current speculative round. Consequently, for a drafting size γ, we require only a single call of the target model to generate the entire draft sequence via (10).

An appealing property of the Frozen Drafter is that it provides a theoretical guarantee on the number of (parallel) target model calls q in the sample verification setting. Specifically, Hu et al. (2025) established that, under mild data assumptions and an appropriate choice of γ, the number of parallel calls scales asymptotically as $\mathcal{O}(K^{2/3}(\beta d\delta)^{1/3})$, where β depends on the data assumption. We adapt this result here to speculative diffusions with block verification as follows.

Proposition 3.3. *Assume the use of the Frozen Drafter and* $\mathrm{Tr}(\mathrm{Cov})[q_{\text{data}}] \leq \beta d$. *For a fixed draft size γ, let $\rho(\gamma) \in [0, 1]$ denote the ratio of the expected number of rounds required to complete speculative diffusion with block verification relative to sample verification. Taking $\gamma \asymp (K/\beta\delta d)^{1/3}$, the expected number of parallel target model calls under block verification is at most $\mathcal{O}(\rho(\gamma)K^{2/3}(\beta d\delta)^{1/3})$.*

The existence of $\rho(\gamma) \in [0, 1]$ is guaranteed from the optimality of block verification over alternative sample verification strategies (Sun et al., 2025, Theorem 2).

While the Frozen Drafter provides theoretical guarantees on the number of parallel target model calls (Prop. 3.3), it admits a significant practical limitation. As per (10), each round of speculative diffusion requires computing $s^q_{1-t_k}(y_k)$. Consequently, the overhead of generating the draft sequence $\hat{y}$ is approximately equal to the complexity of a single target model call q. Hence, even in ideal settings, any speculative diffusion using the Frozen Drafter will achieve limited speedups. This limitation is tied to the *block efficiency* of a speculative diffusion algorithm: the number of denoising steps computed after a round of speculative diffusion.

Proposition 3.4 (Informal). *Let $\mathtt{be}(k)$ denote the block efficiency of a speculative diffusion algorithm with the Frozen Drafter at the kth denoising step. The speedup compared to no speculative diffusion in ideal conditions is approximately equal to $\mathtt{be}(k)/2$.*

Here, ideal conditions assume that the wall-clock time of a single target model evaluation dominates all other costs (*e.g.*, drafter overhead, residual calculations, *etc.*) and that the parallel execution of the target model includes negligible overhead. In practice, with these additional costs, the speedup will be worse than $\mathtt{be}(k)/2$. This suggests that if the Frozen Drafter exhibits low block efficiency or employs a small draft length γ (noting that $\mathtt{be}(k) \in [1, \gamma + 1]$), the use of the Frozen Drafter potentially results in a slow-down compared to vanilla sampling.

To circumvent the wall-time dependence on the target model q inherent in the Frozen Drafter, De Bortoli et al. (2025, Appendix B) proposed a variant that extracts the target model's score function from the verification step of the previous speculative round. Specifically, rather than initiating the current drafting round k by calling the target model to compute $s^q_{1-t_k}(y_k)$, we reuse the previously computed score $s^q_{1-t_k}(\hat{y}_k)$ obtained during the parallel verification phase (Line 10 in Algo. 1). In the case in which the previous speculative round result in a full accept of the draft, we reuse the score utilized to compute the extra sample $\mathsf{Y}_{k+\tau}$ (obtained during verification on Line 10 in Algo. 1).

In this scheme, the target model is only explicitly called at the very beginning of the first speculative round, where no prior verification results are available. We refer to this approach as the *Free Drafter*, as the marginal cost of drafting becomes negligible compared to a full target model evaluation. Given this substantial reduction in overhead, a straightforward corollary of Prop. 3.4 follows.

Corollary 3.5 (Informal)**.** *Under ideal conditions, the Frozen Drafter is only faster than the Free Drafter if its block efficiency is (approximately) twice as large.*

Ideal conditions correspond to the case where the cost (in wall-clock time) of parallelizing the target model $\gamma + 1$ times is similar to a single sequential call, the cost of Free Drafter is negligible, and the cost of residual calculation is negligible. A formal statement and proof are provided in Appendix H. As block efficiency is at most $\gamma + 1$, this corollary implies that if the Free Drafter maintains a sufficiently high block efficiency, it will outperform the Frozen Drafter. This remains true despite the theoretical expectation that the Frozen Drafter might achieve higher block efficiency due to its closer alignment with the exchangeability properties of diffusion models (Hu et al., 2025).

4 Related Work

Accelerating Diffusions. Besides speculative diffusions, various alternative approaches for accelerating diffusion model sampling exist. One line of work aims to distill a teacher diffusion model into a student which maintains high quality samples with a reduced number of denoising steps (Luhman and Luhman, 2021; Salimans and Ho, 2022; Berthelot et al., 2023; Liu et al., 2023; Meng et al., 2023; Sauer et al., 2024; Song et al., 2023; Katzir et al., 2024; Kim et al., 2024; Xu et al., 2024; Yin et al., 2024; Xu et al., 2025; Boffi et al., 2025). Importantly, distillation requires the explicit training of a separate student model and typically has a drop in sample quality when compared to the teacher model (Luo, 2023; Dieleman, 2024). Other approaches propose modifications to the sampling procedure of diffusion models through improved integrators (Dockhorn et al., 2022; Liu et al., 2022a; Lu et al., 2022; Xiao et al., 2022; Zhang and Chen, 2023). Similar to speculative diffusions, other approaches leverage parallel sampling via Picard iteration (Pokle et al., 2022; Shih et al., 2023; Chen et al., 2024a; Li et al., 2024a; Ma et al., 2024; Tang et al., 2024). Unlike the usual diffusion model sampling (and speculative diffusions), Picard iteration relies on repeated parallel sampling calls on a window until a fixed point is detected. It should however be noted that many of these approaches can be used in conjunction with speculative diffusions (De Bortoli et al., 2025).

Speculative Decoding for Diffusions. De Bortoli et al. (2025); Hu et al. (2025) concurrently proposed speculative diffusions by leveraging the reflection coupling residual and the Frozen Drafter for their self-speculative approaches. Hu et al. (2025) uncovered an exchangeability property via stochastic localization (Eldan, 2013; Chen and Eldan, 2022), resulting in a speedup guarantee over regular diffusion model sampling. De Bortoli et al. (2025) outlined the heuristic Free Drafter, but we analysis why the Frozen Drafter comparatively has limited speedup possibilities. In the context of Masked Autoregressive models (Li et al., 2024b), a previous attempt at sampling from the original LLM residual in a continuous sampling space was explored (Wang et al., 2024; Subbaraman et al., 2025; Zou et al., 2025). However, this approach relies on rejection sampling with the target model as its proposal, which is computationally inefficient (Jacob, 2021; De Bortoli et al., 2025). Speculative decoding has successfully sped up Jacobi decoding (Ortega and Rheinboldt, 2000; Song et al., 2021a) for autoregressive image generation (Teng et al., 2025). For text, speculative decoding has sped up discrete diffusions (Agrawal et al., 2026; Gao et al., 2025; Campbell et al., 2026) and diffusion models have been used as drafters for LLM speculative decoding (Li et al., 2026; Christopher et al., 2025). For molecular dynamics (Leimkuhler and Matthews, 2015), speculative sampling first accelerated simple Langevin dynamics (De Bortoli et al., 2025) and was later extended to second-order integrators (Kosmala et al., 2026). Block verification can be extended to these integrators, see Appendix L. Kosmala et al. (2026) also proposed an asynchronous verification pipeline.

5 Experiments

We empirically explore the improvements that block verification has over sample verification in speculative diffusions. We compare the three verification and residual combinations (Table 1): (R) `SampleVerification` with the `Reflection` residual; (D) `SampleVerification` with the `Decomposition` residual; (B) `BlockVerification` with the `Decomposition` residual.

Table 2: Wall-clock speedups over DDPM and block efficiency over all datasets for $K = 250$ denoising steps and drafting size $\gamma = 7$. Free Drafter is used in all instances. Non-FID values are calculated over 500 samples with $\pm$ error ranges approximated via error propagation.

		Wall-clock Speedup over DDPM				Avg. Block Efficiency		
Dataset	ε	R	D	B	R↑B%	R	D	B
CelebA LDM	0.25	2.20 ± 0.09	2.19 ± 0.09	2.25 ± 0.10	2.10%	3.89	3.86	4.19
	0.50	2.15 ± 0.09	2.14 ± 0.09	2.20 ± 0.10	2.43%	3.90	3.90	4.20
	0.75	2.01 ± 0.09	2.00 ± 0.09	2.06 ± 0.09	2.01%	3.68	3.68	3.96
	1.00	1.88 ± 0.08	1.87 ± 0.08	1.90 ± 0.08	1.47%	3.41	3.42	3.67
CelebA Pixel	0.25	3.20 ± 0.20	3.18 ± 0.21	3.35 ± 0.21	4.81%	4.58	4.58	5.00
	0.50	3.15 ± 0.21	3.14 ± 0.20	3.30 ± 0.20	4.83%	4.53	4.53	4.97
	0.75	2.94 ± 0.20	2.95 ± 0.19	3.11 ± 0.20	5.88%	4.22	4.26	4.70
	1.00	2.70 ± 0.19	2.71 ± 0.19	2.87 ± 0.19	6.28%	3.84	3.87	4.34
ImageNet LDM	0.25	2.24 ± 0.10	2.23 ± 0.10	2.28 ± 0.11	1.76%	4.02	4.01	4.25
	0.50	2.20 ± 0.10	2.20 ± 0.10	2.25 ± 0.11	2.32%	4.02	4.02	4.30
	0.75	2.07 ± 0.10	2.08 ± 0.10	2.12 ± 0.11	2.25%	3.85	3.86	4.08
	1.00	1.93 ± 0.09	1.92 ± 0.09	1.97 ± 0.10	2.07%	3.61	3.61	3.84
ImageNet Pixel	0.25	2.82 ± 0.23	2.81 ± 0.22	2.96 ± 0.22	4.70%	4.15	4.14	4.53
	0.50	2.81 ± 0.25	2.80 ± 0.24	2.94 ± 0.24	4.66%	4.17	4.17	4.56
	0.75	2.63 ± 0.24	2.61 ± 0.22	2.75 ± 0.25	4.62%	3.89	3.89	4.28
	1.00	2.42 ± 0.23	2.41 ± 0.21	2.56 ± 0.23	5.57%	3.53	3.54	3.95
LSUN Pixel	0.25	2.74 ± 0.19	2.74 ± 0.17	2.86 ± 0.19	4.44%	4.21	4.23	4.66
	0.50	2.72 ± 0.19	2.72 ± 0.18	2.83 ± 0.20	4.08%	4.19	4.22	4.63
	0.75	2.52 ± 0.18	2.54 ± 0.17	2.65 ± 0.18	5.22%	3.85	3.90	4.34
	1.00	2.31 ± 0.16	2.33 ± 0.16	2.42 ± 0.17	4.59%	3.49	3.53	3.94
CIFAR10 Pixel	0.25	3.61 ± 0.25	3.59 ± 0.24	3.60 ± 0.24	−0.04%	6.07	6.06	6.35
	0.50	3.65 ± 0.24	3.67 ± 0.24	3.67 ± 0.24	0.54%	6.23	6.25	6.52
	0.75	3.58 ± 0.24	3.56 ± 0.23	3.59 ± 0.22	0.24%	6.14	6.16	6.44
	1.00	3.46 ± 0.22	3.45 ± 0.23	3.47 ± 0.21	0.24%	5.98	5.99	6.30

We denote the usual Euler–Maruyama integration of the sampling process as DDPM. For efficiency metrics, we consider wall-clock speedups (w.r.t. sampling without speculation) and average block efficiency—the average accepted number of samples per round of speculative diffusions, averaged over the K denoising steps. We also compute the percentage speedup (B) yields over (R) in R↑B% (*i.e.*, percentage above 100%). To empirically verify that quality does not change, we report the Fréchet Inception Distance (FID) (Heusel et al., 2017). We also show empirically that the heuristic Free Drafter provides practical speedups over the Frozen Drafter, despite its superior block efficiency.

We consider a large collection of image datasets. In pixel space, we consider CIFAR10 ($32 \times 32 \times 3$), LSUN (bedroom) ($64 \times 64 \times 3$), CelebA ($64 \times 64 \times 3$), and ImageNet ($64 \times 64 \times 3$). In latent space, we consider CelebA ($256 \times 256 \times 3$) and ImageNet ($256 \times 256 \times 3$), both with latent spaces of dimension $64 \times 64 \times 3$. All models utilize a U-Net backbone, with further details in Appendix J. We focus on ImageNet LDM when displaying our results. All experiments are computed using Cloud TPU V3 hardware. Following Shih et al. (2023); Hu et al. (2025), when making our parallel primary call in verification, we parallelize over multiple compute units ($\gamma + 1$ given a draft size γ).

Summary. Table 2 summarizes the various efficiency metrics for our experiments. Speculative diffusion speeds up standard DDPM by 1.9–3.6× in wall-clock time. While switching between `Reflection` (R) and `Decomposition` (D) residuals has minimal impact, block verification typically improves wall-clock speedups by 1.5–6.3% over sample verification counterparts. Block verification consistently yields superior block efficiency. The only exception is CIFAR10, where an already high block efficiency (~ 6) prevents further wall-clock gains because the marginal efficiency increase cannot overcome the slightly higher overhead of the block verification scheme.

Number of sampling steps. In Table 3 we explore the wall-clock speedups and FID scores for ImageNet LDM for churn parameter $\varepsilon = 0.5$ and a draft size of $\gamma = 7$ over various denoising steps. From the quality side, we see that the FIDs are all similar across the different speculative diffusions approaches when compared to DDPM. There are small differences when comparing between the speculative diffusions variants, but this is within expected numerical and stochastic variation. One trend that we find is that a higher number of denoising steps typically allows for a larger wall-clock speedup (compared to DDPM). We attribute this trend to the diffusion noise schedule property that as

Table 3: Wall-clock speedups and 50k FID values for ImageNet LDM over multiple denoising steps for churn value $\varepsilon = 0.5$ and window size $\gamma = 7$. Free Drafter is used in all instances. Non-FID values are calculated over 500 samples with $\pm$ error ranges approximated via error propagation.

	Wall-clock Speedup				FID			
Steps	R	D	B	R↑B%	R	D	B	DDPM
50	1.22 ± 0.06	1.22 ± 0.06	1.22 ± 0.07	0.09%	10.83	10.77	10.82	10.78
100	1.55 ± 0.08	1.55 ± 0.08	1.57 ± 0.08	1.47%	10.25	10.06	9.84	10.28
250	2.20 ± 0.10	2.20 ± 0.10	2.25 ± 0.11	2.32%	10.09	9.55	9.63	9.81
500	2.90 ± 0.12	2.91 ± 0.13	3.01 ± 0.14	3.82%	9.98	9.66	9.51	9.64
1000	3.78 ± 0.13	3.79 ± 0.14	3.91 ± 0.14	3.67%	9.90	9.65	9.41	9.61

Table 4: Wall-clock speedups and block efficiency values for ImageNet LDM over multiple denoising steps for churn value $\varepsilon = 1.0$ and window size $\gamma = 7$. Parenthesis are measurements compared to Free Drafter as a percentage improvement. Quantities are calculated over 500 samples.

	Frozen Wall-clock Speedup over DDPM			Frozen Average Block Efficiency		
Steps	R	D	B	R	D	B
50	1.07 (−8.0%)	1.08 (−7.3%)	1.08 (−7.1%)	2.76 (40.9%)	2.76 (44.3%)	2.87 (39.5%)
100	1.26 (−11.5%)	1.26 (−10.8%)	1.28 (−11.4%)	3.39 (32.3%)	3.40 (32.2%)	3.54 (28.2%)
250	1.58 (−18.0%)	1.58 (−17.7%)	1.60 (−18.7%)	4.32 (19.8%)	4.33 (20.0%)	4.48 (16.8%)
500	1.89 (−24.2%)	1.89 (−24.1%)	1.94 (−24.7%)	5.05 (14.4%)	5.05 (14.0%)	5.22 (11.8%)
1000	2.28 (−30.5%)	2.28 (−30.6%)	2.34 (−31.0%)	5.77 (9.8%)	5.77 (9.3%)	5.95 (8.1%)

$k \to K$, $\sigma_k \to 0$. In both versions of verification, as σ_k decreases the acceptance of a draft image decreases. Intuitively, accepting a draft image becomes increasingly more difficult as we get closer to denoising a clean image. As a result, a larger number of denoising steps K allows us to stay in the high noise σ_k regime, and thus allows us to have increased speedups.

Frozen or Free. A comparison between the Frozen and Free Drafter is given in Table 4. Consistently, Frozen Drafter has superior block efficiency. Despite this, it is always worse in terms of its wall-clock speedups than its Free Drafter counterpart: the increase in block efficiency does not overcome the wall-clock cost of a drafter whose cost scales with a single call of the primary model. We note that the speedup from using the Free Drafter is significant even for smaller number of denoising steps.

6 Discussion

Alongside the (up to 6.3%) latency speedups our block verification self-speculative sampling approach provides over its reflection coupling counterpart, we find that experimentally speculative diffusions provides better speedups with large number of denoising steps. Thus, utilizing acceleration methods which decrease the number of denoising steps yield smaller speedups in combination with speculative diffusion. As per the speculative diffusions framework, our acceleration approach is not applicable to deterministic samplers and the draft model must have the same denoising schedule as the target model (De Bortoli et al., 2025). Similar to other parallelization methods (Shih et al., 2023; Chen et al., 2024a), another limitation is that verification increases memory overhead due to the parallel call of the primary model. Although we conjecture it is possible, extending block verification to an asynchronous pipeline—with continuous drafting and asynchronous verification rollbacks–remains unclear (Kosmala et al., 2026). Unlike sample verification where it is unambiguous when rollbacks must occur, for block verification, we would need to determine when to give up on a current draft sequence and calculate a residual—for positive α_i, there is still a chance to accept the block. A common method use to further speed up speculative decoding in LLMs is to exploit different draft topologies, *e.g.*, a draft tree (Miao et al., 2024; Chen et al., 2024b). The goal of deviating from the sequential chain topology of a vanilla drafter is to explore draft possibilities which may yield larger acceptance chances, noting that the chance of draft acceptance decrease as a sequential draft's length increases. We conjecture that our work unlocks the possibility of utilizing more complex draft topologies with block verification, mirroring recent work in LLM speculative decoding, *e.g.*, block verification with multiple draft sequences (Lin et al., 2026; Thomas and Pal, 2026) and draft trees (Weng et al., 2025; 2026). Indeed, residuals in the form of Prop. 3.1 are key to these more sophisticated drafting strategies, see for instance Weng et al. (2025, Algorithm 3, Line 11).

AI USE STATEMENT

In this work, we used generative AI tools for help develop theoretical models or conceptual frameworks, implement methods, formulate mathematical claims, and assist in the writing of proofs. We have not used generative AI tools for propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean and reformat dataset, support qualitative and thematic data analysis, and interpret results, and generate synthetic data sets and assist with translation are not applicable to this work. We have reviewed all AI-assisted work. LLMs were used to check correctness of proofs which were manually verified. The molecular dynamics experiments of Appendix L were implemented with the assistance of LLMs, where the correctness of code was manually verified. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Sudhanshu Agrawal, Risheek Garrepalli, Raghavv Goel, Christopher Lott, Fatih Porikli, and Mingu Lee. Structuring the future: Diffusion LLM speculative decoding via calibrated draft graphs. In *ICML 2026 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2026.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Nima Anari, Carlo Baronio, CJ Chen, Alireza Haqi, Frederic Koehler, Anqi Li, and Thuy-Duong Vuong. Parallel sampling via autospeculation. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, STOC '26, page 1168–1179. Association for Computing Machinery, 2026.

Gregor Bachmann, Sotiris Anagnostidis, Albert Pumarola, Markos Georgopoulos, Artsiom Sanakoyeu, Yuming Du, Edgar Schönfeld, Ali Thabet, and Jonas Kohler. Judge decoding: Faster speculative sampling requires going beyond model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in neural information processing systems*, 35:11423–11436, 2022.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. TRACT: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

Nicholas M Boffi, Michael S Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems*, 2025.

Nawaf Bou-Rabee, Andreas Eberle, and Raphael Zimmer. Coupling and convergence for Hamiltonian Monte Carlo. *The Annals of Applied Probability*, 30(3):1209–1250, 2020.

Andrew Campbell, Valentin De Bortoli, Jiaxin Shi, and Arnaud Doucet. Self-speculative masked diffusions. In *International Conference on Learning Representations*, 2026.

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*, 2024a.

Yuansi Chen and Ronen Eldan. Localization schemes: A framework for proving mixing bounds for markov chains. In *IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 110–122, 2022.

Zhuoming Chen, Avner May, Ruslan Svirschevski, Yuhsun Huang, Max Ryabinin, Zhihao Jia, and Beidi Chen. Sequoia: Scalable and robust speculative decoding. *Advances in Neural Information Processing Systems*, 37:129531–129563, 2024b.

Jacob K Christopher, Brian R Bartoldson, Tal Ben-Nun, Michael Cardei, Bhavya Kailkhura, and Ferdinando Fioretto. Speculative diffusion decoding: Accelerating language generation through diffusion. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12042–12059, 2025.

Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling. In *International Conference on Machine Learning*, 2025.

Sander Dieleman. The paradox of diffusion distillation, 2024. URL https://sander.ai/2024/02/28/paradox.html.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. In *Advances in Neural Information Processing Systems*, 2022.

Ahmed El Alaoui and Andrea Montanari. An information-theoretic view of stochastic localization. *IEEE Transactions on Information Theory*, 68(11):7423–7426, 2022.

Ronen Eldan. Thin shell implies spectral gap up to polylog via a stochastic localization scheme. *Geometric and Functional Analysis*, 23(2):532–569, 2013.

Yifeng Gao, Ziang Ji, Yuxuan Wang, Biqing Qi, Hanlin Xu, and Linfeng Zhang. Self speculative decoding for diffusion large language models. *arXiv preprint arXiv:2510.04147*, 2025.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 2012.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Hengyuan Hu, Aniket Das, Dorsa Sadigh, and Nima Anari. Diffusion models are secretly exchangeable: Parallelizing DDPMs via auto speculation. In *International Conference on Machine Learning*, 2025.

Zongle Huang, Lei Zhu, Zongyuan Zhan, Ting Hu, Weikai Mao, Xianzhi Yu, Yongpan Liu, and Tianyu Zhang. MoESD: Unveil speculative decoding's potential for accelerating sparse MoE. *Advances in Neural Information Processing Systems*, 38:125276–125311, 2025.

Pierre Jacob. Lectures on Couplings and Monte Carlo. https://sites.google.com/site/pierrejacob/cmclectures, 2021.

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*, 2022.

Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. In *International Conference on Learning Representations*, 2024.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations*, 2024.

Donald E Knuth. Two notes on notation. *The American Mathematical Monthly*, 99(5):403–422, 1992.

Arthur Kosmala, Stephan Günnemann, Meng Gao, and Brandon Wood. Speculative sampling for faster molecular dynamics. In *International Conference on Machine Learning*, 2026.

Eun Sul Lee and Ronald N Forthofer. *Analyzing Complex Survey Data*. Sage Publications, 2005.

Ben Leimkuhler and Charles Matthews. *Molecular Dynamics. With Deterministic and Stochastic Numerical Methods*. Springer, 2015.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, 2023.

Guanghao Li, Zhihui Fu, Min Fang, Qibin Zhao, Ming Tang, Chun Yuan, and Jun Wang. DiffuSpec: Unlocking diffusion language models for speculative decoding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 20896–20910, 2026.

Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Kai Li, and Song Han. DistriFusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems*, 2024b.

Ye Li, Jiahe Feng, Yuan Meng, Kangye Ji, Chen Tang, Xinwan Wen, Shutao Xia, Zhi Wang, and Wenwu Zhu. TS-DP: Reinforcement speculative decoding for temporal adaptive diffusion policy acceleration. *arXiv preprint arXiv:2512.15773*, 2025.

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

Torgny Lindvall. *Lectures on the Coupling Method*. John Wiley & Sons, New York, 1992.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2022b.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. InstaFlow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, 2022.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

Weijian Luo. A comprehensive survey on knowledge distillation of diffusion models. *arXiv preprint arXiv:2304.04262*, 2023.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. DeepCache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

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*, pages 932–949, 2024.

James M Ortega and Werner C Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. SIAM, 2000.

Ashwini Pokle, Zhengyang Geng, and J Zico Kolter. Deep equilibrium approaches to diffusion models. In *Advances in Neural Information Processing Systems*, 2022.

Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Tim Duignan, and Mark Neumann. Orb-v3: atomistic simulation at scale. *arXiv preprint arXiv:2504.06231*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

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*, pages 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*, 2023.

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

Yang Song, Chenlin Meng, Renjie Liao, and Stefano Ermon. Accelerating feedforward computation via parallel nonlinear equation solving. In *International Conference on Machine Learning*, 2021a.

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.

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

Pranav Subbaraman, Fang Sun, Yue Yao, Huacong Tang, Xiao Luo, and Yizhou Sun. Accelerating time series foundation models with speculative decoding. *arXiv preprint arXiv:2511.18191*, 2025.

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*, 2023.

Ziteng Sun, Uri Mendlovic, Yaniv Leviathan, Asaf Aharoni, Jae Hun Ro, Ahmad Beirami, and Ananda Theertha Suresh. Block verification accelerates speculative decoding. In *International Conference on Learning Representations*, 2025.

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

Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Accelerating auto-regressive text-to-image generation with training-free speculative Jacobi decoding. In *International Conference on Learning Representations*, 2025.

Rahul Thomas and Arka Pal. Greedy multi-path block verification for faster decoding in speculative sampling. *arXiv preprint arXiv:2602.16961*, 2026.

Vivien Tran-Thien. An optimal lossy variant of speculative decoding, 2023. URL https://vivien000.github.io/blog/journal/a-provably-optimal-lossy-variant-of-speculative-decoding.html.

Zili Wang, Robert Zhang, Kun Ding, Qi Yang, Fei Li, and Shiming Xiang. Continuous speculative decoding for autoregressive image generation. In *Advances in Neural Information Processing Systems*, 2024.

Xinwan Wen, Bowen Li, Jiajun Luo, Ye Li, and Zhi Wang. FREE: Uncertainty-aware autoregression for parallel diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pages 4097–4107, June 2026.

Yepeng Weng, Qiao Hu, Xujie Chen, Li Liu, Dianwen Mei, Huishi Qiu, Jiang Tian, and Zhongchao Shi. Traversal verification for speculative tree decoding. *Advances in Neural Information Processing Systems*, 38:77899–77924, 2025.

Yepeng Weng, Qiao Hu, and Takehisa Yairi. UniVer: A unified perspective for multi-step and multi-draft speculative decoding. *arXiv preprint arXiv:2605.04543*, 2026.

Brandon Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John Kitchin, Daniel Levine, Kyle Michel, Anuroop Sriram, Taco Cohen, Abhishek Das, Sushree Sahoo, Ammar Rizvi, Zachary Ulissi, and Larry Zitnick. UMA: A family of universal models for atoms. In *Advances in Neural Information Processing Systems*, volume 38, pages 129391–129427, 2025.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022.

Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. UFOgen: You forward once large scale text-to-image generation via diffusion GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yilun Xu, Weili Nie, and Arash Vahdat. One-step diffusion models with f-divergence distribution matching. *arXiv preprint arXiv:2502.15681*, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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

Meiyu Zhong, Noel Teku, and Ravi Tandon. Speeding up speculative decoding via sequential approximate verification. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*, 2025.

Zhen Zou, Xiaoxiao Ma, Jie Huang, Zichao Yu, and Feng Zhao. Fast-ARDiff: An entropy-informed acceleration framework for continuous space autoregressive generation. *arXiv preprint arXiv:2512.08537*, 2025.

APPENDIX CONTENTS

A Practical Considerations **16**

B Full Block Verification Algorithm **16**

C General Results **17**

D Decomposition of LLM Residual **21**

E Block Verification Proof **23**

F Limits of Reflection-Style Deterministic Residuals for Block Verification **25**

G Block Verification Complexity **26**

H Frozen and Free Drafters **27**

I Temperature **28**

J Experimental details **34**

K Additional Experiments **35**

L Acceleration of Molecular Dynamics Integrators **38**

A Practical Considerations

A.1 Sampling the Block Verification Residual

A key part of our block verification algorithm (Algo. 6) is sampling Ψ_α ((8)) to sample a correct residual function. Practically, we leverage a bisection search. In addition, we utilize a doubling procedure to find appropriate bounds for our search space. Notice that the condition of $\Psi_\alpha(u) \neq 1$ already gives our bisection search an upper bound. So we only need to heuristically find an appropriate lower bound. This is summarized in Algo. 7.

Algorithm 7: `USampleViaBisection`$(\|\Delta\|, \alpha)$

Require: $\|\Delta\| > 0$, $\alpha \in (0, 1]$, error tolerance `tol` > 0

Sample $u \sim \text{Unif}[0, 1]$.
▷ Bounds.
Initialize bounds $(u_{\text{lower}}, u_{\text{upper}}) = (\log(\alpha)/\|\Delta\| + \|\Delta\|/2 - 1, \log(\alpha)/\|\Delta\| + \|\Delta\|/2)$.
while $\Psi_\alpha(u_{\text{lower}}) > u$ **do**
    Set $u_{\text{lower}} = u_{\text{upper}} - 2(u_{\text{upper}} - u_{\text{lower}})$.
end while
▷ Bisection search.
Initialize $u_{\text{mid}} = (u_{\text{lower}} + u_{\text{upper}})/2$.
while $|\Psi_\alpha(u_{\text{mid}}) - u| >$ `tol` **do**
    if $\Psi_\alpha(u_{\text{mid}}) < u$ **then**
        Set $u_{\text{lower}} = u_{\text{mid}}$.
    else
        Set $u_{\text{upper}} = u_{\text{mid}}$.
    end if
    Re-calculate $u_{\text{mid}} = (u_{\text{lower}} + u_{\text{upper}})/2$.
end while
return u_{mid}.

A.2 Density Ratio Calculations

In the verification algorithms presented (Algos. 2 and 6), we calculate density ratios of distributions (Gaussians). The algorithms are written in probability space to match the standard speculative decoding presentation. However, in practice, for additional numerical stability one can calculate these ratios in log space. Notice that

$$\log \frac{\mathcal{N}(\mathsf{Z} + \Delta; 0, \mathrm{Id})}{\mathcal{N}(\mathsf{Z}; 0, \mathrm{Id})} = -\mathsf{Z}^\top \Delta - \frac{1}{2}\|\Delta\|^2. \tag{11}$$

Thus, for example, the block recursion in Algo. 6 can be simplified to

$$\log \alpha_j = \min\left\{0, \log \alpha_{j-1} - \mathsf{Z}_j^\top \Delta_j - \frac{1}{2}\|\Delta_j\|^2\right\}. \tag{12}$$

Likewise, when computing the differences of scalar c.d.f.s $\alpha\Phi(a) - \Phi(b)$, one should utilize log-c.d.f. primitives (*e.g.*, `scipy.stats.norm.logcdf`) alongside other stable operators, *e.g.*, using `numpy.log1p` to define a stable log-sub-exp operation, for numerical stability.

B Full Block Verification Algorithm

We present the full block verification speculative diffusion pseudo-code in Algo. 8.

Algorithm 8: Speculative Diffusions with Block Verification

Require: Target model q, drafter model p, draft size γ, number of sampling steps K.

Sample $\mathsf{Y}_0 \sim \mathcal{N}(0, \mathrm{Id})$ and set $k = 0$.
while $k < K$ **do**
  Set $\gamma_k = \min\{\gamma, K - k\}$.
  ▷ Construct Draft.
  Initialize draft $\hat{\mathsf{Y}}_k = \mathsf{Y}_k$.
  for $j \in \{1, \dots, \gamma_k\}$ **do**
    Sample $\hat{\mathsf{Y}}_{k+j} \sim p(\cdot \mid \hat{\mathsf{Y}}_{k:k+j-1})$. ▷ Cache draft mean m^p_{k+j-1}.
  end for

  ▷ Verify Draft.
  Compute $m^q_{k+j-1} = m^q_{k+j-1}(\hat{\mathsf{Y}}_{k+j-1})$ for all $j \in \{1, \dots, \min\{\gamma_k + 1, K - k\}\}$ in parallel.
  Initialize $\boldsymbol{c}_0 = 1$ and $\alpha_0 = 1$.
  for $j \in \{1, \dots, \gamma_k\}$ **do**
    Set $\Delta_j = (m^p_{k+j-1} - m^q_{k+j-1})/\sigma_{k+j-1}$ and $\mathsf{Z}_j = (\hat{\mathsf{Y}}_{k+j} - m^p_{k+j-1})/\sigma_{k+j-1}$.
    Calculate $\alpha_j = \min\left\{1, \alpha_{j-1}\frac{\mathcal{N}(\mathsf{Z}_j + \Delta_j; 0, \mathrm{Id})}{\mathcal{N}(\mathsf{Z}_j; 0, \mathrm{Id})}\right\}$.
  end for
  for $j \in \{1, \dots, \gamma_k\}$ **do**
    if $j \neq \gamma_k$ **then**
      Calculate $h_j = v_j/(v_j + 1 - \alpha_j)$, where
      $$v_j = \alpha_j \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} + \frac{\|\Delta_{j+1}\|}{2}\right) - \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} - \frac{\|\Delta_{j+1}\|}{2}\right).$$
    else
      Set $h_{\gamma_k} = \alpha_{\gamma_k}$.
    end if
    Flip coin $\boldsymbol{c}_j = [\![\eta < h_j]\!]$, where $\eta \sim \mathrm{Unif}[0, 1]$.
  end for
  Set $\tau = 1 + \max\{j \in \{0, 1, \dots, \gamma_k\} \mid \boldsymbol{c}_j = 1\}$.

  ▷ Residual Calculation.
  if $k + \tau \leq K$ **then**
    if $\tau = \gamma_k + 1$ **then**
      Sample $\mathsf{Y}_{k+\gamma_k+1} \sim q(\cdot \mid \hat{\mathsf{Y}}_{k+\gamma_k})$.
    else
      Set $\Delta = \Delta_\tau$, $\mathsf{Z} = \mathsf{Z}_\tau$, $m^q = m^q_{k+\tau-1}$, $\sigma = \sigma_{k+\tau-1}$, $\alpha_{\text{res}} = \alpha_{\tau-1}$.
      Sample $\mathsf{U} \sim \psi_{\alpha_{\tau-1}}$ via `USampleViaBisection`$(\|\Delta\|, \alpha_{\text{res}})$ (Algo. 7).
      Sample $\mathsf{G} \sim \mathcal{N}(0, \mathrm{Id})$.
      Set $\mathsf{Z}_\perp = \mathsf{G} - e(e^\top \mathsf{G})$, where $e = \Delta/\|\Delta\|$ so that $\mathsf{Z}_\perp \sim \mathcal{N}(0, \mathrm{Id} - ee^\top)$.
      Set $\mathsf{Y}_{k+\tau} = m^q + \sigma(\mathsf{U}e + \mathsf{Z}_\perp)$.
    end if
  end if
  Set $\mathsf{Y}_{k+j} = \hat{\mathsf{Y}}_{k+j}$ for $j \in \{1, \dots, \tau - 1\}$.
  Set $k = \min\{k + \tau, K\}$.
end while
return $\mathsf{Y}_{0:K}$.

C General Results

In the following, we consider a variety of general results. We prove these in the most general way possible, *i.e.*, with temperature and with "prior" probability α.

Accept Ratio The acceptance probability of speculative diffusion is calculated via clamping a density ratio. It is often useful to consider under what domains we should clamp the ratio.

Lemma C.1. *Let* $p^\omega(y) = \mathcal{N}(m^p, \omega^2\sigma^2\mathrm{Id})$ *and* $q^\omega(y) = \mathcal{N}(m^q, \omega^2\sigma^2\mathrm{Id})$. *Denote* $\Delta = (m^p - m^q)/\sigma$ *and* $e = \Delta/\|\Delta\|$. *Furthermore, let* $\alpha \in (0, 1]$ *and* $\|\Delta\| > 0$.

Then for $y = m^p + \sigma z^p$, we have

$$\alpha \frac{q^\omega(y)}{p^\omega(y)} < 1 \quad \iff \quad (z^p)^\top e > \frac{\omega^2 \log \alpha}{\|\Delta\|} - \frac{\|\Delta\|}{2}; \tag{13}$$

and for $y = m^q + \sigma z^q$, we have

$$\alpha \frac{q^\omega(y)}{p^\omega(y)} < 1 \quad \iff \quad (z^q)^\top e > \frac{\omega^2 \log \alpha}{\|\Delta\|} + \frac{\|\Delta\|}{2}. \tag{14}$$

Proof. After considering the change of variables $y = m^p + \sigma z$, we have:

$$\begin{aligned} \alpha \frac{\mathcal{N}(z + \Delta; 0, \omega^2 \mathrm{Id})}{\mathcal{N}(z; 0, \omega^2 \mathrm{Id})} < 1 \quad &\iff \quad \alpha \exp\left(-\frac{1}{2\omega^2} \left(\|\Delta\|^2 + 2z^\top \Delta \right) \right) < 1 \\ &\iff \quad \log \alpha - \frac{1}{2\omega^2} \left(\|\Delta\|^2 + 2z^\top \Delta \right) < 0 \\ &\iff \quad 2\omega^2 \log \alpha - \|\Delta\|^2 - 2z^\top \Delta < 0. \end{aligned}$$

This yields the stated constraint. The other case follows identically with a modified sign. □

Typically, we will leave the superscript on z implicit (where the most common choice is $z = z^p$ as this corresponds to the noise of the draft model in the speculative diffusions algorithm).

Non-zero Residual Additionally, the residual term depends on clamping the difference of distributions. It is also useful to know when this difference is non-zero.

Corollary C.2. *Let $p(y) = \mathcal{N}(m^p, \sigma^2 \mathrm{Id})$ and $q(y) = \mathcal{N}(m^q, \sigma^2 \mathrm{Id})$. Denote $\Delta = (m^p - m^q)/\sigma$ and $e = \Delta / \|\Delta\|$. Furthermore, let $\alpha \in (0, 1]$ and $\|\Delta\| > 0$.*

Then for $y = m^p + \sigma z^p$, we have

$$\alpha q(y) - p(y) > 0 \quad \iff \quad (z^p)^\top e < \frac{\log \alpha}{\|\Delta\|} - \frac{\|\Delta\|}{2}; \tag{15}$$

and for $y = m^q + \sigma z^q$, we have

$$\alpha q(y) - p(y) > 0 \quad \iff \quad (z^q)^\top e < \frac{\log \alpha}{\|\Delta\|} + \frac{\|\Delta\|}{2}. \tag{16}$$

Closed-form Acceptance Probability The sample acceptance probability can be calculated in closed form. This involves integrating the acceptance probability with the draft measure.

Lemma C.3. *Let $p^\omega(y) = \mathcal{N}(m^p, \omega^2 \sigma^2 \mathrm{Id})$ and $q^\omega(y) = \mathcal{N}(m^q, \omega^2 \sigma^2 \mathrm{Id})$ with $p = p^{\omega=1}$. Furthermore, let $\alpha \in (0, 1]$ and $\|\Delta\| > 0$. Then*

$$\begin{aligned} P_{\text{accept}} &= \int p(y) \min\left\{ 1, \alpha \frac{q^\omega(y)}{p^\omega(y)} \right\} \mathrm{d}y \\ &= \Phi\left(\frac{\omega^2 \log \alpha}{\|\Delta\|} - \frac{\|\Delta\|}{2} \right) + \alpha \exp\left(\frac{\|\Delta\|^2}{2\omega^2} \left(\frac{1}{\omega^2} - 1 \right) \right) \left(1 - \Phi\left(\frac{\omega^2 \log \alpha}{\|\Delta\|} - \frac{\|\Delta\|}{2} + \frac{\|\Delta\|}{\omega^2} \right) \right), \end{aligned} \tag{17}$$

where Φ is the standard normal c.d.f.

Proof. We consider a change of variables of $y = m^p + \sigma z$. Then we have

$$P_{\text{accept}} = \int \mathcal{N}(z, 0, \mathrm{Id}) \min\left\{ 1, \alpha \frac{\mathcal{N}(z + \Delta; 0, \omega^2 \mathrm{Id})}{\mathcal{N}(z; 0, \omega^2 \mathrm{Id})} \right\} \mathrm{d}z.$$

We will evaluate this in cases depending on which value the minimum is evaluated as. From Lem. C.1, the second argument of the min is evaluated when

$$z^\top e > \frac{\omega^2 \log \alpha}{\|\Delta\|} - \frac{\|\Delta\|}{2} = A.$$

We define the region $S = \{z : z^\top e > A\}$.

We now have,

$$\begin{aligned} P_{\text{accept}} &= \int_{S^c} \mathcal{N}(z; 0, \mathrm{Id})\, \mathrm{d}z + \alpha \int_S \mathcal{N}(z; 0, \mathrm{Id}) \frac{\mathcal{N}(z+\Delta; 0, \omega^2 \mathrm{Id})}{\mathcal{N}(z; 0, \omega^2 \mathrm{Id})}\, \mathrm{d}z \\ &= \int_{S^c} \mathcal{N}(z; 0, \mathrm{Id})\, \mathrm{d}z + \alpha \int_S \mathcal{N}(z; 0, \mathrm{Id}) \exp\left(-\frac{\|\Delta\|}{2\omega^2}\left(\|\Delta\| + 2z^\top e\right)\right) \mathrm{d}z. \end{aligned}$$

Now we use a change of coordinates $z = ue + z_\perp$, where $u = e^\top z$. It can be shown that the resulting Jacobian is equal to 1. Hence,

$$\begin{aligned} P_{\text{accept}} &= \int \int_{-\infty}^{A} \mathcal{N}(ue + z_\perp; 0, \mathrm{Id})\, \mathrm{d}u\, \mathrm{d}z_\perp \\ &\quad + \alpha \int \int_{A}^{\infty} \mathcal{N}(ue + z_\perp; 0, \mathrm{Id}) \exp\left(-\frac{\|\Delta\|}{2\omega^2}\left(\|\Delta\| + 2u\right)\right) \mathrm{d}u\, \mathrm{d}z_\perp \\ &= \int \int_{-\infty}^{A} \mathcal{N}(u; 0, 1)\mathcal{N}(z_\perp; 0, \mathrm{Id})\, \mathrm{d}u\, \mathrm{d}z_\perp \\ &\quad + \alpha \int \int_{A}^{\infty} \mathcal{N}(u; 0, 1)\mathcal{N}(z_\perp; 0, \mathrm{Id}) \exp\left(-\frac{\|\Delta\|}{2\omega^2}\left(\|\Delta\| + 2u\right)\right) \mathrm{d}u\, \mathrm{d}z_\perp \\ &= \int_{-\infty}^{A} \mathcal{N}(u; 0, 1)\, \mathrm{d}u + \alpha \int_{A}^{\infty} \mathcal{N}(u; 0, 1) \exp\left(-\frac{\|\Delta\|}{2\omega^2}\left(\|\Delta\| + 2u\right)\right) \mathrm{d}u \\ &= \int_{-\infty}^{A} \mathcal{N}(u; 0, 1)\, \mathrm{d}u + \frac{\alpha}{\sqrt{2\pi}} \exp\left(-\frac{\|\Delta\|^2}{2\omega^2}\right) \int_{A}^{\infty} \exp\left(-\frac{1}{2}\left(u^2 + \frac{2\|\Delta\| u}{\omega^2}\right)\right) \mathrm{d}u \\ &= \int_{-\infty}^{A} \mathcal{N}(u; 0, 1)\, \mathrm{d}u + \frac{\alpha}{\sqrt{2\pi}} \exp\left(-\frac{\|\Delta\|^2}{2\omega^2}\right) \int_{A}^{\infty} \exp\left(-\frac{1}{2}\left(u + \frac{\|\Delta\|}{\omega^2}\right)^2 + \frac{\|\Delta\|^2}{2\omega^4}\right) \mathrm{d}u \\ &= \int_{-\infty}^{A} \mathcal{N}(u; 0, 1)\, \mathrm{d}u + \frac{\alpha}{\sqrt{2\pi}} \exp\left(-\frac{\|\Delta\|^2}{2\omega^2} + \frac{\|\Delta\|^2}{2\omega^4}\right) \int_{A}^{\infty} \exp\left(-\frac{1}{2}\left(u + \frac{\|\Delta\|}{\omega^2}\right)^2\right) \mathrm{d}u \\ &= \int_{-\infty}^{A} \mathcal{N}(u; 0, 1)\, \mathrm{d}u + \alpha \exp\left(\frac{\|\Delta\|^2}{2\omega^2}\left(\frac{1}{\omega^2} - 1\right)\right) \int_{A}^{\infty} \mathcal{N}\left(u + \frac{\|\Delta\|}{\omega^2}; 0, 1\right) \mathrm{d}u \\ &= \int_{-\infty}^{A} \mathcal{N}(u; 0, 1)\, \mathrm{d}u + \alpha \exp\left(\frac{\|\Delta\|^2}{2\omega^2}\left(\frac{1}{\omega^2} - 1\right)\right) \int_{A + \frac{\|\Delta\|}{\omega^2}}^{\infty} \mathcal{N}\left(u; 0, 1\right) \mathrm{d}u. \end{aligned}$$

Finally, writing the solution in terms of the standard normal c.d.f. finishes the proof. □

Orthogonal Decomposition of Noise A key technique we use in our proofs is the decomposition of a random variable $\mathsf{Z} \sim \mathcal{N}(0; \mathrm{Id})$ (in $\mathbb{R}^d$).

Lemma C.4. *Let e be a unit vector and $\mathsf{Z} \sim \mathcal{N}(0; \mathrm{Id})$. Define the orthogonal decomposition $\mathsf{Z} = \mathsf{U}e + \mathsf{Z}_\perp$, where we have $\mathsf{U} = e^\top \mathsf{Z}$ and $\mathsf{Z}_\perp = (\mathrm{Id} - ee^\top)\mathsf{Z}$. We have that*

$$\mathsf{U} \sim \mathcal{N}(0, 1) \qquad \text{and} \qquad \mathsf{Z}_\perp \sim \mathcal{N}(0, \mathrm{Id} - ee^\top). \tag{18}$$

Moreover, U and $\mathsf{Z}_\perp$ are independent.

Proof. Given the orthogonal decomposition $\mathsf{Z} = \mathsf{U}e + \mathsf{Z}_\perp$, where we have $\mathsf{U} = e^\top \mathsf{Z}$ and $\mathsf{Z}_\perp = (\mathrm{Id} - ee^\top)\mathsf{Z}$. We shorthand $P_\perp = \mathrm{Id} - ee^\top$.

We show that the random variable U has the following properties

$$\begin{aligned} \mathbb{E}[\mathsf{U}] &= \mathbb{E}[e^\top \mathsf{Z}] = e^\top \mathbb{E}[\mathsf{Z}] = 0 \\ \mathrm{Cov}[\mathsf{U}] &= \mathbb{E}[\mathsf{U}^2] - \mathbb{E}[\mathsf{U}]^2 = \mathbb{E}[(e^\top \mathsf{Z})^2] = e^\top \mathbb{E}[\mathsf{Z}\mathsf{Z}^\top] e = e^\top (\mathrm{Id}) e = 1. \end{aligned}$$

Furthermore, as U is a linear transform of a Gaussian random variable, we have that $\mathsf{U} \sim \mathcal{N}(0, 1)$.

Similarly, we consider the properties of $\mathsf{Z}_\perp$

$$\mathbb{E}[\mathsf{Z}_\perp] = \mathbb{E}[P_\perp \mathsf{Z}] = P_\perp \mathbb{E}[\mathsf{Z}] = 0$$
$$\mathrm{Cov}[\mathsf{Z}_\perp] = \mathbb{E}[\mathsf{Z}_\perp \mathsf{Z}_\perp^\top] - \mathbb{E}[\mathsf{Z}_\perp]\mathbb{E}[\mathsf{Z}_\perp]^\top = \mathbb{E}[\mathsf{Z}_\perp \mathsf{Z}_\perp^\top] = P_\perp \mathbb{E}[\mathsf{Z}\mathsf{Z}^\top] P_\perp^\top = P_\perp P_\perp^\top.$$

We further note that $P_\perp$ is a symmetric and idempotent matrix. That is

$$P_\perp P_\perp^\top = P_\perp P_\perp = (\mathrm{Id} - ee^\top)(\mathrm{Id} - ee^\top) = \mathrm{Id} - ee^\top - ee^\top + ee^\top ee^\top = \mathrm{Id} - ee^\top = P_\perp.$$

Thus $\mathrm{Cov}[\mathsf{Z}_\perp] = P_\perp$. Similar to U, $\mathsf{Z}_\perp$ is a linear transform of Z, thus $\mathsf{Z}_\perp \sim \mathcal{N}(0, P_\perp)$.

Finally, we compute the cross-covariance:

$$\begin{aligned}
\mathrm{Cov}(\mathsf{U}, \mathsf{Z}_\perp) &= \mathbb{E}[\mathsf{U}\mathsf{Z}_\perp^\top] - \mathbb{E}[\mathsf{U}]\mathbb{E}[\mathsf{Z}_\perp]^\top \\
&= \mathbb{E}[(e^\top \mathsf{Z})(P_\perp \mathsf{Z})^\top] \\
&= e^\top \mathbb{E}[\mathsf{Z}\mathsf{Z}^\top] P_\perp^\top \\
&= e^\top P_\perp \\
&= e^\top (\mathrm{Id} - ee^\top) \\
&= 0.
\end{aligned}$$

Therefore $(\mathsf{U}, \mathsf{Z}_\perp)$ is jointly Gaussian with zero cross-covariance, and hence U and $\mathsf{Z}_\perp$ are independent. □

Orthogonal Change of Variable A useful property of the orthogonal decomposition is that the corresponding change of coordinates has the following property.

Lemma C.5. *Let $e \in \mathbb{R}^d$ be a unit vector. Choose a matrix $E_\perp \in \mathbb{R}^{d\times(d-1)}$ whose columns form an orthonormal basis of $e^\perp = \{x \in \mathbb{R}^d \mid e^\top x = 0\}$. Then the map $T\colon \mathbb{R} \times \mathbb{R}^{d-1} \to \mathbb{R}^d$ given by $(u, v) \mapsto ue + E_\perp v$ is an orthogonal linear isomorphism.*

Hence for every integrable $f\colon \mathbb{R}^d \to \mathbb{R}$,

$$\int_{\mathbb{R}^d} f(z)\,\mathrm{d}z = \int_{\mathbb{R}} \int_{\mathbb{R}^{d-1}} f(ue + E_\perp v)\,\mathrm{d}v\,\mathrm{d}u. \tag{19}$$

Proof. Let $Q = [e, E_\perp] \in \mathbb{R}^{d\times d}$. Since e is a unit vector and $E_\perp$ is an orthonormal basis of $e^\perp$, Q is orthogonal, *i.e.*, $Q^\top Q = \mathrm{Id}$. Thus $|\det Q| = 1$.

Thus, every $z \in \mathbb{R}^d$ can be (uniquely) written as $z = ue + E_\perp v$, where $u \in \mathbb{R}$ and $v \in \mathbb{R}^{d-1}$. The result then follows from the usual change of variables formula:

$$\int_{\mathbb{R}^d} f(z)\,\mathrm{d}z = \int_{\mathbb{R}} \int_{\mathbb{R}^{d-1}} f(ue + E_\perp v)\,|\det Q|\,\mathrm{d}v\,\mathrm{d}u = \int_{\mathbb{R}} \int_{\mathbb{R}^{d-1}} f(ue + E_\perp v)\,\mathrm{d}v\,\mathrm{d}u.$$

As required. □

Combining Lems. C.4 and C.5, we get the following property.

Corollary C.6. *Let $f\colon \mathbb{R}^d \to \mathbb{R}$ and let e be a normalized vector. We consider a change of variables $z \mapsto ue + z_\perp$ where $u = e^\top z$ and $z_\perp = (\mathrm{Id} - ee^\top)z$. We have that*

$$\int f(z)\mathcal{N}(z; 0, \mathrm{Id})\,\mathrm{d}z = \iint f(ue + z_\perp)\mathcal{N}(u; 0, 1)\mathcal{N}(z_\perp; 0, \mathrm{Id} - ee^\top)\,\mathrm{d}u\,\mathrm{d}z_\perp. \tag{20}$$

Remark. In Cor. C.6, we abuse notation in the integration of $z_\perp$. More precisely, the integration should be w.r.t. the $(d-1)$-dimensional subspace $e^\perp = \{x \in \mathbb{R}^d \mid e^\top x = 0\}$. The integration $\mathrm{d}z_\perp$ should be understood as the $(d-1)$ dimensional Lebesgue measure on $e^\perp$, and $\mathcal{N}(z_\perp; 0, \mathrm{Id} - ee^\top)$ denote the Gaussian measure supported on $e^\perp$.

More precisely, if $E_\perp \in \mathbb{R}^{d\times(d-1)}$ has orthonormal columns spanning $e^\perp$, then writing $z_\perp = E_\perp v$ with $v \in \mathbb{R}^{d-1}$, the right-hand side of Cor. C.6 is rigorously defined as

$$\int_{\mathbb{R}} \int_{\mathbb{R}^{d-1}} f(ue + E_\perp v)\mathcal{N}(u; 0, 1)\mathcal{N}(v; 0, \mathrm{Id}_{d-1})\,\mathrm{d}v\,\mathrm{d}u.$$

For notational simplicity, in the sequel, we will utilize the simpler notation of Cor. C.6 and continue to write the integral in terms of $z_\perp$.

D DECOMPOSITION OF LLM RESIDUAL

Proposition D.1. *Let $p = \mathcal{N}(m^p, \sigma^2\mathrm{Id})$, $q = \mathcal{N}(m^q, \sigma^2\mathrm{Id})$, $\alpha \in (0, 1]$, and $\|\Delta\| > 0$. The distribution $r(y; \alpha) \propto \max\{0, \alpha q(y) - p(y)\}$ can be sampled by the following:*

$$\mathsf{U} \sim \psi_\alpha \tag{21}$$

$$\mathsf{Z}_\perp \sim \mathcal{N}(0, \mathrm{Id} - ee^\top) \tag{22}$$

$$\mathsf{Y} = m^q + \sigma(\mathsf{U}e + \mathsf{Z}_\perp) \tag{23}$$

where

$$\psi_\alpha(u) \propto \max\{0, \alpha\mathcal{N}(u; 0, 1) - \mathcal{N}(u - \|\Delta\|; 0, 1)\}. \tag{24}$$

Proof. Given an input y, we consider a reparametrization $y = m^q + \sigma z$. Thus for a random variable $\mathsf{Y} \sim r(\cdot; \alpha)$, we have that

$$\mathsf{Y} = m^q + \sigma\mathsf{Z}. \tag{25}$$

We will examine the distribution of Z. We denote the normalization term of $r(\cdot; \alpha)$ as $\mathcal{Z}_\alpha$. As all but Z are constants here, we have

$$\begin{aligned}
r(y; \alpha) &= \frac{1}{\mathcal{Z}_\alpha} \max\{0, \alpha q(y) - p(y)\} \\
&= \frac{1}{\mathcal{Z}_\alpha} \max\{0, \alpha\mathcal{N}(y; m^q, \sigma^2\mathrm{Id}) - \mathcal{N}(y; m^p, \sigma^2\mathrm{Id})\} \\
&= \frac{1}{\mathcal{Z}_\alpha} \max\{0, \alpha\mathcal{N}(m^q + \sigma z; m^q, \sigma^2\mathrm{Id}) - \mathcal{N}(m^q + \sigma z; m^p, \sigma^2\mathrm{Id})\} \\
&= \frac{1}{\tilde{\mathcal{Z}}_\alpha} \max\{0, \alpha\mathcal{N}(z; 0, \mathrm{Id}) - \mathcal{N}(z - \Delta; 0, \mathrm{Id})\},
\end{aligned}$$

where $\tilde{\mathcal{Z}}_\alpha$ absorbs the $1/\sigma$ change-of-variables constant.

With slight abuse of notation, we can now write $r(\cdot; \alpha)$ in terms of z as

$$r(z; \alpha) \propto \mathcal{N}(z; 0, \mathrm{Id}) \max\left\{0, \alpha - \frac{\mathcal{N}(z - \Delta; 0, \mathrm{Id})}{\mathcal{N}(z; 0, \mathrm{Id})}\right\}. \tag{26}$$

Notice that the ratio simplifies as

$$\begin{aligned}
\frac{\mathcal{N}(z - \Delta; 0, \mathrm{Id})}{\mathcal{N}(z; 0, \mathrm{Id})} &= \exp\left(-\frac{1}{2}\left(\|z - \Delta\|^2 - \|z\|^2\right)\right) \\
&= \exp\left(\Delta^\top z - \frac{\|\Delta\|^2}{2}\right) \\
&= \exp\left((e^\top z)\|\Delta\| - \frac{\|\Delta\|^2}{2}\right).
\end{aligned}$$

We will now consider an orthogonal decomposition with $e = \Delta/\|\Delta\|$, using Lem. C.4 with $\mathsf{Z} = \mathsf{U}e + \mathsf{Z}_\perp$, for $\mathsf{U} = e^\top\mathsf{Z}$ and $\mathsf{Z}_\perp = P_\perp\mathsf{Z}$. We shorthand $P_\perp = \mathrm{Id} - ee^\top$. Notice that the ratio can be solely written in terms of U. In particular, we get

$$\begin{aligned}
r(z; \alpha) = r(u; \alpha)r(z_\perp; \alpha) &\propto \mathcal{N}(z; 0, \mathrm{Id}) \max\left\{0, \alpha - \frac{\mathcal{N}(z - \Delta; 0, \mathrm{Id})}{\mathcal{N}(z; 0, \mathrm{Id})}\right\} \\
&= \mathcal{N}(u; 0, 1)\mathcal{N}(z_\perp; 0, P_\perp) \max\left\{0, \alpha - \exp\left(u\|\Delta\| - \frac{\|\Delta\|^2}{2}\right)\right\}.
\end{aligned}$$

Notice that the sampling of $\mathsf{Z}_\perp$ is independent of U. Hence we can sample $\mathsf{Z}_\perp$ separately.

Now examining U and its p.d.f. $r(u; \alpha)$, we have

$$\begin{aligned}
r(u; \alpha) &\propto \mathcal{N}(u; 0, 1) \max\left\{0, \alpha - \exp\left(u\|\Delta\| - \frac{\|\Delta\|^2}{2}\right)\right\} \\
&= \max\left\{0, \alpha\mathcal{N}(u; 0, 1) - \mathcal{N}(u; 0, 1)\exp\left(u\|\Delta\| - \frac{\|\Delta\|^2}{2}\right)\right\}.
\end{aligned}$$

Simplifying the last term, we get

$$\begin{aligned}\mathcal{N}(u;0,1)\exp\left(u\|\Delta\|-\frac{\|\Delta\|^2}{2}\right)&=\frac{1}{(2\pi)^{1/2}}\exp\left(-\frac{u^2}{2}\right)\exp\left(u\|\Delta\|-\frac{\|\Delta\|^2}{2}\right)\\&=\frac{1}{(2\pi)^{1/2}}\exp\left(-\frac{1}{2}\left(u^2-2u\|\Delta\|+\|\Delta\|^2\right)\right)\\&=\frac{1}{(2\pi)^{1/2}}\exp\left(-\frac{1}{2}\left(u-\|\Delta\|\right)^2\right)\\&=\mathcal{N}(u-\|\Delta\|;0,1).\end{aligned}$$

Combining together, we have that $r(u;\alpha)=\psi_\alpha(u)\propto\max\{0,\alpha\mathcal{N}(u;0,1)-\mathcal{N}(u-\|\Delta\|;0,1)\}$. Further combining the reparameterization of Y, independent sampling U and $\mathsf{Z}_\perp$, yields the desired result. □

Lemma D.2. *Let $\psi_\alpha(u)\propto\max\{0,\alpha\mathcal{N}(u;0,1)-\mathcal{N}(u-v;0,1)\}$ for $\alpha\in(0,1]$ and $v>0$. The c.d.f. of ψ_α is given by*

$$\Psi_\alpha(u)=\begin{cases}\frac{\alpha\Phi(u)-\Phi(u-v)}{\alpha\Phi(\frac{\log\alpha}{v}+\frac{v}{2})-\Phi(\frac{\log\alpha}{v}-\frac{v}{2})} & \text{if } u<\frac{\log\alpha}{v}+\frac{v}{2}\\ 1 & \text{otherwise},\end{cases}\tag{27}$$

where Φ corresponds to the c.d.f. of the standard normal distribution.

Proof. The p.d.f. of ψ_α is given by

$$\psi_\alpha(u)=\frac{\max\left\{0,\alpha\mathcal{N}(u;0,1)-\mathcal{N}(u-v;0,1)\right\}}{\int_{-\infty}^{\infty}\max\left\{0,\alpha\mathcal{N}(u';0,1)-\mathcal{N}(u'-v;0,1)\right\}\,\mathrm{d}u'},\tag{28}$$

where the denominator is strictly positive by definition.

Note that from a modification of the proof of Cor. C.2, ψ_α is non-zero iff

$$\alpha>\frac{\mathcal{N}(u-v;0,1)}{\mathcal{N}(u;0,1)}\iff uv<\log\alpha+\frac{v^2}{2}\iff u<\frac{\log\alpha}{v}+\frac{v}{2}.$$

Notably, the c.d.f. Ψ_α will be 1 whenever u is larger than this upper bound. Otherwise, we can consider the case where this bound holds. We have that the c.d.f. in this case will be

$$\begin{aligned}\Psi_\alpha(u)&=\frac{\int_{-\infty}^{u}\max\left\{0,\alpha\mathcal{N}(u';0,1)-\mathcal{N}(u'-v;0,1)\right\}\,\mathrm{d}u'}{\int_{-\infty}^{\infty}\max\left\{0,\alpha\mathcal{N}(u';0,1)-\mathcal{N}(u'-v;0,1)\right\}\,\mathrm{d}u'}\\&=\frac{\int_{-\infty}^{u}\alpha\mathcal{N}(u';0,1)-\mathcal{N}(u'-v;0,1)\,\mathrm{d}u'}{\int_{-\infty}^{\frac{\log\alpha}{v}+\frac{v}{2}}\alpha\mathcal{N}(u';0,1)-\mathcal{N}(u'-v;0,1)\,\mathrm{d}u'}.\end{aligned}$$

We calculate the numerator and denominator separately.

For the numerator, we get

$$\begin{aligned}\int_{-\infty}^{u}\alpha\mathcal{N}(u';0,1)-\mathcal{N}(u'-v;0,1)\,\mathrm{d}u'&=\int_{-\infty}^{u}\alpha\mathcal{N}(u';0,1)\,\mathrm{d}u'-\int_{-\infty}^{u}\mathcal{N}(u'-v;0,1)\,\mathrm{d}u'\\&=\alpha\Phi(u)-\Phi(u-v).\end{aligned}$$

Then similarly for the denominator,

$$\begin{aligned}&\int_{-\infty}^{\frac{\log\alpha}{v}+\frac{v}{2}}\alpha\mathcal{N}(u';0,1)-\mathcal{N}(u'-v;0,1)\,\mathrm{d}u'\\&=\int_{-\infty}^{\frac{\log\alpha}{v}+\frac{v}{2}}\alpha\mathcal{N}(u';0,1)\,\mathrm{d}u'-\int_{-\infty}^{\frac{\log\alpha}{v}+\frac{v}{2}}\mathcal{N}(u'-v;0,1)\,\mathrm{d}u'\\&=\alpha\Phi\left(\frac{\log\alpha}{v}+\frac{v}{2}\right)-\Phi\left(\frac{\log\alpha}{v}-\frac{v}{2}\right).\end{aligned}$$

Together, alongside considering the constraint on u, yields the result. □

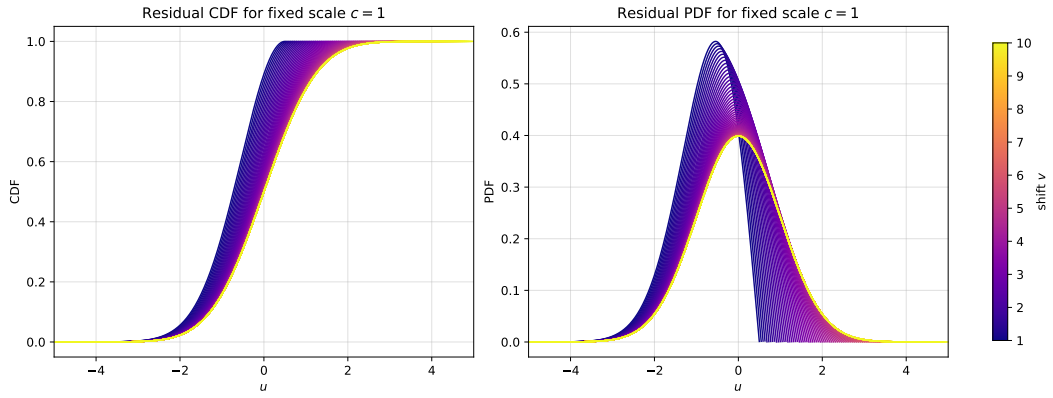

Figure 1: Plot of the p.d.f. $f(u) \propto \max\{0, c\mathcal{N}(u; 0, 1) - \mathcal{N}(u - v; 0, 1)\}$ and its c.d.f. for fixed scale $c = 1$ over various shifts v. Normalization approximated via trapezoid integration.

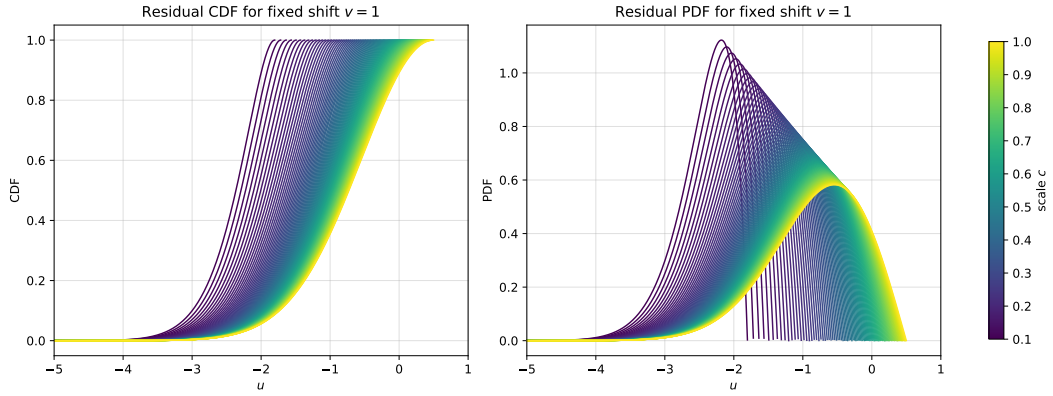

Figure 2: Plot of the PDF $f(u) \propto \max\{0, c\mathcal{N}(u; 0, 1) - \mathcal{N}(u - v; 0, 1)\}$ and its c.d.f. for fixed shift $v = 1$ over various scales c. Normalization approximated via trapezoid integration.

Examples of Residual Plot Comparison

The residuals provided by Algos. 4 and 5 are fundamentally different. Indeed, the reflection coupling gives a deterministic correction, while the decomposition corresponds to a distribution made from the difference of Gaussian distributions (normalized and non-zero). In Figs. 1 and 2, we show a 1D example of what the decomposition/LLM residual r_Γ looks like for Gaussian distributions, *i.e.*, the diffusion model case. Fig. 1 fixes the scale $c = 1$ and varies the shift v, which corresponds to the mismatch between the drafter and primary models. Fig. 2 fixes $v = 1$ and varies the scale c (equivalently α), which is relevant for block verification.

E Block Verification Proof

Proposition E.1. *Given the input of block verification Algo. 6, the acceptance probability of block verification for $j \in \{1, \dots, \gamma - 1\}$ can be calculated as*

$$h_j = \frac{\alpha_j \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} + \frac{\|\Delta_{j+1}\|}{2}\right) - \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} - \frac{\|\Delta_{j+1}\|}{2}\right)}{\alpha_j \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} + \frac{\|\Delta_{j+1}\|}{2}\right) - \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} - \frac{\|\Delta_{j+1}\|}{2}\right) + 1 - \alpha_j}, \tag{29}$$

where Δ_{j+1} and α_j are values calculated in a round of speculative diffusions, as per Algo. 6. For $j = \gamma$, the algorithm sets $h_\gamma = \alpha_\gamma$.

Proof. Without loss of generality, we assume that we are examining step j in a round starting at index k, *i.e.*, we are considering the $k + j + 1$ sample. We denote the draft sequence by $\hat{y}_{k:k+j}$ up to

this step. Through adapting Sun et al. (2025, Equation (5)) to a continuous state space, we get

$$h_j = \frac{\int_{\mathbb{R}^d} \max\{0, \alpha_j q(\hat{y} \mid \hat{y}_{k+j}) - p(\hat{y} \mid \hat{y}_{k:k+j})\} \, \mathrm{d}\hat{y}}{\int_{\mathbb{R}^d} \max\{0, \alpha_j q(\hat{y} \mid \hat{y}_{k+j}) - p(\hat{y} \mid \hat{y}_{k:k+j})\} \, \mathrm{d}\hat{y} + 1 - \alpha_j} \tag{30}$$

where d corresponds to the dimension of our model; we remind that the target model is Markov, but the draft is not necessarily Markov.

We note that (30) can be rewritten in terms of its denoising difference Δ_{j+1} (which depends on "history" $\hat{y}_{k:k+j}$), its variance σ^2_{k+j}, and the sampled noise of the drafter at that step z_{k+j}. For simplicity, we will drop all indices.

Key to our derivation is the simplification of the numerator. To do so, we focus on the region in which the inner term of the integral is positive by utilizing Cor. C.2. In particular, we utilize a change of coordinates via $\hat{y} = m^p + \sigma z$. We designate $S = \{z : z^\top e < A\}$, where $A = \log \alpha / \|\Delta\| - \|\Delta\|/2$. Thus we have,

$$\begin{aligned}\int_{\mathbb{R}^d} \max\{0, \alpha q(\hat{y} \mid \hat{y}_{k+j}) - p(\hat{y} \mid \hat{y}_{k:k+j})\} \, \mathrm{d}\hat{y} &= \int_S \alpha \mathcal{N}(\hat{y}; m^q, \sigma^2 \mathrm{Id}) - \mathcal{N}(\hat{y}; m^p, \sigma^2 \mathrm{Id}) \, \mathrm{d}\hat{y} \\ &= \int_S \alpha \mathcal{N}(z + \Delta; 0, \mathrm{Id}) - \mathcal{N}(z; 0, \mathrm{Id}) \, \mathrm{d}z.\end{aligned}$$

Now we consider the orthogonal decomposition of $z = (e^\top z)e + z_\perp$, noting that $e^\top z$ appears in the condition of S. We denote $u = e^\top z$. Thus, we can simplify

$$\begin{aligned}&\int_S \alpha \mathcal{N}(z + \Delta; 0, \mathrm{Id}) - \mathcal{N}(z; 0, \mathrm{Id}) \, \mathrm{d}z \\ &= \int_S \alpha \mathcal{N}(ue + \|\Delta\| e + z_\perp; 0, \mathrm{Id}) - \mathcal{N}(ue + z_\perp; 0, \mathrm{Id}) \, \mathrm{d}z \\ &= \iint_S \alpha \mathcal{N}(ue + \|\Delta\| e + z_\perp; 0, \mathrm{Id}) - \mathcal{N}(ue + z_\perp; 0, \mathrm{Id}) \, \mathrm{d}u \, \mathrm{d}z_\perp \\ &= \int_{-\infty}^A \int_{\mathbb{R}^d} \alpha \mathcal{N}((u + \|\Delta\|)e + z_\perp; 0, \mathrm{Id}) - \mathcal{N}(ue + z_\perp; 0, \mathrm{Id}) \, \mathrm{d}z_\perp \, \mathrm{d}u.\end{aligned}$$

We shorthand $P_\perp = \mathrm{Id} - ee^\top$. Now, breaking down each integral via Lem. C.4, we have

$$\begin{aligned}&\int_{-\infty}^A \int_{\mathbb{R}^d} \alpha \mathcal{N}((u + \|\Delta\|)e + z_\perp; 0, \mathrm{Id}) - \mathcal{N}(ue + z_\perp; 0, \mathrm{Id}) \, \mathrm{d}z_\perp \, \mathrm{d}u \\ &= \alpha \int_{-\infty}^A \int_{\mathbb{R}^d} \mathcal{N}((u + \|\Delta\|)e + z_\perp; 0, \mathrm{Id}) \, \mathrm{d}z_\perp \, \mathrm{d}u - \int_{-\infty}^A \int_{\mathbb{R}^d} \mathcal{N}(ue + z_\perp; 0, \mathrm{Id}) \, \mathrm{d}z_\perp \, \mathrm{d}u \\ &= \alpha \int_{-\infty}^{A+\|\Delta\|} \int_{\mathbb{R}^d} \mathcal{N}(ue + z_\perp; 0, \mathrm{Id}) \, \mathrm{d}z_\perp \, \mathrm{d}u - \int_{-\infty}^A \int_{\mathbb{R}^d} \mathcal{N}(ue + z_\perp; 0, \mathrm{Id}) \, \mathrm{d}z_\perp \, \mathrm{d}u \\ &= \alpha \int_{-\infty}^{A+\|\Delta\|} \int_{\mathbb{R}^d} \mathcal{N}(u; 0, 1) \mathcal{N}(z_\perp; 0, P_\perp) \, \mathrm{d}z_\perp \, \mathrm{d}u - \int_{-\infty}^A \int_{\mathbb{R}^d} \mathcal{N}(u; 0, 1) \mathcal{N}(z_\perp; 0, P_\perp) \, \mathrm{d}z_\perp \, \mathrm{d}u \\ &= \alpha \int_{-\infty}^{A+\|\Delta\|} \mathcal{N}(u; 0, 1) \, \mathrm{d}u - \int_{-\infty}^A \mathcal{N}(u; 0, 1) \, \mathrm{d}u \\ &= \alpha \Phi(A + \|\Delta\|) - \Phi(A).\end{aligned}$$

Substituting our definition of A, we get

$$\int_{\mathbb{R}^d} \max\{0, \alpha q(\hat{y} \mid \hat{y}_{k+j}) - p(\hat{y} \mid \hat{y}_{k:k+j})\} \, \mathrm{d}\hat{y} = \alpha \Phi\left(\frac{\log \alpha}{\|\Delta\|} + \frac{\|\Delta\|}{2}\right) - \Phi\left(\frac{\log \alpha}{\|\Delta\|} - \frac{\|\Delta\|}{2}\right).$$

Plugging this simplification into our equation of h_j (with $\Delta = \Delta_{j+1}$ and $\alpha = \alpha_j$) proves the result. □

F Limits of Reflection-Style Deterministic Residuals for Block Verification

Proposition F.1. *Fix $\gamma \geq 2$. There does not exist a valid deterministic residual correction of the form $t(\mathsf{Z})$ for block verification that simultaneously*

1. *is invertible; and*
2. *admits a constant Jacobian factor under change of variables.*

Proof. Suppose that $\gamma \geq 2$ and $c_{\gamma-1} = 1$. To derive a contradiction, assume there exists a deterministic residual correction t satisfying (i) and (ii). We note that for a residual to be valid, we additionally require t to yield the primary model q for every possible value of $\alpha_{\gamma-1} \in (0, 1]$.

From our assumptions, for the γ index, we have that

$$\mathsf{Y}_{k+\gamma} = m^q_{k+\gamma-1} + \sigma_{k+\gamma-1} \begin{cases} \mathsf{Z} + \Delta_\gamma & \text{with probability } h_\gamma(\mathsf{Z}) \\ t(\mathsf{Z}) & \text{with probability } 1 - h_\gamma(\mathsf{Z}) \end{cases}, \tag{31}$$

where with some abuse of notation we define $h_\gamma(z) = \min\{1, \alpha_{\gamma-1}\mathcal{N}(z + \Delta_\gamma; 0, \mathrm{Id})/\mathcal{N}(z; 0, \mathrm{Id})\}$ (note that $h_\gamma = \alpha_\gamma$ in Algo. 6), which depends on the randomization of the last draft element. We consider where the draft and target diffusions are non-trivially different: $\|\Delta_\gamma\| \neq 0$. To ensure that the output of block verification follows the distribution of q, we require

$$\begin{cases} \mathsf{Z} + \Delta_\gamma & \text{with probability } h_\gamma(\mathsf{Z}) \\ t(\mathsf{Z}) & \text{with probability } 1 - h_\gamma(\mathsf{Z}) \end{cases} \sim \mathcal{N}(0, \mathrm{Id}). \tag{32}$$

Let us evaluate the law of the left-hand side. By assumption (i), the inverse map t^{-1} exists, and by assumption (ii) the change of variables through t^{-1} contributes a constant Jacobian factor $C > 0$.

$$\begin{aligned} &\int [\![z = z' + \Delta_\gamma]\!] \mathcal{N}(z'; 0, \mathrm{Id}) h_\gamma(z') \mathrm{d}z' + \int [\![z = t(z')]\!] \mathcal{N}(z'; 0, \mathrm{Id})(1 - h_\gamma(z')) \mathrm{d}z' \\ &= \mathcal{N}(z - \Delta_\gamma; 0, \mathrm{Id}) h_\gamma(z - \Delta_\gamma) + C\mathcal{N}(t^{-1}(z); 0, \mathrm{Id})(1 - h_\gamma(t^{-1}(z))) \\ &= \min\{\mathcal{N}(z - \Delta_\gamma; 0, \mathrm{Id}), \alpha_{\gamma-1}\mathcal{N}(z; 0, \mathrm{Id})\} \\ &\quad + C \max\left\{0, \mathcal{N}(t^{-1}(z); 0, \mathrm{Id}) - \alpha_{\gamma-1}\mathcal{N}(t^{-1}(z) + \Delta_\gamma; 0, \mathrm{Id})\right\}, \end{aligned}$$

where $C > 0$ is the constant Jacobian factor. Now matching it with the law on the RHS, we have the equation

$$\begin{aligned} &\max\{\mathcal{N}(z; 0, \mathrm{Id}) - \mathcal{N}(z - \Delta_\gamma; 0, \mathrm{Id}), (1 - \alpha_{\gamma-1})\mathcal{N}(z; 0, \mathrm{Id})\} \\ &\quad = C \max\left\{0, \mathcal{N}(t^{-1}(z); 0, \mathrm{Id}) - \alpha_{\gamma-1}\mathcal{N}(t^{-1}(z) + \Delta_\gamma; 0, \mathrm{Id})\right\}. \end{aligned} \tag{33}$$

For this to be a valid residual, we require this to be true for all z and for every $\alpha_{\gamma-1} \in (0, 1)$.

Suppose $\alpha_{\gamma-1} < 1$. Then the LHS is strictly positive, *i.e.*, $(1 - \alpha_{\gamma-1})\mathcal{N}(z; 0, \mathrm{Id}) > 0$. Thus, we require for all z

$$\max\left\{0, \mathcal{N}(t^{-1}(z); 0, \mathrm{Id}) - \alpha_{\gamma-1}\mathcal{N}(t^{-1}(z) + \Delta_\gamma; 0, \mathrm{Id})\right\} > 0. \tag{34}$$

Hence, we require $\mathcal{N}(t^{-1}(z); 0, \mathrm{Id}) > \alpha_{\gamma-1}\mathcal{N}(t^{-1}(z) + \Delta_\gamma; 0, \mathrm{Id})$ for all z. This can be shown to be equivalent to the condition

$$\exp\left(\frac{1}{2}(2\Delta_\gamma^\top t^{-1}(z) + \|\Delta_\gamma\|^2)\right) > \alpha_{\gamma-1}. \tag{35}$$

However, because t is invertible, the image of t^{-1} must cover all possible z values. The quantity $\exp\left(\frac{1}{2}(2\Delta_\gamma^\top x + \|\Delta_\gamma\|^2)\right)$ ranges over $(0, \infty)$ as x varies over $\mathbb{R}^d$, so (35) cannot hold for every $x = t^{-1}(z)$ and every $\alpha_{\gamma-1} \in (0, 1]$. This contradiction proves that no deterministic residual correction satisfying (i) and (ii) exists. □

G Block Verification Complexity

Proposition G.1. *Assume the use of the Frozen Drafter and* $\mathrm{Tr}(\mathrm{Cov})[q_{\text{data}}] \leq \beta d$. *For a fixed draft size* γ, *let* $\rho(\gamma) \in [0,1]$ *denote the ratio of the expected number of rounds required to complete speculative diffusion with block verification relative to sample verification. Taking* $\gamma \asymp (K/\beta\delta d)^{1/3}$, *the expected number of parallel target model calls under block verification is at most* $\mathcal{O}(\rho(\gamma)K^{2/3}(\beta d\delta)^{1/3})$.

Proof. The result follows from a combination of prior results. We will first establish that the number of rounds to complete speculative diffusion is smaller when utilizing block verification rather than sample verification. Through this connection between sample and block verification, we utilize a prior result that directly bounds the number of rounds required to complete speculative diffusion with sample verification. Finally, we note that the expected number of primary calls is directly proportional to the number of rounds of speculative diffusion—specifically 2, one call for the Frozen Drafter and one (parallel) call for the verification.

Firstly, from Sun et al. (2025, Theorem 2) we know that the expected number of decoded samples after i many rounds of block verification speculative decoding is greater than its corresponding sample algorithm. Let $\mathsf{K}_{\text{sample}}(i)$ and $\mathsf{K}_{\text{block}}(i)$ be random variables that correspond to the number of denoising steps completed by speculative diffusion with sample and block verification, respectively. From Sun et al. (2025, Lemma 5), we know that for any $l \geq 0$

$$\mathbb{P}[\mathsf{K}_{\text{sample}}(i) \geq l] \leq \mathbb{P}[\mathsf{K}_{\text{block}}(i) \geq l]. \tag{36}$$

This result follows by marginalizing Sun et al. (2025, Lemma 5) over the context and output.

We will connect this result to the number of rounds of speculative diffusion required to be completed to complete the full denoising of K fixed steps. The number of rounds required for denoising will be denoted by the random variable $\mathsf{R}_{\text{sample}}$ and $\mathsf{R}_{\text{block}}$. It follows that we can connect the random variable associated with $\mathsf{K}_\cdot$ and $\mathsf{R}_\cdot$ by the following events:

$$\mathsf{R}_a > i \iff \mathsf{K}_a(i) < K, \tag{37}$$

where $a \in \{\text{sample}, \text{block}\}$. As a result, we have $\mathbb{P}[\mathsf{R}_a > i] = \mathbb{P}[\mathsf{K}_a(i) < K]$. Note that R_a is clearly a non-negative integer valued random variable. As a result, the expected number of rounds to denoise will be determined by a sum of tail probabilities. This gives us

$$\begin{aligned}
\mathbb{E}[\mathsf{R}_{\text{sample}}] &= \sum_{i=0}^{\infty} \mathbb{P}[\mathsf{R}_{\text{sample}} > i] \\
&= \sum_{i=0}^{\infty} \mathbb{P}[\mathsf{K}_{\text{sample}}(i) < K] \\
&= \sum_{i=0}^{\infty} (1 - \mathbb{P}[\mathsf{K}_{\text{sample}}(i) \geq K]) \\
&\geq \sum_{i=0}^{\infty} (1 - \mathbb{P}[\mathsf{K}_{\text{block}}(i) \geq K]) \\
&= \sum_{i=0}^{\infty} \mathbb{P}[\mathsf{K}_{\text{block}}(i) < K] \\
&= \sum_{i=0}^{\infty} \mathbb{P}[\mathsf{R}_{\text{block}} > i] = \mathbb{E}[\mathsf{R}_{\text{block}}],
\end{aligned}$$

where the inequality directly comes from (36).

Hence, for a fixed draft size γ, there exists a $\rho(\gamma) \in [0,1]$ such that

$$\rho(\gamma)\mathbb{E}[\mathsf{R}_{\text{sample}}] = \mathbb{E}[\mathsf{R}_{\text{block}}]. \tag{38}$$

Now with our given assumptions that $\mathrm{Tr}(\mathrm{Cov}[q_{\text{data}}]) \leq \beta d$, from the proof of Hu et al. (2025, Theorem 18), we have

$$\mathbb{E}[\mathsf{R}_{\text{sample}}] \lesssim \frac{K}{\gamma} + \sqrt{K\gamma\delta\beta d}. \tag{39}$$

With (38), this immediately gives us

$$\mathbb{E}[\mathsf{R}_{\text{block}}] \lesssim \rho(\gamma)\left(\frac{K}{\gamma} + \sqrt{K\gamma\delta\beta d}\right). \tag{40}$$

Finally, taking $\gamma = \lceil (K/\delta\beta d)^{1/3} \rceil$ yields the desired result. □

Remark. It should be noted that Prop. 3.3 simply converts the rate for sample verification into its block verification counterpart. However, it does not directly account for circumstances where the improvement in acceptance—quantified by $\rho(\gamma)$—can be extremely advantageous ($\rho(\gamma)$ small) for certain γ. Indeed, one can express the expected number of rounds for block verification while optimizing over the draft size γ. From (40) we have

$$\min_{\gamma} \mathbb{E}[\mathsf{R}^{\gamma}_{\text{block}}] \lesssim \left(K^{2/3}(\beta d\delta)^{1/3}\right) \min_{\gamma} \left\{ \rho(\gamma) \left(\frac{1}{\gamma} \left(\frac{K}{\beta d\delta} \right)^{1/3} + \sqrt{\gamma \left(\frac{\beta d\delta}{K} \right)^{1/3}} \right) \right\}, \tag{41}$$

where we make the dependence of γ for $\mathsf{R}^{\gamma}_{\text{block}}$ explicit.

It becomes apparent from (41) that the choice of $\gamma = \lceil (K/\delta\beta d)^{1/3} \rceil$ need not be optimal, and if $\rho(\gamma)$ has certain structure, block verification yields a more advantageous rate than Prop. 3.3 for an alternative draft size.

H Frozen and Free Drafters

In the following section, we prove Prop. 3.4. We begin by formally defining the block efficiency of speculative diffusion. Let $\texttt{be}(k)$ denote the average number of samples returned when the input of a round of speculative diffusion y_k corresponds to the kth denoising step. Note that $\texttt{be}(k) \in [1, \gamma+1]$—at worst we denoise a single (corrected) step and at best we accept the full draft and denoise an extra step with the primary model. We note that $\texttt{be}$ depends on the drafter p, the primary model q, and the exact acceptance/verification choice of the speculative diffusion algorithm. For the purpose of this section, we will leave the dependence of these on $\texttt{be}(k)$ implicit.

To calculate the speedup of speculative diffusion using different drafters (in idealized conditions), we consider a modification of the following formulation explored in Huang et al. (2025):

$$\texttt{Speedup}(k) = \texttt{be}(k)\left(\frac{\texttt{Time}(p,\gamma)}{\texttt{Time}(q,1)} + \frac{\texttt{PTime}(q,\gamma+1)}{\texttt{Time}(q,1)} + \frac{\texttt{Time}(r,1)}{\texttt{Time}(q,1)}\right)^{-1}, \tag{42}$$

where $\texttt{Time}(\cdot, c)$ corresponds to the wall-clock time of evaluating the drafter, primary model, or residual calculation sequentially for c values. $\texttt{PTime}$ corresponds to the parallel time.

For LLM speculative decoding, the drafter typically scales linearly in wall-clock time w.r.t. its return size, *i.e.*, $\texttt{Time}(p,\gamma) = \gamma\texttt{Time}(p,1)$.

In terms of dependencies, only the first term in (42) affects the speedup of speculative diffusion. The other terms can be considered "fixed" in our analysis of drafters. One typically wishes to choose a drafter p which maximizes $\texttt{be}(k)$ while minimizing $\texttt{Time}(p,\gamma)$.

In an ideal setting, one can assume that the wall-clock time for computing the residual is negligible $\texttt{Time}(r,1) \approx 0$. Furthermore, we can assume that the wall-clock time for computing $\gamma+1$ parallel calls to the primary model is roughly equivalent to a single sequential primary model call $\texttt{PTime}(q,\gamma+1) \approx \texttt{Time}(q,1)$. When we state "ideal conditions", we assume that:

$$\epsilon = \frac{\texttt{PTime}(q,\gamma+1) - \texttt{Time}(q,1)}{\texttt{Time}(q,1)} + \frac{\texttt{Time}(r,1)}{\texttt{Time}(q,1)}, \tag{43}$$

for some small constant $\epsilon > 0$.

Remark. The conditions are close to ideal whenever the cost of the primary model dominates all other times and $\texttt{PTime}(q,\gamma+1) \approx \texttt{Time}(q,1)$.

Proposition H.1. *Let $\texttt{be}(k)$ denote the block efficiency of a speculative diffusions algorithm with the Frozen Drafter at the kth denoising step. The speedup compared to no speculation in ideal conditions is equal to $\texttt{be}(k)/(2+\epsilon)$.*

Proof. In ideal conditions, the cost of the Frozen Drafter for an entire draft proposal of size γ is the single sequential call of the primary model. Thus we have $\texttt{Time}(p, \gamma) = \texttt{Time}(q, 1)$. Taking the other ideal conditions into account, (42) simplifies to $\texttt{be}(k)/(2+\epsilon)$, as required. □

The Free Drafter does not have the same $\approx 1/2$ scaling in ideal conditions.

Proposition H.2. *Let $\texttt{be}(k)$ denote the block efficiency of a speculative diffusion algorithm with the Free Drafter at the kth denoising step, for $k \geq 2$. The speedup compared to no speculative diffusion in ideal conditions is equal to $\texttt{be}(k)/(1+\epsilon'+\epsilon)$, where $\epsilon' > 0$ is the cost of computing Free Drafter,* i.e., $\epsilon' = \texttt{Time}(p, \gamma)/\texttt{Time}(q, 1)$.

Proof. In ideal conditions, the cost of Free Drafter is free for an entire draft proposal (the cost of the primary model has been absorbed by the previous round of speculative diffusion). Thus $\texttt{Time}(p, \gamma)/\texttt{Time}(q, 1) = \epsilon'$. Thus the speedup will be simplified to $\texttt{be}(k)/(1+\epsilon+\epsilon')$. □

This provides a simple corollary.

Corollary H.3. *Under ideal conditions, speculative diffusion with the Frozen Drafter is faster than speculative diffusion with the Free Drafter if*

$$\frac{\texttt{be}_{\text{frozen}}(k)}{\texttt{be}_{\text{free}}(k)} > \frac{2+\epsilon}{1+\epsilon+\epsilon'}. \tag{44}$$

From Cor. H.3, we can see that a necessary condition for the Frozen Drafter to be faster is that its block efficiency is close to twice as high as Free Drafter's counterpart.

Remark. It should however be noted that when using the Frozen Drafter, one will at least have a block efficiency of $\texttt{be}_{\text{frozen}}(k) \geq 2$. This is because the first draft image generated by the drafter will be exactly sampled by the primary model q (this is our single primary model call) and thus we will always accept it.

Experimentally, we show that the Free Drafter in practice can yield much better speedups than the Frozen Drafter. One reason—characteristic to speculative diffusion—is that the difficulty of accepting a draft image becomes progressively harder as we get closer to a clean image. Take sample verification for example, where the per-sample acceptance probability is given by one minus the total-variation between the draft and primary model. Indeed, from De Bortoli et al. (2025, Proposition 3.1), the acceptance probability for sample verification speculative diffusions will be

$$\text{TokenAcceptance}(k+1) = 2 - 2\Phi\left(\frac{1}{2}\mathcal{M}(m^q_{k+1}, m^p_{k+1}; \sigma^2_{k+1}\text{Id})\right), \tag{45}$$

where $\mathcal{M}^2(a, b; \Sigma) = (a-b)^\top \Sigma^{-1}(a-b)$ corresponds to the Mahalanobis distance with covariance Σ and we remind that Φ is the c.d.f. of the standard normal distribution.

As one can see, as $\sigma_k \to 0$, the Mahalanobis distance will approach ∞. This causes the acceptance rate to go to 0. Intuitively, when less noise is injected for each DDPM step, it becomes more difficult to accept a draft image. The block acceptance rate has a similar characteristic ($\alpha_k \to 0$). This is independent of the form of the drafter.

Fig. 3 presents the one-step lagged acceptance probability for sample verification on ImageNet over 250 denoising steps and churn $\varepsilon = 0.25$. This is equivalent to the second draft image acceptance probability of the Frozen Drafter or the first draft image acceptance probability of Free Drafter when the previous verification is sampled from the primary model (*i.e.*, a previous full draft). As one can see, the acceptance probability is high but eventually becomes 0 once we approach the end of denoising. In such a case, we are mostly rejecting, thus the larger cost of the Frozen Drafter will heavily affect the speed of speculative diffusion, without its benefit of increased acceptance rate.

I TEMPERATURE

As a form of lenience for speculative diffusions, one can consider applying a temperature term to sample and block verification (De Bortoli et al., 2025). The primary motivation of utilizing a

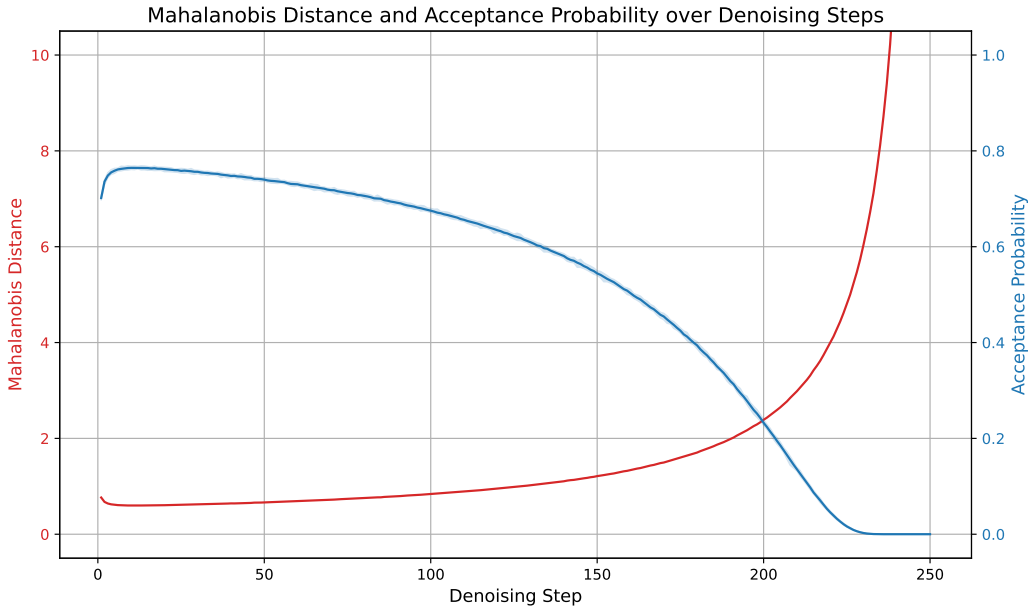

Figure 3: Sample verification acceptance probability on the ImageNet LDM dataset. This plot uses a sampling configuration with 250 denoising steps and churn parameter of $\varepsilon = 0.25$. We use 32 DDPM sampling trajectories. The one-step lagged predictions are then compared to the true model predictions to determine the Mahalanobis distance and acceptance probability.

temperature term is to increase the acceptance rate of the draft sequence, while gracefully degrading quality (via a deviation from the primary model q). One of the original proposals of speculative decoding (Leviathan et al., 2023) suggested a form of lenience by replacing the density ratio "q/p" in verification with a scaled ratio "lq/p", where an increase in acceptance occurs with $l > 1$. Verification of this form can be shown to be part of an optimal relaxation (in terms of matching q) when the target distribution is slightly changed (Tran-Thien, 2023). It should be noted that this optimal form of lossy speculative decoding could not be previously utilized as the residual deviates from what the reflection coupling calculates (Algo. 4). However, the decomposition residual (Algo. 5) naturally allows for the required target distribution to implement lenience. Recently, other approaches have also looked into replacing the verification procedure in speculative decoding to increase acceptance rate (Zhong et al., 2025; Bachmann et al., 2025).

In the following section, we study the unique temperature lenience available in speculative diffusion, as introduced in De Bortoli et al. (2025). Instead of linearly scaling the density ratio "q/p" we directly alter p, q by changing their identity covariance matrices to a scaled version $\mathrm{Id} \mapsto \omega^2 \mathrm{Id}$. For sample verification, we consider a version of De Bortoli et al. (2025, Proposition F.1) specific to the decomposition residual with sample verification, as shown in Algo. 9. We also discuss and present a variant of block verification with temperature, as shown in Algo. 10.

We note that when we consider $\omega \neq 1$, the output distribution of speculative diffusion (of any kind) will not sample from the original target diffusion—the exactness guarantees in the main text only apply to the untempered $\omega = 1$ case.

I.1 Sample Decomposition

In the following, we explore using a temperature term in the verification step of sample-decomposition speculative diffusions. We recall that the output distribution of this setting corresponds to

$$\pi_{\text{sample}}(y) = p(y)\alpha(y) + r(y)\left(1 - \int p(\bar{y})\alpha(\bar{y})\,\mathrm{d}\bar{y}\right), \tag{46}$$

Algorithm 9:
`SampleVerificationWithTemperature`$((\hat{y}_{k+i}, m^p_{k+i-1}, m^q_{k+i-1}, \sigma_{k+i-1})_{i=1}^{\gamma}; \omega)$

Require: Draft $\hat{y}_{k+1:k+\gamma}$, draft means $m^p_{k:k+\gamma-1}$, target means $m^q_{k:k+\gamma-1}$, variances $\sigma^2_{k:k+\gamma-1}$, temperature $\omega > 0$.

1: Set $\boldsymbol{c}_0 = 1$.
2: **for** $j \in \{1, \dots, \gamma\}$ **do**
3: Set $\Delta_j = (m^p_{k+j-1} - m^q_{k+j-1})/\sigma_{k+j-1}$ and $\mathsf{Z}_j = (\hat{y}_{k+j} - m^p_{k+j-1})/\sigma_{k+j-1}$.
4: Calculate $\alpha_j = \min\left\{1, \frac{\mathcal{N}(\mathsf{Z}_j+\Delta_j;0,\omega^2\mathrm{Id})}{\mathcal{N}(\mathsf{Z}_j;0,\omega^2\mathrm{Id})}\right\}$.
5: Flip coin $\boldsymbol{c}_j = [\![\eta < \alpha_j]\!]$, where $\eta \sim \mathrm{Unif}[0,1]$.
6: **end for**
7: Set $\tau = \min(\{j \mid \boldsymbol{c}_j = 0\} \cup \{\gamma + 1\})$.
8: **return** τ, $\Delta = \Delta_\tau$, $\mathsf{Z} = \mathsf{Z}_\tau$, $m^q = m^q_{k+\tau-1}$, $\sigma = \sigma_{k+\tau-1}$, $\alpha_{\text{res}} = 1$.[1]

where

$$
\begin{aligned}
p(y) &= \mathcal{N}(y; m^p, \sigma^2 \mathrm{Id}) \\
\alpha(y) &= \min\left\{1, \frac{\mathcal{N}(y; m^q, \sigma^2 \mathrm{Id})}{\mathcal{N}(y; m^p, \sigma^2 \mathrm{Id})}\right\} \\
r(y) &\propto \max\{0, \mathcal{N}(y; m^q, \sigma^2 \mathrm{Id}) - \mathcal{N}(y; m^p, \sigma^2 \mathrm{Id})\}.
\end{aligned}
$$

Here no temperature has been added.

To add temperature, we modify the acceptance coin c via

$$
\alpha^\omega(y) = \min\left\{1, \frac{\mathcal{N}(y; m^q, \omega^2\sigma^2 \mathrm{Id})}{\mathcal{N}(y; m^p, \omega^2\sigma^2 \mathrm{Id})}\right\}.
$$

One of course has $\alpha^{\omega=1} = \alpha$.

Now we can calculate

$$
\begin{aligned}
\pi^\omega_{\text{sample}}(y) = \mathcal{N}(y; m^p, \sigma^2 \mathrm{Id}) \min\left\{1, \frac{\mathcal{N}(y; m^q, \omega^2\sigma^2 \mathrm{Id})}{\mathcal{N}(y; m^p, \omega^2\sigma^2 \mathrm{Id})}\right\} \\
+ \mathrm{norm}(\max\{0, \mathcal{N}(y; m^q, \sigma^2 \mathrm{Id}) - \mathcal{N}(y; m^p, \sigma^2 \mathrm{Id})\})\,(1 - P_{\text{accept}}(\omega))\,,
\end{aligned}
$$

where $\mathrm{norm}(f(x)) \doteq f(x)/\int f(x)\,\mathrm{d}x$ unit normalizes a function and

$$
P_{\text{accept}}(\omega) = \int p(\bar{y})\alpha^\omega(\bar{y})\,\mathrm{d}\bar{y}. \tag{47}
$$

Through a change of variables $y = m^q + \sigma z$, we have that the expected value of a function f is given by,

$$
\begin{aligned}
&\mathbb{E}_{\pi^\omega_{\text{sample}}}[f(\mathsf{Y}^\omega)] \\
&= \int f(m^q + \sigma z)\mathcal{N}(z - \Delta; 0, \mathrm{Id}) \min\left\{1, \frac{\mathcal{N}(z; 0, \omega^2\mathrm{Id})}{\mathcal{N}(z - \Delta; 0, \omega^2\mathrm{Id})}\right\} \\
&\qquad + f(m^q + \sigma z)(1 - P_{\text{accept}}(\omega))\mathrm{norm}(\max\{0, \mathcal{N}(z; 0, \mathrm{Id}) - \mathcal{N}(z - \Delta; 0, \mathrm{Id})\})\,\mathrm{d}z \\
&= \int f(m^q + \sigma z)\pi^\omega_{\text{sample}}(z)\,\mathrm{d}z,
\end{aligned}
$$

where with abuse of notation we define $\pi^\omega_{\text{sample}}(z)$ (and each corresponding sub-function) as a function of z. One can check that $\pi^\omega_{\text{sample}}(z)$ is a valid distribution over noise values z and that $P_{\text{accept}}(\omega)$ can also be rewritten in terms of z.

As a result we have the following result.

Proposition I.1. *Let Y^ω be the output of speculative diffusion Algo. 1 with sample verification with temperature $\omega > 0$ Algo. 9 and decomposition residual (Algo. 5, $\alpha = 1$). Then we have that*

$$
\mathbb{E}_{\pi^\omega_{\text{sample}}}[\mathsf{Y}^\omega] = m^q + \sigma eg(\|\Delta\|, \omega), \tag{48}
$$

where $e = \Delta/\|\Delta\|$ *and* $\Delta = (m^p - m^q)/\sigma$.

Furthermore, we have $g(\|\Delta\|, \omega) = 0$ *for* $\omega = 1$ *and is an increasing function of* ω*. Additionally for* $\omega > 1$*,* $g(\|\Delta\|, \omega) \geq 0$*; otherwise when* $\omega < 1$*,* $g(\|\Delta\|, \omega) \leq 0$*.*

Proof. One immediately has that

$$\mathbb{E}_{\pi^{\omega}_{\text{sample}}}[\mathsf{Y}^{\omega}] = m^q + \sigma \int z \pi^{\omega}_{\text{sample}}(z)\, \mathrm{d}z,$$

which gives us the functional form of utilizing a temperature in verification. What remains is the explicit analysis of the latter term.

To do so, we will consider a decomposition that involves the average tempered acceptance. Let $P_{\text{accept}}(\omega)$ denote the average (w.r.t. the drafter distribution) rate of acceptance, *i.e.*,

$$P_{\text{accept}}(\omega) = \int p(z) \alpha^{\omega}(z)\, \mathrm{d}z. \tag{49}$$

Let

$$E(\omega) = \mathbb{E}_{\pi^{\omega}_{\text{sample}}(z)}[\mathsf{Z}] = \int z \pi^{\omega}_{\text{sample}}(z)\, \mathrm{d}z,$$

where we note that $P_{\text{accept}}(\omega)$ normalizes $p(z)\alpha^{\omega}(z)$ to be a probability measure.

As $\omega = 1$ corresponds to the untempered case, we know that $E(1) = 0$. Then, from the fundamental theorem of calculus, we have that

$$E(\omega) = E(\omega) - E(1) = \int_1^{\omega} \frac{\mathrm{d}E(t)}{\mathrm{d}t}\, \mathrm{d}t,$$

Noting that our terms are continuous functions and (at least) piece-wise differentiable. We will examine the sign of the derivative to determine the sign of $E(\omega)$. Notice that

$$\frac{\mathrm{d}E(\omega)}{\mathrm{d}\omega} = \int z p(z) \left(\frac{\partial}{\partial \omega} \alpha^{\omega}(z) \right) \mathrm{d}z - \mu_{\text{res}} \frac{\partial P_{\text{accept}}(\omega)}{\partial \omega}.$$

First, we give a simplification of $\alpha^{\omega}(z)$.

$$\begin{aligned} \alpha^{\omega}(z) &= \min \left\{ 1, \frac{\mathcal{N}(z; 0, \omega^2 \mathrm{Id})}{\mathcal{N}(z - \Delta; 0, \omega^2 \mathrm{Id})} \right\} \\ &= \min \left\{ 1, \exp\left(\frac{1}{2\omega^2} \left(\|\Delta\|^2 - 2z^{\top}\Delta \right) \right) \right\}. \end{aligned}$$

Furthermore, the derivative of $\alpha^{\omega}(z)$ is non-zero only for values $2z^{\top}\Delta > \|\Delta\|^2$, *i.e.*, when $\alpha^{\omega}(z) < 1$. In this case we have,

$$\begin{aligned} \frac{\partial \alpha^{\omega}(z)}{\partial \omega} &= \frac{\partial}{\partial \omega} \left(\frac{\mathcal{N}(z; 0, \omega^2 \mathrm{Id})}{\mathcal{N}(z - \Delta; 0, \omega^2 \mathrm{Id})} \right) \\ &= \frac{\partial}{\partial \omega} \left(\exp\left(\frac{1}{2\omega^2} \left(\|\Delta\|^2 - 2z^{\top}\Delta \right) \right) \right) \\ &= \exp\left(\frac{1}{2\omega^2} \left(\|\Delta\|^2 - 2z^{\top}\Delta \right) \right) \frac{2z^{\top}\Delta - \|\Delta\|^2}{\omega^3}. \end{aligned}$$

Thus, recombining to give our integral, we have

$$\begin{aligned} &\int z p(z) \left(\frac{\partial}{\partial \omega} \alpha^{\omega}(z) \right) \mathrm{d}z \\ &\quad = \int_{z:\ 2z^{\top}\Delta > \|\Delta\|^2} z \mathcal{N}(z - \Delta; 0, \mathrm{Id}) \exp\left(\frac{1}{2\omega^2} \left(\|\Delta\|^2 - 2z^{\top}\Delta \right) \right) \frac{2z^{\top}\Delta - \|\Delta\|^2}{\omega^3}\, \mathrm{d}z. \end{aligned}$$

To evaluate this, we consider an orthogonal decomposition of $z = (e^\top z)e + z_\perp$, where we remind that $e = \Delta/\|\Delta\|$. We will use a change of coordinates onto a basis corresponding to this decomposition (*i.e.*, a basis where the first coordinate is along e). In particular, we utilize the reflection matrix $\tilde{P}_{\text{r}} = \text{Id} - 2vv^\top/(v^\top v)$ where $v = \xi_1 - e$ and $\xi_1 = (1, 0, \ldots, 0)$ is the first standard basis vector. Here we note that $\tilde{P}_{\text{r}}\xi_1 = e$ and

$$\det(\tilde{P}_{\text{r}})^2 = \det(\tilde{P}_{\text{r}}\tilde{P}_{\text{r}}) = \det(\text{Id}) = 1 \quad \implies \quad |\det(\tilde{P}_{\text{r}})| = 1.$$

Thus through a change of coordinates via $\tilde{P}_{\text{r}}$ we have

$$\begin{aligned}
&\frac{1}{\|\Delta\|}\int zp(z)\left(\frac{\partial}{\partial\omega}\alpha^\omega(z)\right)\mathrm{d}z \\
&= \int\int_{u:\ 2u>\|\Delta\|}(ue+z_\perp)\mathcal{N}(ue+z_\perp;\Delta,\text{Id})\exp\left(\frac{\|\Delta\|}{2\omega^2}(\|\Delta\|-2u)\right)\frac{2u-\|\Delta\|}{\omega^3}\,\mathrm{d}u\,\mathrm{d}z_\perp \\
&= \int\int_{\|\Delta\|/2}^{\infty}(ue+z_\perp)\mathcal{N}(ue+z_\perp;\Delta,\text{Id})\exp\left(\frac{\|\Delta\|}{2\omega^2}(\|\Delta\|-2u)\right)\frac{2u-\|\Delta\|}{\omega^3}\,\mathrm{d}u\,\mathrm{d}z_\perp \\
&= \int\int_{\|\Delta\|/2}^{\infty}(ue+z_\perp)\mathcal{N}(ue+z_\perp-\|\Delta\|e;0,\text{Id})\exp\left(\frac{\|\Delta\|}{2\omega^2}(\|\Delta\|-2u)\right)\frac{2u-\|\Delta\|}{\omega^3}\,\mathrm{d}u\,\mathrm{d}z_\perp \\
&= \int\int_{\|\Delta\|/2}^{\infty}(ue+z_\perp)\mathcal{N}((u-\|\Delta\|)e+z_\perp;0,\text{Id})\exp\left(\frac{\|\Delta\|}{2\omega^2}(\|\Delta\|-2u)\right)\frac{2u-\|\Delta\|}{\omega^3}\,\mathrm{d}u\,\mathrm{d}z_\perp \\
&= \int\int_{\|\Delta\|/2}^{\infty}(ue+z_\perp)\mathcal{N}(u-\|\Delta\|;0,1)\mathcal{N}(z_\perp;0,\text{Id})\exp\left(\frac{\|\Delta\|}{2\omega^2}(\|\Delta\|-2u)\right)\frac{2u-\|\Delta\|}{\omega^3}\,\mathrm{d}u\,\mathrm{d}z_\perp,
\end{aligned}$$

where we are using the fact that $e^\top z_\perp = 0$ by construction. Now noting that the expectation of $\mathcal{N}(z_\perp; 0, \text{Id})$ is 0 (and that it normalizes to 1), we have

$$\begin{aligned}
&\int zp(z)\left(\frac{\partial}{\partial\omega}\alpha^\omega(z)\right)\mathrm{d}z \\
&= e\|\Delta\|\int_{\|\Delta\|/2}^{\infty}u\mathcal{N}(u-\|\Delta\|;0,1)\exp\left(\frac{\|\Delta\|}{2\omega^2}(\|\Delta\|-2u)\right)\frac{2u-\|\Delta\|}{\omega^3}\,\mathrm{d}u \\
&\doteq eA(\omega).
\end{aligned}$$

Notice, the integral term $A(\omega)$ is non-negative due to the bounds of integration. Thus the entire term's sign is dictated by $e = \Delta/\|\Delta\|$.

Similarly, we can simplify $\partial_\omega P_{\text{accept}}(\omega)$ via

$$\begin{aligned}
&\frac{\partial P_{\text{accept}}(\omega)}{\partial\omega} \\
&= \int p(z)\left(\frac{\partial}{\partial\omega}\alpha^\omega(z)\right)\mathrm{d}z \\
&= \|\Delta\|\int\int_{\|\Delta\|/2}^{\infty}\mathcal{N}(u-\|\Delta\|;0,1)\mathcal{N}(z_\perp;0,\text{Id})\exp\left(\frac{\|\Delta\|}{2\omega^2}(\|\Delta\|-2u)\right)\frac{2u-\|\Delta\|}{\omega^3}\,\mathrm{d}u\,\mathrm{d}z_\perp \\
&= \|\Delta\|\int_{\|\Delta\|/2}^{\infty}\mathcal{N}(u-\|\Delta\|;0,1)\exp\left(\frac{\|\Delta\|}{2\omega^2}(\|\Delta\|-2u)\right)\frac{2u-\|\Delta\|}{\omega^3}\,\mathrm{d}u \\
&\doteq B(\omega).
\end{aligned}$$

Which again $B(\omega)$ is positive.

Finally, we consider μ_{res}. We consider the integral of the un-normalized term:

$$\int z\max\{0, \mathcal{N}(z;0,\text{Id}) - \mathcal{N}(z-\Delta;0,\text{Id})\}\,\mathrm{d}z.$$

The inner term is only positive when

$$\begin{aligned}\mathcal{N}(z;0,\mathrm{Id})-\mathcal{N}(z-\Delta;0,\mathrm{Id})>0 \iff & \frac{\mathcal{N}(z;0,\mathrm{Id})}{\mathcal{N}(z-\Delta;0,\mathrm{Id})}>1\\ \iff & \exp\left(\frac{1}{2\omega^2}\left(\|\Delta\|^2-2z^\top\Delta\right)\right)>1\\ \iff & \frac{1}{2\omega^2}\left(\|\Delta\|^2-2z^\top\Delta\right)>0\\ \iff & 2z^\top e<\|\Delta\|.\end{aligned}$$

Notice, that the condition on $z^\top e$ is flipped this time. Now finally, we do the same change of coordinates as above to yield

$$\begin{aligned}&\int z\max\{0,\mathcal{N}(z;0,\mathrm{Id})-\mathcal{N}(z-\Delta;0,\mathrm{Id})\}\,\mathrm{d}z\\ &=\int\int_{-\infty}^{\|\Delta\|/2}(ue+z_\perp)\left(\mathcal{N}(ue+z_\perp;0,\mathrm{Id})-\mathcal{N}((u-\|\Delta\|)e+z_\perp;0,\mathrm{Id})\right)\,\mathrm{d}u\,\mathrm{d}z_\perp\\ &=\int\int_{-\infty}^{\|\Delta\|/2}(ue+z_\perp)\mathcal{N}(z_\perp;0,\mathrm{Id})\left(\mathcal{N}(u;0,1)-\mathcal{N}(u-\|\Delta\|;0,1)\right)\,\mathrm{d}u\,\mathrm{d}z_\perp\\ &=e\int_{-\infty}^{\|\Delta\|/2}u\mathcal{N}(u;0,1)-u\mathcal{N}(u-\|\Delta\|;0,1)\,\mathrm{d}u\\ &=e\int_{-\infty}^{\|\Delta\|/2}u\mathcal{N}(u;0,1)\,\mathrm{d}u-e\int_{-\infty}^{-\|\Delta\|/2}(v+\|\Delta\|)\mathcal{N}(v;0,1)\,\mathrm{d}v\\ &=e\int_{-\infty}^{\|\Delta\|/2}u\mathcal{N}(u;0,1)\,\mathrm{d}u-e\int_{-\infty}^{-\|\Delta\|/2}v\mathcal{N}(v;0,1)\,\mathrm{d}v-e\|\Delta\|\int_{-\infty}^{-\|\Delta\|/2}\mathcal{N}(v;0,1)\,\mathrm{d}v\\ &=e\left(\left[-\mathcal{N}(u;0,1)\Big|_{u=-\infty}^{u=\|\Delta\|/2}\right]-\left[-\mathcal{N}(u;0,1)\Big|_{u=-\infty}^{u=-\|\Delta\|/2}\right]-\|\Delta\|\Phi\left(-\frac{\|\Delta\|}{2}\right)\right)\\ &=e\left(\mathcal{N}\left(-\frac{\|\Delta\|}{2};0,1\right)-\mathcal{N}\left(\frac{\|\Delta\|}{2};0,1\right)-\|\Delta\|\Phi\left(-\frac{\|\Delta\|}{2}\right)\right)\\ &=-e\|\Delta\|\Phi\left(-\frac{\|\Delta\|}{2}\right),\end{aligned}$$

where we are using the symmetry of the normal distribution p.d.f.

Thus together, we get

$$\begin{aligned}\mu_{\text{res}}&=-e\|\Delta\|\frac{\Phi\left(-\frac{\|\Delta\|}{2}\right)}{\int_{-\infty}^{\|\Delta\|/2}\left(\mathcal{N}(u;0,1)-\mathcal{N}(u-\|\Delta\|;0,1)\right)\,\mathrm{d}u}\\ &=-e\|\Delta\|\frac{\Phi\left(-\frac{\|\Delta\|}{2}\right)}{\Phi\left(\frac{\|\Delta\|}{2}\right)-\Phi\left(-\frac{\|\Delta\|}{2}\right)}\\ &=-e\|\Delta\|\frac{1-\Phi\left(\frac{\|\Delta\|}{2}\right)}{2\Phi\left(\frac{\|\Delta\|}{2}\right)-1}\\ &\doteq -eC.\end{aligned}$$

where the fractional term C is non-negative.

Hence we have that

$$\frac{\mathrm{d}E(\omega)}{\mathrm{d}\omega}=e\left(A(\omega)+B(\omega)C\right). \tag{50}$$

And thus

$$E(\omega) = e \int_1^{\omega} A(t) + B(t)C \, \mathrm{d}t, \tag{51}$$

with $A(t)+B(t)C \geq 0$ for all $t \geq 0$. Thus with $\omega > 1$, $E(\omega)$ increases in magnitude while following the sign of e. When $\omega < 1$, $E(\omega)$ increases in magnitude in the opposite direction/sign of e. □

I.2 BLOCK

Adding temperature to block verification is not as straightforward as its sample verification counterpart. There are multiple ways of doing this, where the differences come from determining where the influence of the temperature ω starts and stops, *i.e.*, should α_i, which is passed to Algo. 5, be computed with temperature. For simplicity, we propose and experimentally test a variant of temperature using the verification of Algo. 10. In addition, when calculating the decomposition residual, we utilize the temperature scaled α_i's. Again, we note that for $\omega \neq 1$, Algo. 10 is lossy and does not guarantee that speculative diffusion recovers the target diffusion.

Algorithm 10:
`BlockVerificationWithTemperature`$((\hat{y}_{k+i}, m^p_{k+i-1}, m^q_{k+i-1}, \sigma_{k+i-1})_{i=1}^{\gamma}; \omega)$

Require: Draft $\hat{y}_{k+1:k+\gamma}$, draft means $m^p_{k:k+\gamma-1}$, target means $m^q_{k:k+\gamma-1}$, variances $\sigma^2_{k:k+\gamma-1}$, temperature $\omega > 0$.

Initialize $\boldsymbol{c}_0 = 1$ and $\alpha_0 = 1$.
for $j \in \{1, \ldots, \gamma\}$ **do**
  Set $\Delta_j = (m^p_{k+j-1} - m^q_{k+j-1})/\sigma_{k+j-1}$.
  Set $\mathsf{Z}_j = (\hat{y}_{k+j} - m^p_{k+j-1})/\sigma_{k+j-1}$.
  Calculate $\alpha_j = \min\left\{1, \alpha_{j-1} \frac{\mathcal{N}(\mathsf{Z}_j+\Delta_j; 0, \omega^2 \mathrm{Id})}{\mathcal{N}(\mathsf{Z}_j; 0, \omega^2 \mathrm{Id})}\right\}$.
end for
for $j \in \{1, \ldots, \gamma\}$ **do**
  if $j \neq \gamma$ **then**
    Calculate $h_j = v_j/(v_j + 1 - \alpha_j)$, where

$$v_j = \alpha_j \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} + \frac{\|\Delta_{j+1}\|}{2}\right) - \Phi\left(\frac{\log \alpha_j}{\|\Delta_{j+1}\|} - \frac{\|\Delta_{j+1}\|}{2}\right). \tag{52}$$

  else
    Set $h_\gamma = \alpha_\gamma$
  end if
  Flip coin $\boldsymbol{c}_j = [\![\eta < h_j]\!]$, where $\eta \sim \mathrm{Unif}[0, 1]$.
end for
Set $\tau = 1 + \max\{j \in \{0, 1, \ldots, \gamma\} \mid \boldsymbol{c}_j = 1\}$.
return τ, $\Delta = \Delta_\tau$, $\mathsf{Z} = \mathsf{Z}_\tau$, $m^q = m^q_{k+\tau-1}$, $\sigma = \sigma_{k+\tau-1}$, $\alpha_{\mathrm{res}} = \alpha_{\tau-1}$.[1]

J EXPERIMENTAL DETAILS

Pixel Experiments. We follow the experimental setup of De Bortoli et al. (2025). For all pixel experiments, we consider a U-Net architecture (Ronneberger et al., 2015). For each U-Net, the channel size is 192 for ImageNet and 256 for the other datasets. We also apply multipliers for the channels at each level, where the multipliers are $(1, 2, 3, 4)$ for ImageNet and $(1, 2, 2, 2)$ for other datasets. At each level, we consider 3 residual blocks for ImageNet and 2 residual blocks for other datasets. An attention layer is applied on the second level of the U-Net and attention is also used at the bottleneck block. Each residual block consists of the following: the input is normalized, a 3×3 convolution block is applied, non-linear activation is applied, a dropout layer, and then a final 3×3 convolution block is applied. A residual connection is also utilized to add the input via a 1×1 convolutional block. RMS normalization and GELU are used for the normalization layer and non-linearity, respectively. When applicable, multi-head attention with 8 heads is used after the convolutional residual block. For a time embedding, we use a 192-dimensional sinusoidal embedding

for ImageNet and 256-dimensional sinusoidal embedding for other datasets. When the dataset has labels, we embed the different classes and consider a conditional model, where we also condition on the augmentation vector. For training, we use the Adam optimizer with additional norm clipping (equal to 1) with learning rate $1e-5$.

Latent Experiments. Our latent models follow the recipe of Rombach et al. (2022). We train an autoencoder to encode images of shape $256 \times 256 \times 3$ to latent tensors of shape $64 \times 64 \times 3$. The autoencoder architectures utilized mirror LDM-4 for their respective datasets, where we use β-VAE with $\beta = 1e-6$ for both datasets. For CelebA, we utilize a 10.0 coefficient on adversarial loss and 10.0 coefficient on generator loss. For ImageNet, we utilize a 0.5 coefficient on adversarial loss and 1.0 coefficient on generator loss. For the actual latent diffusion model, we also mirror LDM-4 in (Rombach et al., 2022). For CelebA, we use a channel size of 224 with $(1, 2, 3, 4)$ multipliers. We utilize a dropout rate of 0.2 and 2 residual blocks. A learning rate of $9.6e-5$ is used with Adam (with norm clipping equal to 1). For ImageNet, we use a channel size of 256 with $(1, 2, 3, 5)$ multipliers. We use a dropout of 0.1 and 2 residual blocks. A learning rate of $1e-4$ is used with the same optimizer.

Sampling. Both in sampling and training, we use the rectified flow noise schedule (Liu et al., 2022b). We utilize the sampling method in Ho et al. (2020) with a safety epsilon of $1e-4$. Speculative diffusions are then used to speed up this base sampling method. In Algo. 7, the two while-loops suggest that we repeat the loop until a termination criterion is met. In practice, we find that fixing the number of iterations does not cause a major degradation in performance. For ease of implementation in `JAX`, we establish the lower bound u_{lower} by a maximum of 3 iterations and run the binary search with a maximum of 4 iterations.

Error Propagation. In some of our result tables, we give an estimate of the std. for the speedup w.r.t. the DDPM wall-clock time. Although we have the std. of the wall-clock time of the speculative diffusion sampling and the DDPM sampling, we do not have the std. of the ratio. We propagate the error via a first-order Taylor series approximation, see for instance Lee and Forthofer (2005, Chapter 4).

K Additional Experiments

Additional FID scores We present additional results to verify speedups and FID scores. Tables 5 and 6 present these results over all datasets for churn parameters $\varepsilon = 0.25$ and $\varepsilon = 0.5$. This is fixed for a draft sequence size of $\gamma = 7$. FID scores for all datasets are calculated over 50k samples, except for CelebA LDM which utilizes 30k samples.

We note that the original sampling FID and speculative diffusion FID are similar across these results. The FID scores of the decomposition (D) and block (B) approaches tend to be more similar in FID scores, which is expected as the residual distributions they target are similar.

Temperature In the following section, we consider a simple experiment utilizing the temperature lenience schemes described in Appendix I. For a fixed churn of $\varepsilon = 0.25$ and draft size $\gamma = 7$, Table 7 presents the wall-clock times of sampling and Table 8 presents the FID score. An increase in ω corresponds to an increase in acceptance rate. $\omega = 1$ corresponds to the case without temperature. Notice that w.r.t. wall-clock time, an increase in ω (usually) decreases the wall-clock time. This is consistent for denoising steps $50, 100, 250$. However, for larger denoising steps $500, 1000$, there is an unexpected decrease in the wall-clock time for $\omega = 0.5$. This is unexpected as $\omega < 1$ should decrease acceptance rate, *i.e.*, the verification value α_i will be smaller. A possible reason for this is that rejecting the drafter more at the start of the denoising process can allow for higher acceptance later on.

In contrast, the FID scores generally worsen as ω increases, consistent with accepting more draft proposals that would be rejected by the exact sampler. Despite this, the difference can be minimal when the number of denoising steps increases. A reason for this behavior is that as the number of denoising steps increases, the number of steps at the end of the denoising also increases. But even for a large $\omega > 1$, we will almost always be rejecting all draft sequences at the end of speculative diffusion. Thus, the final sampling from the primary model may allow for some "correction" of the increased initial draft sequence acceptance. Nevertheless, in this specific experiment, it is clear that if one is utilizing a large number of denoising steps ($K = 1000$), a higher temperature ($\omega = 2.0$) can

Table 5: Wall-clock speedups and FID scores over all datasets and different number of denoising steps. The churn parameter is set to $\varepsilon = 0.25$ and the window size is set to $\gamma = 7$. Non-FID quantities are calculated over 500 samples and the $\pm$ error ranges are approximated via error propagation.

		Wall-clock Speedup				FID			
Dataset	Steps	R	D	B	R↑B%	R	D	B	DDPM
CelebA LDM	50	1.10 ± 0.06	1.10 ± 0.06	1.10 ± 0.06	-0.59%	6.17	6.33	6.36	6.38
	100	1.48 ± 0.07	1.47 ± 0.07	1.48 ± 0.08	0.54%	6.19	6.13	6.18	6.23
	250	2.20 ± 0.09	2.19 ± 0.09	2.25 ± 0.10	2.10%	6.42	6.11	6.13	6.23
	500	2.94 ± 0.11	2.95 ± 0.11	3.04 ± 0.11	3.47%	6.43	6.09	6.08	6.27
	1000	3.83 ± 0.11	3.83 ± 0.10	3.95 ± 0.11	3.27%	6.27	6.12	6.11	6.22
CelebA Pixel	50	1.41 ± 0.11	1.40 ± 0.12	1.42 ± 0.13	0.67%	6.24	6.28	6.20	6.24
	100	2.02 ± 0.16	2.03 ± 0.17	2.11 ± 0.18	4.52%	3.45	3.46	3.41	3.44
	250	3.20 ± 0.20	3.18 ± 0.21	3.35 ± 0.21	4.81%	2.77	2.61	2.57	2.63
	500	4.16 ± 0.19	4.16 ± 0.19	4.32 ± 0.21	3.79%	2.81	2.62	2.54	2.63
	1000	5.08 ± 0.17	5.09 ± 0.16	5.22 ± 0.16	2.75%	2.73	2.62	2.60	2.68
CIFAR10 Pixel	50	1.52 ± 0.19	1.51 ± 0.19	1.51 ± 0.21	-0.90%	3.76	3.77	3.80	3.85
	100	2.33 ± 0.24	2.34 ± 0.25	2.35 ± 0.24	1.10%	2.76	2.75	2.78	2.85
	250	3.61 ± 0.25	3.59 ± 0.24	3.60 ± 0.24	-0.04%	2.13	2.15	2.14	2.23
	500	4.49 ± 0.21	4.49 ± 0.20	4.45 ± 0.20	-0.90%	2.17	2.21	2.22	2.29
	1000	5.22 ± 0.15	5.23 ± 0.15	5.12 ± 0.14	-2.00%	2.23	2.26	2.25	2.37
ImageNet LDM	50	1.18 ± 0.06	1.18 ± 0.07	1.18 ± 0.06	0.35%	11.31	10.97	11.01	11.22
	100	1.53 ± 0.08	1.53 ± 0.08	1.55 ± 0.07	0.98%	10.70	10.13	10.14	10.77
	250	2.24 ± 0.10	2.23 ± 0.10	2.28 ± 0.11	1.76%	10.81	9.64	9.70	10.52
	500	2.99 ± 0.12	2.99 ± 0.13	3.10 ± 0.14	3.80%	10.83	9.58	9.66	10.16
	1000	3.88 ± 0.12	3.90 ± 0.12	4.05 ± 0.13	4.20%	10.59	9.60	9.59	10.18
ImageNet Pixel	50	1.24 ± 0.09	1.24 ± 0.10	1.25 ± 0.11	1.20%	3.47	3.42	3.44	3.50
	100	1.77 ± 0.16	1.77 ± 0.16	1.83 ± 0.18	3.47%	2.93	2.97	2.93	2.97
	250	2.82 ± 0.23	2.81 ± 0.22	2.96 ± 0.22	4.70%	2.60	2.59	2.59	2.66
	500	3.72 ± 0.24	3.71 ± 0.23	3.86 ± 0.23	3.75%	2.52	2.49	2.49	2.57
	1000	4.58 ± 0.20	4.60 ± 0.21	4.76 ± 0.19	3.92%	2.47	2.45	2.45	2.55
LSUN Pixel	50	1.18 ± 0.10	1.17 ± 0.10	1.17 ± 0.10	-0.43%	6.49	6.56	6.58	6.46
	100	1.71 ± 0.14	1.70 ± 0.14	1.75 ± 0.16	2.42%	5.10	5.15	5.10	5.08
	250	2.74 ± 0.19	2.74 ± 0.17	2.86 ± 0.19	4.44%	4.11	4.05	4.05	4.03
	500	3.64 ± 0.18	3.63 ± 0.18	3.75 ± 0.18	3.03%	3.76	3.64	3.64	3.70
	1000	4.48 ± 0.16	4.49 ± 0.15	4.59 ± 0.15	2.53%	3.67	3.50	3.51	3.55

achieve similar FID scores while also being almost twice as fast as its no temperature counterpart ($\omega = 1.0$).

Table 6: Wall-clock speedups and FID scores over all datasets and different number of denoising steps. The churn parameter is set to $\varepsilon = 0.5$ and the window size is set to $\gamma = 7$. Non-FID quantities are calculated over 500 samples, and the $\pm$ error ranges are approximated via error propagation.

		Wall-clock Speedup				FID			
Dataset	Steps	R	D	B	R↑B%	R	D	B	DDPM
CelebA LDM	50	1.13 ± 0.06	1.13 ± 0.07	1.13 ± 0.07	-0.39%	6.34	6.53	6.62	6.64
	100	1.47 ± 0.07	1.47 ± 0.08	1.49 ± 0.07	1.31%	6.31	6.20	6.31	6.55
	250	2.15 ± 0.09	2.14 ± 0.09	2.20 ± 0.10	2.43%	6.43	6.22	6.27	6.59
	500	2.84 ± 0.10	2.85 ± 0.11	2.94 ± 0.11	3.43%	6.46	6.34	6.32	6.73
	1000	3.70 ± 0.11	3.72 ± 0.10	3.82 ± 0.11	3.07%	6.58	6.62	6.62	6.75
CelebA Pixel	50	1.44 ± 0.12	1.43 ± 0.12	1.47 ± 0.13	1.60%	7.02	7.18	7.02	6.99
	100	2.04 ± 0.17	2.04 ± 0.16	2.14 ± 0.18	4.95%	3.96	3.90	3.84	3.91
	250	3.15 ± 0.21	3.14 ± 0.20	3.30 ± 0.20	4.83%	2.91	2.77	2.74	2.86
	500	4.09 ± 0.20	4.09 ± 0.19	4.24 ± 0.22	3.90%	2.93	2.62	2.65	2.83
	1000	5.03 ± 0.19	5.04 ± 0.17	5.16 ± 0.17	2.58%	2.87	2.65	2.67	2.94
CIFAR10 Pixel	50	1.65 ± 0.21	1.64 ± 0.21	1.67 ± 0.22	1.27%	3.91	3.92	3.91	3.95
	100	2.48 ± 0.26	2.47 ± 0.26	2.50 ± 0.26	0.97%	2.91	2.86	2.85	3.00
	250	3.65 ± 0.24	3.67 ± 0.24	3.67 ± 0.24	0.54%	2.19	2.21	2.24	2.38
	500	4.51 ± 0.21	4.52 ± 0.21	4.45 ± 0.18	-1.34%	2.22	2.21	2.23	2.52
	1000	5.23 ± 0.15	5.24 ± 0.15	5.13 ± 0.14	-1.91%	2.27	2.30	2.31	2.73
ImageNet LDM	50	1.22 ± 0.06	1.22 ± 0.06	1.22 ± 0.07	0.09%	10.83	10.77	10.82	10.78
	100	1.55 ± 0.08	1.55 ± 0.08	1.57 ± 0.08	1.47%	10.25	10.06	9.84	10.28
	250	2.20 ± 0.10	2.20 ± 0.10	2.25 ± 0.11	2.32%	10.09	9.55	9.63	9.81
	500	2.90 ± 0.12	2.91 ± 0.13	3.01 ± 0.14	3.82%	9.98	9.66	9.51	9.64
	1000	3.78 ± 0.13	3.79 ± 0.14	3.91 ± 0.14	3.67%	9.90	9.65	9.41	9.61
ImageNet Pixel	50	1.30 ± 0.11	1.29 ± 0.11	1.32 ± 0.12	1.68%	3.58	3.62	3.64	3.66
	100	1.84 ± 0.18	1.81 ± 0.17	1.89 ± 0.19	3.05%	2.97	2.99	2.99	3.04
	250	2.81 ± 0.25	2.80 ± 0.24	2.94 ± 0.24	4.66%	2.59	2.64	2.68	2.73
	500	3.68 ± 0.25	3.69 ± 0.24	3.83 ± 0.26	4.12%	2.56	2.50	2.51	2.65
	1000	4.57 ± 0.23	4.59 ± 0.23	4.75 ± 0.23	3.95%	2.51	2.50	2.50	2.61
LSUN Pixel	50	1.21 ± 0.11	1.21 ± 0.11	1.21 ± 0.11	-0.13%	7.12	7.18	7.24	7.01
	100	1.73 ± 0.14	1.73 ± 0.14	1.79 ± 0.15	3.39%	5.54	5.52	5.55	5.46
	250	2.72 ± 0.19	2.72 ± 0.18	2.83 ± 0.20	4.08%	4.28	4.26	4.35	4.46
	500	3.60 ± 0.19	3.61 ± 0.18	3.71 ± 0.19	2.87%	3.90	3.79	3.86	4.20
	1000	4.50 ± 0.17	4.51 ± 0.15	4.59 ± 0.17	2.08%	3.83	3.62	3.62	4.22

Table 7: Wall-clock (s) of speculative diffusion ($\varepsilon = 0.25$, $\gamma = 7$) for ImageNet LDM over different temperature values ω. Quantities are calculated over 500 samples; the $\pm$ ranges are the empirical std. values.

	$\omega = 0.5$			$\omega = 1.0$			$\omega = 2.0$		
Steps	R	D	B	R	D	B	R	D	B
50	1.05 ± 0.06	1.05 ± 0.06	1.01 ± 0.06	1.01 ± 0.05	1.01 ± 0.05	1.01 ± 0.05	0.85 ± 0.05	0.85 ± 0.05	0.84 ± 0.05
100	1.64 ± 0.08	1.65 ± 0.08	1.55 ± 0.08	1.56 ± 0.07	1.56 ± 0.07	1.56 ± 0.07	1.20 ± 0.06	1.21 ± 0.07	1.18 ± 0.06
250	2.94 ± 0.13	2.96 ± 0.13	2.66 ± 0.13	2.85 ± 0.10	2.84 ± 0.10	2.82 ± 0.10	1.91 ± 0.09	1.90 ± 0.09	1.85 ± 0.08
500	4.59 ± 0.17	4.59 ± 0.19	3.97 ± 0.18	4.60 ± 0.12	4.62 ± 0.13	4.55 ± 0.13	2.81 ± 0.10	2.80 ± 0.11	2.72 ± 0.10
1000	7.17 ± 0.24	7.11 ± 0.26	6.12 ± 0.22	7.79 ± 0.15	7.78 ± 0.16	7.65 ± 0.16	4.42 ± 0.11	4.41 ± 0.11	4.30 ± 0.09

Table 8: FID 50k of speculative diffusion ($\varepsilon = 0.25$, $\gamma = 7$) for ImageNet LDM over different temperature values ω.

	$\omega = 0.5$			$\omega = 1.0$			$\omega = 2.0$		
Steps	R	D	B	R	D	B	R	D	B
50	10.54	10.27	10.36	11.20	11.07	11.07	13.60	12.97	13.11
100	9.79	9.22	9.40	10.67	10.32	10.33	12.89	11.54	11.65
250	10.07	8.94	9.02	10.62	10.00	10.02	11.74	10.30	10.29
500	10.48	8.97	9.00	10.69	9.85	9.89	11.11	9.84	9.92
1000	10.39	9.23	9.17	10.53	9.88	9.84	10.64	9.71	9.78

L ACCELERATION OF MOLECULAR DYNAMICS INTEGRATORS

Molecular dynamics seeks to simulate the motion of atoms under interatomic interactions (Leimkuhler and Matthews, 2015). Evaluating the interatomic force, *e.g.*, using a machine learned interatomic potential (MLIP) (Batzner et al., 2022; Batatia et al., 2022; Wood et al., 2025; Rhodes et al., 2025), is often the dominant cost of each integration step. In the following section, we consider accelerating these updates using speculative sampling. In particular, we look to leverage block verification to provide further speedups to previously proposed Langevin speculative dynamics algorithms (De Bortoli et al., 2025; Kosmala et al., 2026).

Notation. Throughout this appendix, we will use specific notation associated with molecular dynamics. Let $\mathbf{x}, \mathbf{m} \in \mathbb{R}^{dN}$ denote the position and momenta of N atom in d dimensions. The phase-space state is denoted by $\mathbf{y} = (\mathbf{x}, \mathbf{m})$. Similar to the main text, p and q will denote the draft and target transition kernels (models), respectively. They will additionally have corresponding interatomic forces $\mathbf{F}^p$ and $\mathbf{F}^q$, *i.e.*, MLIPs. We use Y_k to denote random output states, Y'_k for intermediate states before post-processing, and $\bar{\mathsf{Y}}_k$ for staggered states which will be utilized in the OBABO integrator. We denote that two random variables have the same law by $\mathsf{X} \overset{\mathrm{d}}{=} \mathsf{Y}$.

L.1 MOLECULAR DYNAMICS AND LANGEVIN DYNAMICS

A common setting is to consider thermostatted molecular dynamics. The underdamped Langevin diffusion (ULD) couples the atoms to a heat bath (thermostat), where for the target force field $\mathbf{F}^q$ its dynamics are given by

$$\begin{aligned} \mathrm{d}\mathbf{x}_t &= \mathbf{M}^{-1}\mathbf{m}_t \,\mathrm{d}t, \\ \mathrm{d}\mathbf{m}_t &= \mathbf{F}^q(\mathbf{x}_t)\,\mathrm{d}t - \lambda \mathbf{m}_t \,\mathrm{d}t + \sqrt{2\lambda k_B T \mathbf{M}}\,\mathrm{d}\mathsf{W}_t, \end{aligned} \tag{53}$$

where W_t is a dN-dimensional Wiener process, $T > 0$ is the heat bath's temperature, $\lambda > 0$ is the friction coefficient, k_B is the Boltzmann constant, and $\mathbf{M}$ is a diagonal (positive) mass matrix. The latter 4 variables defined are fixed and define the settings of the dynamic. The first equation moves the atoms' position with the velocities $\mathbf{M}^{-1}\mathbf{m}_t$. The second equation updates the momentum based on the force $\mathbf{F}^q(\mathbf{x}_t) = -\nabla U^q(\mathbf{x}_t)$, where U^q is the interatomic potential (*e.g.*, modeled via a MILP). Additionally, the second equation accounts for energy dissipation due to friction $-\lambda\mathbf{m}_t$ and noise caused by the heat bath $\sqrt{2\lambda k_B T \mathbf{M}}\,\mathrm{d}\mathsf{W}_t$.

If the interatomic potential U^q satisfies ergodicity and weak regularity assumptions, then the dynamics converges to the Gibbs–Boltzmann density (Leimkuhler and Matthews, 2015):

$$\pi(\mathbf{x}, \mathbf{m}) \propto \exp\left(-\frac{U^q(\mathbf{x}) + \frac{1}{2}\mathbf{m}^\top \mathbf{M}^{-1}\mathbf{m}}{k_B T}\right). \tag{54}$$

The numerator of the stationary distribution (54) denotes the total energy function, also known as the Hamiltonian $H(\mathbf{x}, \mathbf{m}) \doteq U^q(\mathbf{x}) + \frac{1}{2}\mathbf{m}^\top \mathbf{M}^{-1}\mathbf{m}$.

The draft model defines a similar dynamics with the target force $\mathbf{F}^q$ replaced with the draft $\mathbf{F}^p$ (and their interatomic potentials). Similar to the speculative diffusions case of the main text, the draft force may also depend on the available history, but with slight abuse of notation, we suppress this dependence.

To integrate the dynamics of (53), a number of different simplifications and decomposition tricks are used. In De Bortoli et al. (2025, Section 5 and Appendix K), speculative sampling is applied to a simplified variant of the ULD dynamics. In particular, they consider the overdamped Langevin diffusion (OLD) limit of the ULD dynamic, where $\gamma \to \infty$. In such a regime, the Langevin dynamics is reduced to

$$\mathrm{d}\mathbf{x}_t = \mathbf{F}^q(\mathbf{x}_t)\,\mathrm{d}t + \sqrt{2\lambda k_B T}\,\mathrm{d}\mathsf{W}_t, \tag{55}$$

which no longer depends on momentum $\mathbf{m}_t$. Taking the standard Euler–Maruyama discretization yields the unadjusted Langevin algorithm (ULA) which corresponds to Gaussian draft and target update kernels with identical covariance matrices.

Instead of considering an overdamped limit, one can integrate (53) via considering a number of splitting methods. This involves decomposing the dynamics into updates corresponding to (A)

kinetic motion (drift), (B) force application (kick), and (O) an Ornstein–Uhlenbeck process. Each subproblem can be integrated (exactly) for a duration $h \geq 0$ to give the updates (Leimkuhler and Matthews, 2015, Section 7.3.1)

$$\mathrm{A}_h(\mathbf{x}, \mathbf{m}) = (\mathbf{x} + h\mathbf{M}^{-1}\mathbf{m}, \mathbf{m}), \tag{56}$$

$$\mathrm{B}_h^q(\mathbf{x}, \mathbf{m}) = (\mathbf{x}, \mathbf{m} + h\mathbf{F}^q(\mathbf{x})), \tag{57}$$

$$\mathrm{O}_h(\mathbf{x}, \mathbf{m}) = \left(\mathbf{x}, \exp(-\lambda h)\mathbf{m} + \sqrt{\mathbf{M}k_B T(1 - \exp(-2\lambda h))}\mathsf{Z}\right), \qquad \mathsf{Z} \sim \mathcal{N}(0, \mathrm{Id}), \tag{58}$$

where a draft version of the kick update B_h^p is defined analogously. Notice that the only difference between the draft and target splitting dynamics is exactly the force used in the kick update. Notationally, we will sometimes suppress the superscript q when referring to the target splitting.

Different integrators can be defined as different compositions of (56) to (58) across different steps. For instance the ABOBA and OBABO integrators corresponds to updates

$$\mathrm{ABOBA}_h = \mathrm{A}_{h/2} \circ \mathrm{B}_{h/2} \circ \mathrm{O}_h \circ \mathrm{B}_{h/2} \circ \mathrm{A}_{h/2} \tag{59}$$

$$\mathrm{OBABO}_h = \mathrm{O}_{h/2} \circ \mathrm{B}_{h/2} \circ \mathrm{A}_h \circ \mathrm{B}_{h/2} \circ \mathrm{O}_{h/2}. \tag{60}$$

Kosmala et al. (2026) introduced Langevin speculative dynamics, which allowed vanilla speculative sampling with the reflection coupling to speed up the ABOBA and OBABO integrators. Importantly, an equal covariance Gaussian update (comparing the draft and target dynamics) was identified within ABOBA and a time staggered OBABO. In what follows, we also exploit these internal Gaussian updates to provide a block verification variant of Langevin speculative dynamics. In particular, we show that pre-processing and post-processing state updates (under certain assumptions which hold for ABOBA and OBABO) preserve the target sampling target using block verification.

L.2 Block Verification Under Pre- and Post-Processing

In the following, we will analyze the effects of pre- and post-processing of state variables in dynamics when we utilize speculative sampling with block verification. Suppose that the target q and the draft p update kernels are defined on an event space $\mathcal{E}$. We further assume that these kernels can be decomposed into intermediary kernels P', Q' on $\mathcal{E}'$, which involves a common pre-processing (measurable) map $f\colon \mathcal{E} \to \mathcal{E}'$ and a common post-processing bijective (measurable) map $g\colon \mathcal{E}' \to \mathcal{E}$ with measurable inverse. In particular, we have for any $y \in \mathcal{E}$ that

$$p(\cdot \mid y) = g_{\#}P'(\cdot \mid f(y)), \qquad \text{and} \qquad q(\cdot \mid y) = g_{\#}Q'(\cdot \mid f(y)), \tag{61}$$

where $g_{\#}$ denotes the pushforward.

This structure raises the question of whether drafting, verification, and residual sampling can be performed on the intermediary kernels P' and Q'. For an intermediary state $y' \in \mathcal{E}'$, we can state the update distribution to yield the next intermediary state. Indeed, the current full state from y' corresponds to $g(y')$. Thus, the next draft and target intermediary state update distributions are given, respectively, by

$$P'(\cdot \mid f(g(y'))), \qquad \text{and} \qquad Q'(\cdot \mid f(g(y'))). \tag{62}$$

Notice, that both the pre-processing and post-process is required to be computed for each draft and target update (within a block).

To apply a speculative sampler, we map an initial full state $y_0 \in \mathcal{E}$ to its intermediate state $y_0' = g^{-1}(y_0)$, so that the first transition depends on the input $f(y_0)$. We denote $(\mathsf{Y}_k')_{k\geq 0}$ as the random intermediate output states of the speculative sampler: we run Algo. 1 with draft size $\gamma \geq 1$ on the conditional distributions in (62), using the block acceptance rule in (30), exact sampling from the residual in (9), and a target bonus upon full acceptance. For Gaussian intermediary kernels with identical p.s.d. covariance, these operations are implemented by `BlockVerification` and `Decomposition` after the change of variables described below. The full output states can then be recovered by

$$\mathsf{Y}_0' = g^{-1}(y_0), \qquad \text{and} \qquad \mathsf{Y}_k = g(\mathsf{Y}_k'), \qquad k \geq 0. \tag{63}$$

A new round of speculative sampling resumes from the previous intermediate output Y_k', noting that $f(g(\mathsf{Y}_k')) = f(\mathsf{Y}_k)$. Our goal will be to prove that Y_k in (63) matches the distribution of running the original target kernel q in (61).

In the context of sample verification, pre-processing and post-processing have also been previously studied for speculative sampling. De Bortoli et al. (2025, Appendix G) considered the maximal coupling of Gaussian draft and target states after a linear transformation, *i.e.*, projection into a latent space. More specifically, De Bortoli et al. (2025, Lemma G.1) expresses the transformed likelihood ratio and reflection correction in terms of the original state variables where the resulting maximal coupling guarantees are w.r.t. to the projected states. Kosmala et al. (2026, Theorem 3.1) show that a valid intermediate coupling yields the correct full target distribution after (common) post-processing. Invertibility of g additionally preserves the coupling maximality. This result also applies to sample verification with our proposed `Decomposition` residual ($\alpha = 1$).

The following establishes distributional correctness of block verification under these transformations.

Proposition L.1. *Under the setup of (61) to (63), assume $Q'(\cdot \mid f(y)) \ll P'(\cdot \mid f(y))$ for all $y \in \mathcal{E}$. Then the output sequence satisfies*

$$(\mathsf{Y}_0, \dots, \mathsf{Y}_n) \overset{\mathrm{d}}{=} (\mathsf{Y}_0^q, \dots, \mathsf{Y}_n^q), \qquad n \geq 0, \tag{64}$$

where $(\mathsf{Y}_k^q)_{k \geq 0}$ is the sequential target Markov chain with initial state $\mathsf{Y}_0^q = y_0$.

Proof. Fix the current full state $y \in \mathcal{E}$. For clarity, we notationally, suppress the condition of $f(y)$ on P', Q', *i.e.*, $P'(\cdot \mid f(y')) = P'(\cdot)$. Further define $P \doteq p(\cdot \mid y) = g_{\#}P'$ and $Q \doteq q(\cdot \mid y) = g_{\#}Q'$. Since g is bijective, we have

$$\frac{\mathrm{d}Q}{\mathrm{d}P}(g(y')) = \frac{\mathrm{d}Q'}{\mathrm{d}P'}(y'), \tag{65}$$

and for $\alpha \in [0, 1]$

$$g_{\#}[\alpha Q' - P']_+ = [\alpha Q - P]_+, \tag{66}$$

where $[\cdot]_+$ denotes the positive component of the measure.

The former follows from the change of variables formulation of the pushforward and the definition of the Radon–Nikodym derivatives. The latter follows from

$$\begin{aligned}
(g_{\#}[\alpha Q' - P']_+)(B) &= [\alpha Q' - P']_+(g^{-1}(B)) \\
&= \int_{g^{-1}(B)} \left[\alpha \frac{\mathrm{d}Q'}{\mathrm{d}P'} - 1\right]_+ \mathrm{d}P' \\
&= \int_B \left[\alpha \frac{\mathrm{d}Q}{\mathrm{d}P} - 1\right]_+ \mathrm{d}P = [\alpha Q - P]_+(B),
\end{aligned}$$

for any measurable subset $B \subseteq \mathcal{E}$.

As a result, applying these identities at each verification step of Algo. 6 yields the same acceptance weights α_j and coin probabilities h_j for the intermediate and full state kernels P, Q. That is, block verification does not change whether we utilize intermediary kernels P', Q' with post-processing g and pre-processed input state $f(y)$ or utilize the original kernels p, q when computing the quantities; using (65) and (66):

$$\begin{aligned}
\alpha_j &= \min\left\{1, \alpha_{j-1} \frac{\mathrm{d}Q'}{\mathrm{d}P'}(\hat{y}_{k+j-1})\right\} = \min\left\{1, \alpha_{j-1} \frac{\mathrm{d}Q}{\mathrm{d}P}(\hat{y}_{k+j-1})\right\} \\
h_j &= \frac{[\alpha_j Q' - P']_+(\mathcal{E}')}{[\alpha_j Q' - P']_+(\mathcal{E}') + 1 - \alpha_j} = \frac{[\alpha_j Q - P]_+(\mathcal{E})}{[\alpha_j Q - P]_+(\mathcal{E}) + 1 - \alpha_j},
\end{aligned}$$

where $\hat{y}_{k+j-1}$ is the draft state and h_j is given by (30). Note, we are considering the general block verification algorithm of Sun et al. (2025, Algorithm 2) which works for any pair of draft and target distributions p, q assuming one can sample from the residual; whereas we remind that Algo. 6 is specialized for speculative diffusions.

Furthermore, for the block verification residual, the pushforward identity (66) gives

$$g_{\#}\left(\frac{[\alpha Q' - P']_+}{[\alpha Q' - P']_+(\mathcal{E}')}\right) = \frac{[\alpha Q - P]_+}{[\alpha Q - P]_+(\mathcal{E})}, \tag{67}$$

Thus applying g to an intermediate residual sample gives the correct full state. A target bonus in the case of a full draft acceptance also maps correctly since $g_{\#}Q' = Q$.

Finally, via Sun et al. (2025, Theorem 1) the intermediate output sequence has the law of the target intermediary Markov chain with initial state $y'_0 = g^{-1}(y_0)$ and transition kernel $Q'(\cdot \mid f(g(y')))$. Applying g to each state of this chain gives initial state y_0 and transition kernel

$$g_{\#}Q'(\cdot \mid f(g(y'))) = q(\cdot \mid g(y')),$$

by (61). Hence the full output sequence has the same joint law as the target. □

It should be noted that post-processing by g does not change the number of draft steps accepted in each round of speculative sampling. Thus block verification as per Prop. L.1 maintains its optimality over alternative sample verification strategies (Sun et al., 2025, Theorem 2).

One particular important instantiation of Prop. L.1 is the case where $p = \mathcal{N}(\boldsymbol{\mu}^p, \boldsymbol{\Sigma})$ and $q = \mathcal{N}(\boldsymbol{\mu}^q, \boldsymbol{\Sigma})$ are Gaussian distributions with shared positive definite covariance matrix $\boldsymbol{\Sigma}$. For a fixed pair of distributions p, q, consider the invertible affine map

$$g(y') = \boldsymbol{\mu}^p + \boldsymbol{\Sigma}^{1/2} y', \qquad \text{and} \qquad g^{-1}(y) = \boldsymbol{\Sigma}^{-1/2}(y - \boldsymbol{\mu}^p). \tag{68}$$

This yields the intermediate pre- and post-processing decompositions

$$p = g_{\#}\mathcal{N}(0, \mathrm{Id}), \qquad \text{and} \qquad q = g_{\#}\mathcal{N}(-\Delta, \mathrm{Id}), \qquad \Delta = \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\mu}^p - \boldsymbol{\mu}^q). \tag{69}$$

The latter definition of Δ provides a generalized of (6) which reduces to the $\Sigma = \sigma^2 \mathrm{Id}$ case. Thus `BlockVerification` and `Decomposition` apply in these intermediate coordinates (with $\sigma = 1$), followed by the inverse post-processing g to recover the full output state. We will be considering covariances $\boldsymbol{\Sigma}$ which corresponds to positive scaling of the positive diagonal mass matrix $\mathbf{M}$, *i.e.*, $\boldsymbol{\Sigma}$ will be positive definite.

L.3 ABOBA

We consider the ABOBA integrator as described by (59). Following Kosmala et al. (2026, Section 3.1), we identify that a set of ABOBA_h for a timestep h corresponds to a Gaussian update with an affine pre- and post-processing function. Indeed we break up the integrator into: (1) a pre-processing $\mathrm{A}_{h/2}$ drift step; (2) momentum stochastic update $\mathrm{B}_{h/2} \circ \mathrm{O}_h \circ \mathrm{B}_{h/2}$; and (3) a post-processing $\mathrm{A}_{h/2}$ drift step.

The former first drift step $\mathrm{A}_{h/2}$ fixes a position and covariance:

$$\mathbf{x}_{k+\frac{1}{2}} = \mathbf{x}_k + \frac{h}{2}\mathbf{M}^{-1}\mathbf{m}_k, \quad \text{and} \quad \boldsymbol{\Sigma} = k_B T(1 - \exp(-2\lambda h))\mathbf{M}.$$

At this position, the combined $\mathrm{BOB}_h \doteq \mathrm{B}_{h/2} \circ \mathrm{O}_h \circ \mathrm{B}_{h/2}$ yield an intermediary updates (Kosmala et al., 2026, Equation (7))

$$\begin{aligned} P'_k &= \mathcal{N}(\boldsymbol{\mu}^p_k, \boldsymbol{\Sigma}), \quad \boldsymbol{\mu}^p_k = \exp(-\lambda h)\mathbf{m}_k + \frac{h}{2}(1 + \exp(-\lambda h))\mathbf{F}^p(\mathbf{x}_{k+\frac{1}{2}}) \\ Q'_k &= \mathcal{N}(\boldsymbol{\mu}^q_k, \boldsymbol{\Sigma}), \quad \boldsymbol{\mu}^q_k = \exp(-\lambda h)\mathbf{m}_k + \frac{h}{2}(1 + \exp(-\lambda h))\mathbf{F}^q(\mathbf{x}_{k+\frac{1}{2}}), \end{aligned} \tag{70}$$

where only the mean differs. In other words, the draft and target model for ABOBA corresponds to the sampling of the momentum variable $\mathbf{m}_{k+1}$. Using block verification, we will recover $\mathbf{m}_{k+1}$.

Finally, the last drift update $\mathrm{A}_{h/2}$ shifts the position

$$\mathbf{x}_{k+1} = \mathbf{x}_{k+\frac{1}{2}} + \frac{h}{2}\mathbf{M}^{-1}\mathbf{m}_{k+1}.$$

In the phase-space, the A-update is also invertible. More precisely, the product space phase-space intermediary update corresponds to a Dirac delta at position $\mathbf{x}_{k+\frac{1}{2}}$ (position is not updated in (70)) and a Gaussian momentum update. This allows us to verify the correctness of ABOBA with block verification.

Corollary L.2. *Run Algo. 1 with* `BlockVerification` *and* `Decomposition` *on the momentum kernels in (70) while using the rescaling in (68). Retain the intermediate position and apply the final drift* $\mathrm{A}_{h/2}$ *to each output. The resulting state sequence distributionally matches the target ABOBA integrator.*

Proof. Apply Prop. L.1 with $f = g = \mathrm{A}_{h/2}$ and intermediate phase-space draft kernel $\delta_{\mathbf{x}_{k+1/2}} \otimes P'_k$ and target kernel $\delta_{\mathbf{x}_{k+1/2}} \otimes Q'_k$ (as per (70)). The common position value $\mathbf{x}_{k+1/2}$ leaves verification unchanged when compared to just considering the intermediary momentum kernels of (70) (density ratio between p' and q' in phase-space do not change). Then, by the invertibility of the affine rescaling (68) and the final drift, the proposition directly yields the desired result. □

L.4 OBABO

We consider the OBABO integrator as described by (60). Following the time-staggering construction of Kosmala et al. (2026, Appendix F), consecutive steps can be regrouped by combining adjacent $\mathrm{O}_{h/2}$ updates into O_h. Indeed, examine two consecutive OBABO_h updates:

$$\begin{aligned}&\mathrm{O}_{h/2} \circ \mathrm{B}_{h/2} \circ \mathrm{A}_h \circ \mathrm{B}_{h/2} \circ \mathrm{O}_{h/2} \circ \mathrm{O}_{h/2} \circ \mathrm{B}_{h/2} \circ \mathrm{A}_h \circ \mathrm{B}_{h/2} \circ \mathrm{O}_{h/2} \\ &= \underbrace{\mathrm{O}_{h/2} \circ \mathrm{B}_{h/2}}_{\text{init}} \circ \underbrace{\mathrm{A}_h}_{\text{pre-processing}} \circ \underbrace{\mathrm{B}_{h/2} \circ \mathrm{O}_{h/2} \circ \mathrm{B}_{h/2}}_{\mathrm{BOB}_h} \circ \underbrace{\mathrm{Id}}_{\text{post-processing}} \circ \underbrace{\mathrm{A}_h}_{\text{pre-processing}} \circ \underbrace{\mathrm{B}_{h/2} \circ \mathrm{O}_{h/2}}_{\dots}\end{aligned} \tag{71}$$

After an initialization $\mathrm{B}^q_{h/2} \circ \mathrm{O}_{h/2}$, the OBABO updates follow a familiar pre-processing, momentum stochastic update BOB_h, and post-processing pattern. As such, we can integrate the staggered chain via

$$\bar{\mathsf{Y}}^q_0 = \mathrm{B}_{h/2}\left(\mathrm{O}_{h/2}(y_0)\right), \qquad \bar{\mathsf{Y}}^q_{k+1} = \mathrm{BOB}^q_h\left(\bar{f}(\bar{\mathsf{Y}}^q_k)\right), \tag{72}$$

where we have the pre-processing $\bar{f} = \mathrm{A}_h$ (and no post-processing $\bar{g} = \mathrm{Id}$).

For a staggered input $\bar{y}_k = (\bar{\mathbf{x}}_k, \bar{\mathbf{m}}_k)$, the pre-processing drift A_h fixes

$$\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + h\mathbf{M}^{-1}\bar{\mathbf{m}}_k.$$

The subsequent momentum kernels for the BOB_h step P'_k, Q'_k are precisely those in (70), with $(\mathbf{x}_{k+1/2}, \mathbf{m}_k)$ replaced by $(\bar{\mathbf{x}}_{k+1}, \bar{\mathbf{m}}_k)$ and the same covariance $\boldsymbol{\Sigma}$. Thus we can apply block verification identically to the staggered chain (72) as per the procedure described for ABOBA with pre-processing $\bar{f} = \mathrm{A}_h$ and no post-processing $\bar{g} = \mathrm{Id}$. This yields a speculative sampling procedure for the staggered states, but we want the original un-staggered states which need to be recovered.

The ordinary OBABO state occurs exactly between the two merged $\mathrm{O}_h = \mathrm{O}_{h/2} \circ \mathrm{O}_{h/2}$ half-steps in BOB_h. To recover the original state variable, we consider two consecutive staggered states $\bar{y}_k, \bar{y}_{k+1}$. The momenta entering and leaving the update are given by

$$\mathbf{m}_{\text{in}} = \bar{\mathbf{m}}_k + \frac{h}{2}\mathbf{F}^q(\bar{\mathbf{x}}_{k+1}), \qquad \text{and} \qquad \mathbf{m}_{\text{out}} = \bar{\mathbf{m}}_{k+1} - \frac{h}{2}\mathbf{F}^q(\bar{\mathbf{x}}_{k+1}).$$

The conditional Gaussian reconstruction of Kosmala et al. (2026, Equations (26)–(30)) gives the following reconstruction kernel for the original OBABO state:

$$\mathcal{R}(\cdot \mid \bar{y}_k, \bar{y}_{k+1}) = \delta_{\bar{\mathbf{x}}_{k+1}} \otimes \mathcal{N}\left(\frac{\exp(-\lambda h/2)}{1+\exp(-\lambda h)}(\mathbf{m}_{\text{in}} + \mathbf{m}_{\text{out}}), \frac{1-\exp(-\lambda h)}{1+\exp(-\lambda h)} k_B T \mathbf{M}\right). \tag{73}$$

Through the reconstruction, we can verify the correctness of the OBABO integrator with block verification.

Corollary L.3. *Run Algo. 1 with* `BlockVerification` *and* `Decomposition` *on the staggered momentum kernels above while using the rescaling in (68) and the target initialization in (72). Retain the staggered positions. Given the staggered output sequence $(\bar{\mathsf{Y}}_k)$, set $\mathsf{Y}_0 = y_0$ and reconstruct each Y_{k+1} using (73), independently conditional on the staggered path. The resulting recovered state sequence distributionally matches the target OBABO integrator.*

Proof. Apply Prop. L.1 with $f = \mathrm{A}_h$, $g = \mathrm{Id}$ and intermediate phase-space kernels $\delta_{\bar{\mathbf{x}}_{k+1}} \otimes P'_k$ and $\delta_{\bar{\mathbf{x}}_{k+1}} \otimes Q'_k$ (time staggered (70)). Identically to the proof of Cor. L.2, the common position reduces verification to the updates given by the momentum kernels. Conditional on the initial staggered state, the proposition yields

$$(\bar{\mathsf{Y}}_0, \dots, \bar{\mathsf{Y}}_n) \overset{\mathrm{d}}{=} (\bar{\mathsf{Y}}^q_0, \dots, \bar{\mathsf{Y}}^q_n).$$

Table 9: FCC copper wall-clock speedups relative to sequential UMA-S target integration. Each speedup is averaged over 5 runs and ± uncertainties are propagated standard deviations.

	ABOBA speedup			OBABO speedup		
Atoms	REFLECTION	DECOMP.	BLOCK	REFLECTION	DECOMP.	BLOCK
32	1.928 ± 0.086	1.926 ± 0.191	1.850 ± 0.094	2.053 ± 0.242	2.023 ± 0.229	2.053 ± 0.299
108	1.888 ± 0.115	1.901 ± 0.090	1.908 ± 0.191	1.902 ± 0.189	2.004 ± 0.173	1.951 ± 0.157
256	2.078 ± 0.310	2.081 ± 0.309	2.107 ± 0.317	1.943 ± 0.158	1.991 ± 0.123	2.055 ± 0.106
500	2.496 ± 0.155	2.530 ± 0.116	2.541 ± 0.096	2.547 ± 0.105	2.451 ± 0.109	2.603 ± 0.110
864	2.891 ± 0.117	2.792 ± 0.130	2.976 ± 0.128	2.895 ± 0.118	2.895 ± 0.140	3.028 ± 0.055
1372	3.170 ± 0.095	3.200 ± 0.077	3.301 ± 0.085	3.198 ± 0.078	3.131 ± 0.086	3.275 ± 0.084
2048	3.182 ± 0.087	3.187 ± 0.047	3.365 ± 0.041	3.181 ± 0.127	3.121 ± 0.095	3.342 ± 0.114

For $(\bar{\mathsf{Y}}_0^q, \ldots, \bar{\mathsf{Y}}_n^q)$, it is known that via the reconstruction kernel $\mathcal{R}(\cdot \mid \bar{y}_k, \bar{y}_{k+1})$ ((73)) we can reconstruct the original state (Kosmala et al., 2026, Appendix F). As $(\mathsf{Y}_0, \ldots, \mathsf{Y}_n)$ match the sequential target staggered OBABO, applying the same reconstruction will yield

$$(\mathsf{Y}_0, \ldots, \mathsf{Y}_n) \overset{\mathrm{d}}{=} (\mathsf{Y}_0^q, \ldots, \mathsf{Y}_n^q),$$

where (Y_k^q) is the sequential target OBABO starting from y_0. □

L.5 EXPERIMENTS

We evaluate ABOBA and OBABO integrators on the FCC copper system examined in Kosmala et al. (2026). The system consists of $4s^3$ atoms for $s = 2, \ldots, 8$ with temperature $T = 1500$K, timestep size $h = 1$fs, and friction coefficient $\lambda = 0.001\text{fs}^{-1}$. Unlike the the other experiments on vision datasets and diffusion models, we do not consider a self-speculative approach. Instead, mirroring Kosmala et al. (2026), we utilize the same target pretrained UMA-S model (Wood et al., 2025) (`uma-s-1p1`) and draft pretrained Orb model (Rhodes et al., 2025) (`orb-v3-direct-20-omat`). Each run uses 5000 warm-up steps which was not timed, with an additional 5000 steps which was timed, repeated over five seeds for timing. The code utilized is an adaption of Kosmala et al. (2026)'s official implementation: https://github.com/facebookresearch/LSD.

We compare the three combinations of verification and residual calculations, as per the main text's Table 1. All instances of speculative integration uses synchronous non-pipelined verification, *i.e.*, identical to the diffusion setting. We utilize 8 GH200 GPUs for this experiment: one for drafting, and seven for parallel target verification. This yields a fixed $\gamma = 6$ for draft lengths (with an additional GPU being used for a possible extra accepted step).

Table 9 reports wall-clock speedups relative to sequential target model integration. Block verification yields a 1.85–3.37× speedup for ABOBA and 1.95–3.34× for OBABO. At 2048 atoms, we have a percentage performance increase of block verification versus reflection sample verification R↑B% of 5.7% and 5.1%, respectively. As the number of atoms increase, we generally have a large relative increase when switching to block verification from reflection sample verification (also compared to the original sequential target integration).

To ensure correctness of speculative integrators, we also evaluated the squared maximum mean discrepancy (MMD^2) (Gretton et al., 2012) over a short integration trajectory. For each speculative variant and integrator, we generate 3 independent ensembles of 1000 trajectories (10 steps) for 108 atoms. For each integrator, we set the Gaussian kernel bandwidth to the median pairwise Euclidean distance across a random subset of 1000 trajectories drawn from the pool of all the target ensembles across all 3 random initializations. Table 10 shows similar distances for all speculative and integrator combinations.

Table 10: FCC copper trajectory MMD^2 relative to the sequential target UMA-S with the same integrator (in units of 10^{-3}). Quantities are means with $\pm$ sample standard deviations over 3 comparisons of 1000 ten step trajectories.

	ABOBA MMD^2 ($\times 10^{-3}$)			OBABO MMD^2 ($\times 10^{-3}$)		
Atoms	REFLECTION	DECOMP.	BLOCK	REFLECTION	DECOMP.	BLOCK
108	0.770 ± 0.039	0.798 ± 0.042	0.792 ± 0.046	0.798 ± 0.050	0.794 ± 0.047	0.794 ± 0.072