ManifoldCache: Training-Free Diffusion Acceleration via Constraint Manifold Caching

Prashant Pandey[*,†]
Indian Institute of Technology, Delhi

Devineni Sri Venkatraya Chowdary[*]
Indian School of Mines, Dhanbad

Brejesh Lall
Indian Institute of Technology, Delhi

Abstract

Diffusion models for structured scientific generation must produce samples satisfying hard geometric constraints imposed by physics, chemistry, or biology, yet inference in these settings is prohibitively slow, demanding hundreds to thousands of neural-function evaluations per sample. We unify eight state-of-the-art models spanning medical volumetrics, molecular conformations, protein backbone design, crystal structure prediction, and multi-view 3D scenes under a single abstraction, *Constraint-Manifold Diffusion Models* (CMDMs), in which the target distribution is supported on a manifold defined by an externally specified constraint map. Existing acceleration families transfer poorly to this class: quantization exhausts memory on high-dimensional volumetric operators; pruning breaks constraint fidelity; fast ODE solvers allow trajectories to drift off the constraint manifold; and feature-caching heuristics are blind to constraint geometry and can cause mode confusion in the high-noise regime. We introduce MANIFOLDCACHE, the first training-free, data-free accelerator designed from first principles for CMDMs. The key insight is that the conditional score decomposes orthogonally into a normal component, which enforces constraint satisfaction, and a tangential component, which navigates within the manifold. Building on this structure, we show for a two-mode Gaussian model that the noise-schedule midpoint, where signal and noise have equal energy, separates two regimes: once signal dominates, the error of a cached estimate is provably bounded, while before that point a strictly positive fraction of cached estimates land in the wrong mode, and this fraction does not vanish as the midpoint is approached. We further prove that, under an explicit condition on the balance between the two score components, deeper network blocks admit larger certified cache strides, as a consequence of the score decomposition propagating through block Jacobians. The resulting schedule requires one integer comparison per block per step, zero calibration data, and zero training overhead. Across all eight CMDMs, MANIFOLDCACHE delivers wall-clock speedups of 1.2–$1.9\times$ and stays closest to full-inference quality on each model's primary metric among the feature-caching methods we compare; on Med-DDPM it also offers a better quality–speed trade-off than quantization, pruning, and fast-solver baselines. Our code is available at `https://github.com/prinshul/Manifoldcache.git`.

1 Introduction

Diffusion models have transformed generative modelling across images [Ho et al., 2020, Rombach et al., 2022], audio [Kong et al., 2020], and video [Ho et al., 2022]. Their most consequential frontier lies in the structured spaces of science and engineering: volumetric anatomical atlases [Dorjsembe et al., 2024], molecular conformations and graphs [Liu et al., 2026, Frank et al., 2025, Kong et al.,

[*]Equal contribution.
[†]Correspondence to: `getprashant57@gmail.com`

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

2025], protein backbones [Geffner et al., 2025], periodic crystal lattices [Jiao et al., 2024], and multi-view 3D scenes [Yao et al., 2025, Bahmani et al., 2026]. What unites these applications is the fact that their target distributions are supported not on the full ambient space $\mathcal{X}$, but on a *constraint manifold* $\mathcal{M}_c = \{x \in \mathcal{X} : g(x; c) = 0\}$, where the constraint map g encodes physics, chemistry, geometry, or biology. We call this class *Constraint-Manifold Diffusion Models* (**CMDMs**). CMDMs are central to drug discovery [Kong et al., 2025], *de novo* protein design [Geffner et al., 2025], and crystal structure prediction [Jiao et al., 2024], yet inference is prohibitively slow: Med-DDPM [Dorjsembe et al., 2024] requires 2 min 45 sec per sample on a single A100 GPU, demanding hundreds to thousands of neural-function evaluations (NFEs).

Why existing methods fall short on CMDMs. CMDMs employ fundamentally different backbone architectures (volumetric convolutions, $\mathrm{SE}(3)$-equivariant networks, graph transformers), whose feature dynamics are poorly matched by heuristics designed for pixel-space UNets and vision transformers. ***Post-training quantization*** [Shang et al., 2023, Li et al., 2023, Huang et al., 2024, He et al., 2023, Feng et al., 2025, Gao et al., 2025, Wang et al., 2025] requires a calibration pass that exhausts GPU memory on volumetric operators such as the Conv3D layers in Med-DDPM. ***Pruning*** [Han et al., 2015, Fang et al., 2023] removes weights that collectively enforce g; even 30% sparsity produces samples violating hard geometric constraints. ***Faster ODE solvers*** [Song et al., 2020, Lu et al., 2025] exploit trajectory smoothness that holds only in the clean limit; large steps allow drift off $\mathcal{M}_c$, and for discrete-diffusion variants [Liu et al., 2026] they are categorically inapplicable. *Feature-caching methods* (DeepCache [Ma et al., 2024], SmoothCache [Liu et al., 2025c], TeaCache [Liu et al., 2025a], FORA [Selvaraju et al., 2024], and others) exploit near-stationarity of activations but apply cache-trigger rules blind to constraint geometry, causing over-caching (collapsing constraint fidelity) or under-caching (negligible speedup). Crucially, none identifies the schedule midpoint $\bar{\alpha}_t = \frac{1}{2}$ as a safe-caching boundary, and none assigns block-level strides from depth-dependent score structure.

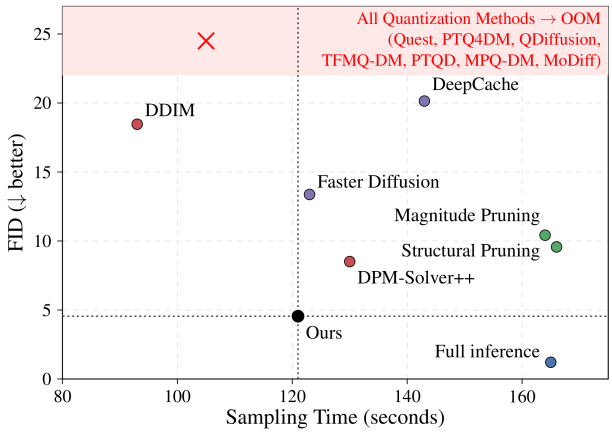

Figure 1: Quality–speed trade-off on Med-DDPM [Dorjsembe et al., 2024] across quantization, pruning, solver, and caching baselines. MANIFOLDCACHE achieves 2 min 1 sec per sample (full inference: 2 min 45 sec); all baselines either degrade quality, slow down, or incur OOM.

We propose **MANIFOLDCACHE**, the first inference-acceleration framework for CMDMs, derived from the orthogonal decomposition of the conditional score into a normal component $s_t^{\perp}$ (enforcing constraint satisfaction) and a tangential component $s_t^{\parallel}$ (navigating along $\mathcal{M}_c$). Theorem 1 shows, for a two-mode Gaussian model, that $T^* = \max\{t : \bar{\alpha}_t \geq \frac{1}{2}\}$, which depends only on the noise schedule, separates two regimes: for $\tau \leq T^*$ the expected squared error of the frozen Tweedie estimate is at most $\frac{1}{2} \operatorname{tr} \Sigma_c$ and the probability of landing in the wrong mode is small for well-separated modes, while for $\tau > T^*$ a strictly positive fraction of frozen estimates lands in the wrong manifold mode for every within-mode spread. Theorem 2 establishes that within the safe phase, deeper network blocks admit provably larger certified cache strides, as a formal consequence of the score decomposition propagating through block Jacobians when the normal-to-tangential power ratio κ_t falls below an explicit threshold κ^*. MANIFOLDCACHE instantiates both results as a lightweight, training-free schedule: one integer comparison per block per step, zero calibration, zero training overhead. On Med-DDPM this yields $\approx 1.4\times$ wall-clock speedup with the lowest FID among the caching methods we compare (Figure 1, Table 5).

Contributions. **(1)** A unified framework (§3.1) showing that eight state-of-the-art models (Med-DDPM [Dorjsembe et al., 2024], SV4D 2.0 [Yao et al., 2025], DemoDiff [Liu et al., 2026], Proteina [Geffner et al., 2025], DiT-MC [Frank et al., 2025], UniMoMo [Kong et al., 2025], DiffCSP++ [Jiao et al., 2024], Lyra [Bahmani et al., 2026]) are all special cases of the CMDM construction. **(2)** A schedule-derived safe-caching threshold T^* (Theorem 1, §3.3) that separates a regime of provably bounded caching error from one with a provably non-vanishing mode-flip probability,

depending only on the noise schedule. **(3)** Depth-monotone cacheability via Jacobian decomposition (Theorem 2, §3.4), providing the first theoretical justification for depth-adaptive caching with an explicit regime condition. **(4)** MANIFOLDCACHE (§3.4, Figure 1), which on the eight CMDMs is faster than every baseline while scoring best among the accelerators on the primary quality metric.

2 Related Work

Constraint-manifold diffusion models. A growing class of diffusion models generates samples satisfying hard geometric, physical, or biological constraints [Dorjsembe et al., 2024, Jiao et al., 2024, Yao et al., 2025, Liu et al., 2026, Geffner et al., 2025, Frank et al., 2025, Kong et al., 2025]. Their backbones (volumetric 3D ConvNets, $\mathrm{SE}(3)$-equivariant networks, graph transformers, and multi-view ViTs) are heavier than standard 2D UNets; constraint fidelity degrades sharply under step reduction since the probability flow is only tangent to $\mathcal{M}_c$ in the clean limit; and guidance strategies multiply NFEs by 2–4×. Together, these factors significantly increase the inference and generation times for CMDMs.

Acceleration methods. *Post-training quantization* [Shang et al., 2023, Li et al., 2023, Huang et al., 2024, He et al., 2023, Feng et al., 2025, Gao et al., 2025, Wang et al., 2025] reduces per-step compute via INT8/FP8/INT4 weight and activation compression. On volumetric CMDMs, calibration passes exhaust GPU memory, resulting in Out of Memory (OOM) (mainly block reconstruction approaches), and quantization noise accumulates along directions normal to $\mathcal{M}_c$, where any error directly manifests as constraint violation. *Pruning.* Magnitude [Han et al., 2015] and structural [Fang et al., 2023] pruning remove low-saliency weights to reduce per-step cost. On CMDMs every weight participates in satisfying the constraint map g; 30% sparsity already drives generation quality below an acceptable level, and neither variant reduces the NFE count. *Fast solvers.* DDIM [Song et al., 2020] and DPM-Solver++ [Lu et al., 2025] reduce NFEs by taking larger probability-flow ODE integration steps, exploiting trajectory smoothness in unconstrained image models. On CMDMs, large steps allow trajectories to drift off $\mathcal{M}_c$ before any corrective force restores them, producing constraint-violating samples. For discrete-diffusion variants such as DemoDiff [Liu et al., 2026], which operate on token simplices, high-order ODE solvers are categorically inapplicable.

Feature caching. DeepCache [Ma et al., 2024], FasterDiffusion [Li et al., 2024], FORA [Selvaraju et al., 2024], SmoothCache [Liu et al., 2025c], TeaCache [Liu et al., 2025a], AB-Cache [Yu et al., 2025], PFC [Sommer et al., 2025], ToCa [Zou et al., 2024], dKV-Cache [Ma et al., 2025], TaylorSeer [Liu et al., 2025b], PAB [Zhao et al., 2025], AdaCache [Kahatapitiya et al., 2025], and FrDiff [So et al., 2024] exploit the near-stationarity of hidden activations between adjacent denoising steps via fixed intervals, calibration thresholds, input-difference proxies, or numerical extrapolation. These methods share a common limitation: none uses the geometric structure of $\mathcal{M}_c$ as a caching criterion, and none identifies $\bar{\alpha}_t = \frac{1}{2}$ as a safe-caching boundary, so they may reuse features in the high-noise phase where, as Theorem 1 shows for a two-mode model, the modes are not yet resolved and reuse leads to mode confusion with non-vanishing probability. MANIFOLDCACHE *is the first framework to derive a depth-adaptive caching schedule with provable error bounds from constraint geometry, requiring no training, no calibration data, and no per-step runtime measurement.*

Speculative sampling. Speculative methods for diffusion models [De Bortoli et al., 2025] and diffusion language models [Agrawal et al., 2025] draft future denoising states with a cheaper proposal and then accept or reject them in a verification step, which reduces the number of expensive model calls while preserving the output distribution. MANIFOLDCACHE works on a different axis: it keeps every denoising step and lowers the cost of each step by reusing block outputs according to a fixed schedule, with no draft model or verifier. The two approaches are therefore complementary; Appendix E discusses the differences in more detail and compares them on Proteina.

3 Methodology

MANIFOLDCACHE rests on a single observation: the physics or biology that defines a constraint manifold also determines *when* and *where* a denoiser's computation can be safely skipped. We formalise this in four steps: **(i)** defining the CMDM class, **(ii)** decomposing the score geometrically, **(iii)** deriving a universal safe-caching threshold, and **(iv)** assembling a closed-form depth-adaptive schedule. Figure 2 illustrates the main ideas.

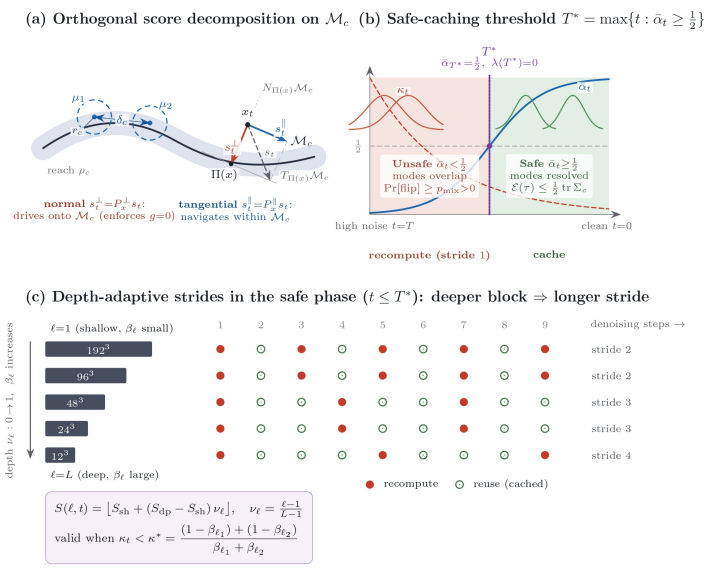

Figure 2: Geometric picture behind MANIFOLDCACHE. **(a)** Near $\mathcal{M}_c$, the score splits into a normal part $s_t^\perp$, which pulls the sample toward $\mathcal{M}_c$, and a tangential part $s_t^\parallel$, which moves it within $\mathcal{M}_c$ toward one of the valid modes (here μ_1 and μ_2, separated by δ_c; §3.2). **(b)** The threshold $T^* = \max\{t : \bar{\alpha}_t \geq \frac{1}{2}\}$ splits the schedule: for $t > T^*$ every block is recomputed, and caching is enabled only for $t \leq T^*$ (Theorem 1); the κ_t curve is schematic. **(c)** For $t \leq T^*$, the cache stride grows with block depth (Eq. (6)): shallow blocks are recomputed often, and deep blocks are reused for longer.

3.1 Constraint-Manifold Diffusion Models

Let $\mathcal{X} \subseteq \mathbb{R}^D$ be the *ambient space*, a voxel grid, sequence of residue coordinates, periodic unit cell, or latent video tensor depending on the application. Structured generative models do not place mass freely across $\mathcal{X}$; they confine samples to a surface carved out by domain knowledge.

Definition 1 (CMDM)**.** *A* Constraint-Manifold Diffusion Model *is specified by a C^2 constraint map $g : \mathcal{X} \times \mathcal{C} \to \mathbb{R}^k$ and a conditioning signal $c \in \mathcal{C}$. The target distribution $p(\cdot \mid c)$ is supported on the* constraint manifold $\mathcal{M}_c := \{x \in \mathcal{X} : g(x; c) = 0\}$. *We assume $\nabla_x g(x; c)$ has full rank k at every $x \in \mathcal{M}_c$ (submersion condition), so by the Implicit Function Theorem $\mathcal{M}_c$ is a closed C^2 submanifold of $\mathcal{X}$ of dimension $d_c := D - k$.*

State-of-the-art CMDM models across various domains are special cases: Med-DDPM [Dorjsembe et al., 2024] ($\mathcal{X} = \mathbb{R}^{4\times192\times192\times144}$, g enforces anatomical intensity and continuity), DiffCSP++ [Jiao et al., 2024] (g encodes crystallographic space-group symmetry), Proteina [Geffner et al., 2025] ($\mathcal{X} = \mathbb{R}^{3N}$, the C_α coordinates of N residues; g encodes Ramachandran validity and steric clash avoidance), SV4D 2.0 [Yao et al., 2025] and Lyra [Bahmani et al., 2026] (g imposes multi-view photometric consistency). Despite radically different ambient spaces, all share three properties our theory exploits: g is externally specified, not learned **(P1)**; codimension $k \gg 1$ so a random point in $\mathcal{X}$ violates the constraints with probability approaching one **(P2)**; and c is exact at every diffusion step **(P3)**. The constraint map g describes where the data lie: it is a modelling assumption about the support of $p(\cdot \mid c)$, not an operation performed at inference. MANIFOLDCACHE never evaluates g, projects onto $\mathcal{M}_c$, or checks constraint satisfaction; it reads only the timestep and the block index. How a model keeps its samples on $\mathcal{M}_c$ (through training data, conditioning, or an architecture that builds the constraint in) therefore does not affect how the schedule is applied. Table 9 in Appendix B lists the constraint and this mechanism for each of the eight models.

3.2 Orthogonal Score Decomposition

We adopt a variance-preserving (VP) forward process $q(x_t \mid x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\, x_0, (1 - \bar{\alpha}_t)\mathbf{I}_D)$, where $\bar{\alpha}_t \in (0, 1]$ is strictly decreasing with $\bar{\alpha}_0 = 1$. Denoising runs from $t = T$ (near-isotropic noise) to $t = 0$ (clean data on $\mathcal{M}_c$). The conditional score is $s_t(x; c) := \nabla_x \log q_t(x \mid c)$, where $q_t(x \mid c) := \int q(x_t \mid x_0)\, p(x_0 \mid c)\, dx_0$.

Geometric structure of the score. At high noise, x_t almost surely violates g; the score must point the trajectory back toward $\mathcal{M}_c$. At low noise, x_t is already near $\mathcal{M}_c$, and what remains is to select which valid sample to produce by moving *within* $\mathcal{M}_c$. These two roles are geometrically orthogonal. Since $\mathcal{M}_c$ is a closed C^2 submanifold with positive reach $\rho_c > 0$ (Assumption (A2), §A.2), Federer's theorem [Federer, 1959] guarantees a tubular neighbourhood of radius ρ_c in which the nearest-point projection $\Pi : \mathcal{X} \to \mathcal{M}_c$ is smooth. For any x in this tube, the ambient tangent space splits as $T_x\mathcal{X} = T_{\Pi(x)}\mathcal{M}_c \oplus N_{\Pi(x)}\mathcal{M}_c$, where the normal space $N_{\Pi(x)}\mathcal{M}_c = \operatorname{span}\{\nabla_x g_j(\Pi(x); c)\}_{j=1}^{k}$ is spanned by the constraint gradients (well-defined by the submersion condition). Letting $P_x^{\perp}$ and $P_x^{\parallel}$ denote the orthogonal projectors onto the normal and tangential subspaces, the conditional score decomposes as follows [Stanczuk et al., 2024]:

$$s_t(x; c) \;=\; \underbrace{P_x^{\perp}\, s_t(x; c)}_{s_t^{\perp}\ \text{(drives toward } \mathcal{M}_c)} \;+\; \underbrace{P_x^{\parallel}\, s_t(x; c)}_{s_t^{\parallel}\ \text{(navigates within } \mathcal{M}_c)} . \tag{1}$$

As $t \to 0$, the Tweedie posterior mean $\widehat{x}_0(x_t, t; c) := \mathbb{E}[x_0 \mid x_t, c]$ converges onto $\mathcal{M}_c$ and $g(x_t; c) \to 0$. We measure the balance between the two components through the *normal-to-tangential power ratio* $\kappa_t := \mathbb{E}\|s_t^{\perp}\|^2 / \mathbb{E}\|s_t^{\parallel}\|^2 \geq 0$. Theorem 2 below shows how κ_t determines which network blocks change least between adjacent steps; MANIFOLDCACHE itself never measures κ_t at runtime.

3.3 The Safe-Caching Threshold T^*

Define the log signal-to-noise ratio $\lambda(t) := \log[\bar{\alpha}_t/(1-\bar{\alpha}_t)]$ and the *schedule-derived threshold*

$$T^* \;:=\; \max\{t : \bar{\alpha}_t \geq \tfrac{1}{2}\} \;=\; \max\{t : \lambda(t) \geq 0\}, \tag{2}$$

the latest step at which signal energy is still at least as large as noise energy. T^* is a scalar computable in closed form from the noise schedule, independent of $\mathcal{M}_c$, c, the network, and any data.

Theorem 1 (Manifold-aware caching threshold)**.** *Under Assumption* (A3) *(§A.2), let Σ_c be the total covariance of $p(\cdot \mid c)$ restricted to $T\mathcal{M}_c$ with eigenvalues $\{\lambda_i\}_{i=1}^{d_c}$ all satisfying $\lambda_i \geq 1$* (A3.ii)*, and let $x_0^{\tau} := \widehat{x}_0(x_\tau, \tau; c)$ be the frozen Tweedie estimate with cache error $\mathcal{E}(\tau) := \mathbb{E}\|x_0^{\tau} - x_0\|^2$. Write $\delta_c := \|\mu_1 - \mu_2\|$ for the mode separation, σ_u^2 for the within-mode variance along the axis joining the two modes (§A.1), $K \in \{1, 2\}$ for the mode from which x_0 is drawn, Φ for the standard Gaussian CDF, and $p_{\text{flip}}(\tau) := \Pr[\Pi_{\mathcal{M}_c}(x_0^{\tau}) \neq \mu_K]$ for the probability that the frozen estimate is closer to the other mode. For every step $\tau \geq 1$,*

$$p_{\text{flip}}(\tau) = \Phi\Big(-\frac{\delta_c}{2}\sqrt{\frac{\bar{\alpha}_\tau}{V_\tau^u}}\Big), \qquad V_\tau^u := 1 - \bar{\alpha}_\tau + \bar{\alpha}_\tau \sigma_u^2, \tag{3}$$

which is continuous and strictly decreasing in $\bar{\alpha}_\tau$.

Safe *($\tau \leq T^*$): the cache error satisfies $\mathcal{E}(\tau) \leq \sum_i (1-\bar{\alpha}_\tau)\lambda_i/(\bar{\alpha}_\tau\lambda_i + 1 - \bar{\alpha}_\tau) \leq \frac{1}{2}\operatorname{tr}\Sigma_c$ (Lemma 1), and $p_{\text{flip}}(\tau) \leq \Phi(-\delta_c\sqrt{\bar{\alpha}_\tau}/2) \leq \Phi(-\delta_c/(2\sqrt{2}))$.*

Unsafe *($\tau > T^*$): mode confusion has positive probability, $p_{\text{flip}}(\tau) > \Phi(-\delta_c/2) =: p_{\text{mix}} > 0$, and this bound does not vanish as $\bar{\alpha}_\tau \uparrow \frac{1}{2}$.*

Role of *T^*: $p_{\text{flip}}(\tau) \geq p_{\text{mix}}$ holds if and only if $\bar{\alpha}_\tau \leq 1/(2-\sigma_u^2)$. Hence $\bar{\alpha}_\tau = \frac{1}{2}$, the level that defines T^*, is the largest signal level at which $p_{\text{flip}}(\tau) \geq p_{\text{mix}}$ holds for every within-mode spread $\sigma_u \in (0, 1]$.*

The safe bound says that freezing the Tweedie estimate at any $\tau \leq T^*$ costs at most half the within-manifold variance, and that the frozen estimate lands in the wrong mode with probability at most $\Phi(-\delta_c/(2\sqrt{2}))$, which is small for well-separated modes. Freezing the estimate is the extreme case of caching, in which nothing is recomputed after step τ. The unsafe bound says that at every step $\tau > T^*$, and for every within-mode spread, more than a fraction p_{mix} of frozen estimates land in the wrong mode: when $\bar{\alpha}_\tau < \frac{1}{2}$, noise dominates signal and the noisy state cannot reliably tell the modes apart. Because p_{flip} decreases continuously, T^* does not mark a sudden jump; it is the largest signal level at which the unsafe bound is guaranteed for every within-mode spread. The two-mode model is a simplified stand-in for the multimodal distributions of real CMDMs (§A.6). The proof is in §A.4; caching is enabled only for $t \leq T^*$.

3.4 Depth-Adaptive Caching Schedule

Theorem 1 tells us *when* to cache. A uniform stride across all blocks wastes the available budget: deep blocks, operating at coarse resolution and tracking slowly-evolving global structure, are far

more cacheable than shallow blocks tracking rapidly-changing fine-grained details. We formalise this with a second theorem.

Let the denoiser consist of L residual blocks $\{f_\ell\}_{\ell=1}^{L}$ with features $h_\ell(x,t,c) \in \mathbb{R}^{D_\ell}$ and $h_0 := x$. The *per-block drift* $\delta_\ell(t) := \|h_\ell(x_{t'},t,c) - h_\ell(x_t,t,c)\|$ measures output change between adjacent steps $t > t'$. Block ℓ is *ε-cacheable with stride S* if $\mathbb{E}\sum_{k=1}^{S-1}\delta_\ell(t-k\Delta t) \le \varepsilon$.

The score decomposition (1) passes through the input Jacobian $J_\ell := \nabla_x h_\ell$: input perturbations $\Delta x = P_x^{\perp}\Delta x + P_x^{\parallel}\Delta x$ produce feature perturbations $J_\ell \Delta x$. By Assumption (A5) (§A.2), J_ℓ has directional upper-Lipschitz constants $L_\ell^{\perp}$ and $L_\ell^{\parallel}$ in the normal and tangential directions. Under unit-gain normalisation $L_\ell^{\perp} + L_\ell^{\parallel} = 1$, set $\beta_\ell := L_\ell^{\perp} \in (0,1)$, the *normal sensitivity* of block ℓ. The depth-ordering condition $\beta_{\ell_1} < \beta_{\ell_2}$ for $\ell_1 < \ell_2$ (Assumption (A5)) captures the backbone hierarchy; in a U-Net such as Med-DDPM, shallow blocks at resolution $192\times192\times144$ must respond to rapidly changing local structure, while deep blocks at $12\times12\times9$ attend to global, constraint-relevant directions. Define the *drift envelope* $B(\beta;\kappa) := \beta^2\kappa + (1-\beta)^2$.

Theorem 2 (Depth-monotone cacheability)**.** *Under Assumptions* (A1)*,* (A2)*, and* (A4)–(A6) *(§A.2), for any $t \le T^*$ the mean-square drift of every block ℓ satisfies*

$$\mathbb{E}\,\delta_\ell(t)^2 \;\le\; 2B(\beta_\ell;\kappa_t)\,\Theta_t\,\Delta t^2 + O(\Delta t^3), \tag{4}$$

where $\Theta_t > 0$ is defined in §A.5. For any $\ell_1 < \ell_2$ with $\beta_{\ell_1} < \beta_{\ell_2}$, the envelope of the deeper block is strictly smaller, $B(\beta_{\ell_2};\kappa_t) < B(\beta_{\ell_1};\kappa_t)$, if and only if

$$\kappa_t \;<\; \kappa^* \;:=\; \frac{(1-\beta_{\ell_1})+(1-\beta_{\ell_2})}{\beta_{\ell_1}+\beta_{\ell_2}} \;>\; 0. \tag{5}$$

When (5) *holds, the certified strides $\widehat{S}_\ell$ of §A.5 satisfy*

$$\frac{\widehat{S}_{\ell_2}-1}{\widehat{S}_{\ell_1}-1} = \sqrt{\frac{B(\beta_{\ell_1};\kappa_t)}{B(\beta_{\ell_2};\kappa_t)}}\,\big(1+O(\Delta t)\big),$$

which exceeds one for small enough Δt: deeper blocks receive strictly longer certified strides. Under the additional transversality condition (A5′)*, the deeper block also has strictly smaller actual drift, $\mathbb{E}\,\delta_{\ell_2}(t)^2 < \mathbb{E}\,\delta_{\ell_1}(t)^2$, whenever $B(\beta_{\ell_1};\kappa_t)/B(\beta_{\ell_2};\kappa_t) > 2/\big((1-\alpha)\gamma_*^2\big)$ and Δt is small enough, with α and γ_* as in* (A5′).

The threshold κ^* is explicit and depends only on β_{ℓ_1} and β_{ℓ_2}. Theorem 2 is conditional: it favours longer strides for deeper blocks when the tangential part of the score dominates ($\kappa_t < \kappa^*$), and we do not derive this condition from the noise schedule. The inversion ablation in §5 tests the resulting depth ordering directly. The proof is in §A.5.

The MANIFOLDCACHE schedule. Both theorems together yield a complete closed-form policy. Let $\nu_\ell := (\ell-1)/(L-1) \in [0,1]$ be the normalised block depth and $1 \le S_{\mathrm{sh}} \le S_{\mathrm{dp}}$ be integer strides (e.g. 2 and 4). The stride assigned to block ℓ at step t is

$$S(\ell,t) \;=\; \begin{cases} 1, & t > T^*, \\ \big\lfloor S_{\mathrm{sh}} + (S_{\mathrm{dp}} - S_{\mathrm{sh}})\nu_\ell \big\rfloor, & t \le T^*, \end{cases} \tag{6}$$

where stride 1 means always recompute. Block ℓ is reused at step $t \le T^*$ iff $t_\ell^{\mathrm{last}} - t < S(\ell,t)$, where t_ℓ^{last} is the step at which it was last computed; otherwise it is recomputed and the cache updated. The policy costs one integer comparison per block per step: T^* is read from the noise schedule, ν_ℓ from the block index, and no calibration or profiling is required.

4 Implementation and Dataset Details

4.1 Models, Datasets, and Metrics

We evaluate MANIFOLDCACHE on eight CMDMs (§3.1) across five scientific domains using official open-source checkpoints. **Med-DDPM** [Dorjsembe et al., 2024] is a Conv3D U-Net DDPM for segmentation-conditioned semantic 3D brain MRI synthesis on $192\times192\times144$ BraTS 2021 volumes, evaluated with volumetric 3D **FID**. **SV4D 2.0** [Yao et al., 2025] is a multi-view video diffusion transformer with joint 3D/frame attention for monocular-video-to-4D asset generation, evaluated on Objaverse [Yao et al., 2025] over 8 view pairs using **PSNR**, **SSIM**, and **LPIPS**, computed against full inference (§5). **Lyra** [Bahmani et al., 2026] is an NVIDIA self-distilled video diffusion model featuring

Table 1: **Proteina** [Geffner et al., 2025] (de novo C_α backbone generation, flow-matching DiT trained on AFDB): runtime, mean self-consistency RMSD (**scRMSD**; ProteinMPNN → ESMFold), and peak **VRAM**. The (w) row runs the first fraction w of the 100 steps with full computation before enabling caching; (T^*) corresponds to $w = 0.15$. MANIFOLDCACHE $(T^*)\,(2, 4)$ has the lowest scRMSD among the accelerators (2.04 vs. 1.93 for full inference).

Method	Time (sec) ↓	Mean scRMSD ↓	VRAM (MiB) ↓
Full Inference	90.47	**1.93**	15438
SmoothCache [Liu et al., 2025c]	60.96	2.11	16430
FORA [Selvaraju et al., 2024]	70.62	2.31	15980
TeaCache [Liu et al., 2025a]	61.10	8.07	17800
TaylorSeer [Liu et al., 2025b]	61.08	9.11	16700
MANIFOLDCACHE $(w{=}0.1)\,(2, 4)$	45.12	2.35	16000
MANIFOLDCACHE $(T^*)\,(3, 5)$	43.76	2.85	16190
MANIFOLDCACHE $(T^*)\,(2, 5)$	45.10	2.61	16197
MANIFOLDCACHE $(T^*)\,(2, 4)$	46.78	**2.04**	16190

a camera-controlled GEN3C-based video DiT for single-image-to-3D Gaussian Splatting scene generation, evaluated using rendered novel views with **PSNR**, **SSIM**, and **LPIPS**, computed against full inference (§5). **Proteina** [Geffner et al., 2025] is a 400M-parameter non-equivariant flow-matching transformer for hierarchical-CATH-conditioned C_α protein backbone generation trained on AFDB, evaluated by the ProteinMPNN→ESMFold self-consistency pipeline using mean **scRMSD**, where scRMSD$<$2 Å denotes designable samples. **DiT-MC** [Frank et al., 2025] adapts diffusion transformers to graph-conditioned 3D molecular conformer generation and is evaluated on GEOM-QM9 using mean **COV-R** and **COV-P** at 0.5 Å RMSD. **UniMoMo** [Kong et al., 2025] is an E(3)-equivariant geometric latent diffusion model for unified peptide, antibody, and small-molecule binder generation from block-graph representations, evaluated on PepBench using mean **AAR**, median **C-RMSD**, and median **L-RMSD**. **DiffCSP++** [Jiao et al., 2024] is a periodic equivariant diffusion model jointly denoising lattices, fractional coordinates, and atom types under explicit space-group/Wyckoff constraints for crystal structure prediction, evaluated on MP-20 with **Match Rate** and **RMSD**. **DemoDiff** [Liu et al., 2026] is a 0.7B-parameter discrete graph diffusion transformer for in-context molecular design on Node-Pair-Encoding motif-tokenised graphs, pretrained on 1.6M ChEMBL+polymer tasks and evaluated on its 33-task benchmark using task-averaged **Score** and ICL **Consistency Score** [Liu et al., 2026].

4.2 Hardware and Hyperparameters

All inference experiments are run on a single NVIDIA A100 40GB GPU with the official inference scripts and default schedulers, samplers, guidance scales, and step counts of each model. MANIFOLDCACHE is parameterised by $(S_{\text{sh}}, S_{\text{dp}})$, the shallow and deep stride bounds in Eq. (6), and by the cache-enable cutoff which defaults to T^* from Eq. (2). We also report an ablation that enables caching earlier: on Med-DDPM and Proteina, (w) runs the first fraction w of the denoising steps with full computation and enables caching for the rest, with $w \in \{0.1, 0.2, 0.3, 0.4\}$ on Med-DDPM and $w = 0.1$ on Proteina. For UniMoMo and Lyra we use $(2, 4)$ and $(2, 3)$ respectively; for all other models we sweep $(S_{\text{sh}}, S_{\text{dp}}) \in \{(2, 3), (2, 4), (2, 5), (3, 5)\}$, together with the inverted setting $(4, 2)$ as an ablation, and report the configurations shown in the tables.

4.3 Caching Baselines and Their Applicability

The CMDM literature spans four backbone families: volumetric 3D U-Nets (Med-DDPM), SE(3)/E(3)-equivariant networks (UniMoMo, DiffCSP++), graph and flow-matching transformers (DemoDiff, DiT-MC, Proteina), and multi-view (4D) video models (SV4D 2.0, Lyra), and existing caching methods are designed against very different architectural assumptions. We therefore restrict each table to baselines that are *actually runnable* on that backbone with their published assumptions intact; the rest are categorically inapplicable. We summarise the per-model selection rationale below; **MANIFOLDCACHE is backbone-agnostic and runs on all eight.**

For U-Net-style CMDMs, including **Med-DDPM** and the convolutional VAE branches of **SV4D 2.0** and **Lyra**, we use **DeepCache** [Ma et al., 2024] and, for Med-DDPM, **FasterDiffusion** [Li et al., 2024], which explicitly cache encoder/decoder or skip-connected U-Net features. Transformer-based CMDMs such as **Proteina**, **DiT-MC**, and **DemoDiff** instead use DiT-native cachers including **FORA** [Selvaraju et al., 2024], **TeaCache** [Liu et al., 2025a], **SmoothCache** [Liu et al., 2025c], **TaylorSeer** [Liu et al., 2025b], **ToCa** [Zou et al., 2024], **AB-Cache** [Yu et al., 2025], and **PFC** [Sommer et al., 2025], since these operate on isotropic transformer block outputs or token-feature trajectories

Table 2: **DiT-MC** [Frank et al., 2025] (3D molecular conformer generation on GEOM-QM9, graph-conditioned DiT): runtime, mean recall (**COV-R**) and precision (**COV-P**) at a 0.5 Å threshold, and **VRAM**.

Method	Time (sec) ↓	COV-R (mean) ↑	COV-P (mean) ↑	VRAM (MiB) ↓
Full Inference	621	**0.9695**	**0.9569**	30706
TeaCache [Liu et al., 2025a]	565	0.9473	0.9424	32491
AB-Cache [Yu et al., 2025]	501	0.9460	0.9430	31900
PFC (ABC-Forecasting) [Sommer et al., 2025]	561	0.9517	0.9428	32998
MANIFOLDCACHE $(T^*)\,(2,4)$	**377.3**	0.9693	0.9563	32360

Table 3: **UniMoMo** [Kong et al., 2025] (peptide binder design via E(3)-equivariant latent diffusion): runtime, amino-acid recovery (**AAR**, mean), complex (**C-RMSD**, median, C_α after target alignment), ligand (**L-RMSD**, median, after peptide alignment), and **VRAM**.

Method	Time (sec) ↓	AAR ↑	C-RMSD ↓	L-RMSD ↓	VRAM (MiB) ↓
Full Inference	687	**0.2562**	**3.14**	**1.59**	28072
SmoothCache [Liu et al., 2025c]	578	0.1890	6.10	6.04	28126
DeepCache [Ma et al., 2024]	555	0.1622	4.98	3.10	28960
MANIFOLDCACHE $(T^*)\,(2,4)$	**540**	0.2550	3.29	1.75	28148

compatible with graph and flow-matching transformers. **DemoDiff** additionally uses **dKV-Cache** [Ma et al., 2025], the only published cacher designed for discrete masked-token diffusion with bidirectional attention. For **UniMoMo** we use **SmoothCache** and **DeepCache**; other DiT cachers violate its E(3)-equivariant message-passing structure. For **DiffCSP++** we compare against **DeepCache** only, because current DiT/video-DiT methods assume non-periodic isotropic attention or separable spatial–temporal attention absent in periodic equivariant diffusion. Methods such as **PAB** [Zhao et al., 2025] and **AdaCache** [Kahatapitiya et al., 2025] are only applicable to video DiTs with explicit spatial/temporal attention splits, while U-Net cachers are excluded from isotropic transformers lacking skip connections, and continuous-trajectory methods are unsuitable for discrete motif-token diffusion. We use PAB and AdaCache for Lyra, and we additionally compare with **FrDiff** [So et al., 2024] on SV4D 2.0.

5 Results and Analysis

We report runtime and generation quality through various metrics across all eight CMDMs in Tables 1–8. Each table is ordered as follows: Full Inference, applicable caching baselines, and then MANIFOLDCACHE configurations. Figure 4 (Appendix C) plots the quality–speed trade-off for all eight models, Appendix D reports FLOPs on six models and NFE on two, and Appendix E compares MANIFOLDCACHE with speculative sampling. **MANIFOLDCACHE consistently delivers the best quality–speed trade-off, often *strictly* dominating every baseline on both axes simultaneously.**

On Med-DDPM (Table 5), enabling caching earlier than T^* worsens FID at every point of the warm-up sweep: FID is 4.55 at T^* and rises to 4.83, 5.51, 7.01, and 12.56 as the warm-up fraction w shrinks from 0.4 to 0.1. On Proteina (Table 1), enabling caching before T^* (w=0.1, whereas T^* corresponds to w=0.15) raises mean scRMSD from 2.04 to 2.35. This is the behaviour Theorem 1 predicts: **reusing features while noise still dominates ($t > T^*$) risks mode confusion**. When more speedup is needed and some loss of quality is acceptable, the warm-up can still be tuned as a deliberate trade-off: on Med-DDPM (Table 5), shortening it from T^* to w=0.1 raises the speedup from $1.4\times$ to $1.9\times$, with FID rising from 4.55 to 12.56.

The $(S_{\text{sh}}, S_{\text{dp}}) = (4, 2)$ *inversion* ablation isolates whether deeper-block-skipping is the source of the gain predicted by Theorem 2. On Med-DDPM (Table 5), $(4, 2)$ raises FID from 4.55 to 5.63. Because Med-DDPM's shallow blocks run at full resolution, the inverted schedule also skips more

Table 4: **DiffCSP++** [Jiao et al., 2024] (space-group constrained crystal structure prediction on MP-20): **Match Rate** and **RMSD**, runtime, and **VRAM**. MANIFOLDCACHE $(T^*)\,(2,4)$ has the highest Match Rate and the lowest RMSD among the accelerators.

Method	Match Rate ↑	RMSD ↓	Time (sec) ↓	VRAM (MiB) ↓
Full Inference	**0.7840**	**0.0289**	1269	1886
DeepCache [Ma et al., 2024]	0.7370	0.0464	957	1894
MANIFOLDCACHE $(T^*)\,(3,5)$	0.7450	0.0459	**883**	2014
MANIFOLDCACHE $(T^*)\,(2,4)$	**0.7550**	**0.0327**	938	2014

Table 5: **Med-DDPM** [Dorjsembe et al., 2024] (semantic 3D brain MRI synthesis on BraTS, $192\times192\times144$ Conv3D U-Net): runtime per sample and 3D **FID** between generated and real volumes. (w) rows run the first fraction w of the 250 steps with full computation before enabling caching; MANIFOLDCACHE $(T^*)\,(2,4)$ is $1.4\times$ faster than full inference *and* attains the lowest FID among all accelerators; Figure 3 shows samples.

Method	Time per sample (sec) ↓	FID ↓	VRAM (MiB) ↓
Full Inference	165	**1.21**	22392
DeepCache [Ma et al., 2024]	143	20.14	22994
FasterDiffusion [Li et al., 2024]	123	13.37	25780
MANIFOLDCACHE $(w{=}0.1)\,(2,4)$	85	12.56	30411
MANIFOLDCACHE $(w{=}0.2)\,(2,4)$	95	7.01	30498
MANIFOLDCACHE $(w{=}0.3)\,(2,4)$	101	5.51	30498
MANIFOLDCACHE $(w{=}0.4)\,(2,4)$	110	4.83	30497
MANIFOLDCACHE $(T^*)\,(4,2)$	107	5.63	30531
MANIFOLDCACHE $(T^*)\,(3,5)$	109	5.29	30528
MANIFOLDCACHE $(T^*)\,(2,5)$	120	4.57	30550
MANIFOLDCACHE $(T^*)\,(2,4)$	121	**4.55**	30611

Table 6: **DemoDiff** [Liu et al., 2026] (in-context molecular design, 0.7B discrete graph diffusion transformer with motif-token NPE): runtime, official task-averaged design **Score**, and ICL **Consistency Score**. MANIFOLDCACHE $(T^*)\,(2,4)$ retains 88% of the design Score at $1.5\times$ speedup; alternatives lose $24{-}46\%$.

Method	Time (sec) ↓	Score ↑	Consistency Score ↑	VRAM (MiB) ↓
Full Inference	1323.49	**0.8050**	**0.2083**	15584
FORA [Selvaraju et al., 2024]	867.61	0.5970	0.0799	17761
ToCa [Zou et al., 2024]	975.87	0.6118	0.1237	18450
dKV-Cache [Ma et al., 2025]	1111.50	0.4381	0.0780	16109
MANIFOLDCACHE $(T^*)\,(2,4)$	866.00	0.7048	0.1443	18119

computation (107 s vs 121 s per sample), so we compare at matched cost: $(T^*)\,(3,5)$ reaches FID 5.29 at 109 s and $(w{=}0.3)\,(2,4)$ reaches 5.51 at 101 s, both better than $(4,2)$. **Giving shallow blocks the longer stride is more costly than giving it to deep blocks**, which matches the depth ordering of Theorem 2. All baseline schedules are taken as the optimal settings reported in their respective papers. We mainly report MANIFOLDCACHE under the schedules (2,3), (2,4), and (2,5), as these consistently yielded quality generation comparable to the baselines while spanning the most informative range of speedup–fidelity trade-offs.

On Proteina (Table 1), TeaCache and TaylorSeer, both relying on continuous feature-trajectory extrapolation, reach scRMSD 8.07 and 9.11 respectively, $4{-}5\times$ worse than full inference. This is not a mild quality loss but a regime change. A likely cause is that forecasting extrapolates features without any correction toward $\mathcal{M}_c$, which accelerates drift off the manifold. In contrast, MANIFOLDCACHE at $(2,4)$ delivers scRMSD 2.04, close to the 1.93 of full inference, while still running $1.9\times$ faster. The same pattern holds for DemoDiff (Table 6), where dKV-Cache (the only competing discrete-diffusion cacher) drops the design Score by $\sim46\%$ relative to full inference.

On Lyra (Table 7 and Figure 5), MANIFOLDCACHE achieves PSNR 39.27 at 3416 s versus DeepCache's PSNR 31.42 at 3522 s and AdaCache's PSNR 34.35 at 3491 s, making it simultaneously *faster and* $\sim5{-}8$ *dB higher in PSNR*. On SV4D 2.0 (Table 8), MANIFOLDCACHE reaches PSNR 40.75 / LPIPS 0.058 in 336 s, an absolute improvement of 13.9 dB PSNR and 0.131 LPIPS over DeepCache at slightly lower latency. *Such joint gains in speed and quality over caching baselines suggest that aligning the cache schedule with the structure of the score avoids much of the usual speed–quality trade-off.* Note that for SV4D 2.0 and Lyra, we treat the full-inference run as ground truth, so it reports only wall-clock time and no quality metrics, while all baseline and MANIFOLDCACHE numbers are computed against this full-inference reference. Several baselines are both slower and less accurate than MANIFOLDCACHE: they are not designed to restrict reuse to $t \le T^*$, and our warm-up ablations show that reuse during the high-noise steps is costly; reducing their reuse to recover quality would make them slower still.

Caching incurs an unavoidable activation-storage cost proportional to cached feature volume [Lv et al., 2024]. Across the tables, MANIFOLDCACHE uses more VRAM than full inference, from 0.3% more on UniMoMo to 37% more on Med-DDPM, where it also uses more than every baseline; on the other models its memory use is comparable to that of the caching baselines.

Table 7: **Lyra** [Bahmani et al., 2026] (single-image → 3D Gaussian Splatting via camera-controlled video DiT, evaluated on multi-view synthesis of dynamic scenes): **PSNR**, **SSIM**, **LPIPS**, and runtime. MANIFOLDCACHE has the best PSNR, SSIM, and runtime; PAB has a slightly lower LPIPS but is $1.4\times$ slower.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Time (sec) ↓
Full Inference	–	–	–	5577.64
DeepCache [Ma et al., 2024]	31.42	0.9416	0.0890	3522.21
PAB [Zhao et al., 2025]	36.85	0.9750	**0.0552**	4744.55
AdaCache [Kahatapitiya et al., 2025]	34.35	0.9734	0.0910	3491.62
MANIFOLDCACHE (T^*) $(2, 3)$	**39.27**	**0.9817**	0.0564	**3416.05**

Table 8: **SV4D 2.0** [Yao et al., 2025] (multi-view video diffusion for dynamic 4D asset generation, 8 view pairs on Objaverse): **PSNR**, **SSIM**, **LPIPS**, and runtime. MANIFOLDCACHE (T^*) $(2, 3)$ is fastest *and* delivers $+13.9$ dB PSNR over DeepCache.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Time (sec) ↓	VRAM (MiB) ↓
Full Inference	–	–	–	409	16520
DeepCache [Ma et al., 2024]	26.88	0.8730	0.1891	350	17885
FrDiff [So et al., 2024]	31.94	0.9192	0.0764	347	18673
MANIFOLDCACHE (T^*) $(2, 3)$	**40.75**	**0.9909**	**0.0579**	**336**	18540

The configuration sweet spot is consistent. $(2, 4)$ at cutoff T^* is our default and the best of the reported configurations on Proteina, DiffCSP++, and Med-DDPM. Other strides expose interpretable trade-offs: $(2, 5)$ pushes deep strides further and gains marginal speed at marginal quality cost (Med-DDPM FID 4.57 vs 4.55 at $(2, 4)$); $(2, 3)$ is the most conservative setting and is the setting we use for SV4D 2.0 and Lyra. This stability across radically different ambient spaces, from $192\times192\times144$ voxel grids to motif graphs, crystal lattices, and multi-view (4D) video latents, empirically supports the universality claim of §3.1: a single closed-form schedule, derived purely from the noise schedule and block depth, governs cacheability across CMDMs. Please refer to Appendix G for detailed Limitations and Future Work.

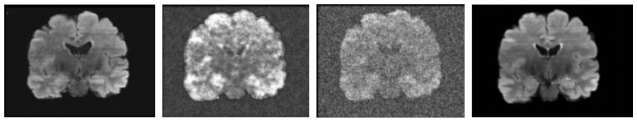

Figure 3: From left to right: MANIFOLDCACHE, FasterDiffusion, DeepCache, and Full Inference. MANIFOLDCACHE achieves visual quality most similar to Full Inference on Med-DDPM (3D Brain MRI).

6 Conclusion

We introduced MANIFOLDCACHE, the first training-free acceleration framework designed specifically for Constraint-Manifold Diffusion Models (CMDMs). By exploiting the orthogonal decomposition of the diffusion score into manifold-normal and manifold-tangential components, MANIFOLDCACHE derives a principled safe-caching threshold directly from the noise schedule together with a depth-adaptive caching policy grounded in score geometry. The resulting method requires no retraining, no calibration data, and negligible runtime overhead. Across eight diverse CMDMs spanning medical imaging, molecular generation, protein design, crystal prediction, and multi-view 4D synthesis, MANIFOLDCACHE consistently achieves superior quality–speed trade-offs and often simultaneously improves both inference time and generation fidelity over strong existing accelerators. Our results suggest that constraint geometry is not only central to generation quality, but also a fundamental signal for determining when and where diffusion computation can be safely reused.

References

Sudhanshu Agrawal, Risheek Garrepalli, Raghavv Goel, Mingu Lee, Christopher Lott, and Fatih Porikli. Spiffy: Multiplying diffusion llm acceleration via lossless speculative decoding. *arXiv e-prints*, pages arXiv–2509, 2025.

Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David B. Lindell, Zan Gojcic, Sanja Fidler, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra: Generative 3d scene reconstruction via video diffusion model self-distillation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tIVCfVnIHo.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15, 2024.

Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? A fine-grained analysis in a simple setting. In *Advances in Neural Information Processing Systems*, 2024.

Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling. *arXiv preprint arXiv:2501.05370*, 2025.

Zolnamar Dorjsembe, Hsing-Kuo Pao, Sodtavilan Odonchimed, and Furen Xiao. Conditional diffusion models for semantic 3d brain mri synthesis. *IEEE Journal of Biomedical and Health Informatics*, 28(7):4084–4093, 2024.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=d4f40zJJIS.

Herbert Federer. Curvature measures. *Transactions of the American mathematical Society*, 93(3): 418–491, 1959.

Weilun Feng, Haotong Qin, Chuanguang Yang, Zhulin An, Libo Huang, Boyu Diao, Fei Wang, Renshuai Tao, Yongjun Xu, and Michele Magno. Mpq-dm: Mixed precision quantization for extremely low bit diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16595–16603, 2025.

Thorben Frank, Winfried Ripken, Gregor Lied, Klaus Robert Muller, Oliver T. Unke, and Stefan Chmiela. Sampling 3d molecular conformers with diffusion transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tmbx9zGVWb.

Weizhi Gao, Zhichao Hou, Junqi Yin, Feiyi Wang, Linyu Peng, and Xiaorui Liu. Modulated diffusion: Accelerating generative modeling with modulated quantization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=y8hMadAgrz.

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Proteina: Scaling flow-based protein structure generative models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TVQLu34bdw.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36:13237–13249, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7362–7371, 2024.

Rui Jiao, Wenbing Huang, Yu Liu, Deli Zhao, and Yang Liu. Space group constrained crystal generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jkvZ7v4OmP.

Kumara Kahatapitiya, Haozhe Liu, Sen He, Ding Liu, Menglin Jia, Chenyang Zhang, Michael S Ryoo, and Tian Xie. Adaptive caching for faster video generation with diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15240–15252, 2025.

Xiangzhe Kong, Zishen Zhang, Ziting Zhang, Rui Jiao, Jianzhu Ma, Wenbing Huang, Kai Liu, and Yang Liu. Unimomo: Unified generative modeling of 3d molecules for de novo binder design. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KUN7A7Okb6.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Marvin Li and Sitan Chen. Critical windows: Non-asymptotic theory for feature emergence in diffusion models. In *International Conference on Machine Learning*, 2024.

Senmao Li, Taihang Hu, Joost van de Weijer, Fahad S Khan, Tao Liu, Linxuan Li, Shiqi Yang, Yaxing Wang, Ming-Ming Cheng, and Jian Yang. Faster diffusion: Rethinking the role of the encoder for diffusion model inference. *Advances in Neural Information Processing Systems*, 37:85203–85240, 2024.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17535–17545, 2023.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7353–7363, 2025a.

Gang Liu, Jie Chen, Yihan Zhu, Michael Sun, Tengfei Luo, Nitesh V Chawla, and Meng Jiang. Graph diffusion transformers are in-context molecular designers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=lJ87GN5zJc.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15853–15863, 2025b.

Joseph Liu, Joshua Geddes, Ziyu Guo, Haomiao Jiang, and Mahesh Kumar Nandwana. Smoothcache: A universal inference acceleration technique for diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3229–3238, 2025c.

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

Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K Wong. Fastercache: Training-free video diffusion model acceleration with high quality. *arXiv preprint arXiv:2410.19355*, 2024.

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*, pages 15762–15772, 2024.

Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dKV-cache: The cache for diffusion language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Gppo2JImHs.

Gabriel Raya and Luca Ambrogioni. Spontaneous symmetry breaking in generative diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024.

Kulin Shah, Sitan Chen, and Adam Klivans. Learning mixtures of Gaussians using the DDPM objective. In *Advances in Neural Information Processing Systems*, 2023.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1972–1981, 2023.

Junhyuk So, Jungwon Lee, and Eunhyeok Park. Frdiff: Feature reuse for universal training-free acceleration of diffusion models. In *European Conference on Computer Vision*, pages 328–344. Springer, 2024.

Johanna Sommer, John Rachwan, Nils Fleischmann, Stephan Günnemann, and Bertrand Charpentier. Predictive feature caching for training-free acceleration of molecular geometry generation. *arXiv preprint arXiv:2510.04646*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Jan Pawel Stanczuk, Georgios Batzolis, Teo Deveney, and Carola-Bibiane Schönlieb. Diffusion models encode the intrinsic dimension of data manifolds. In *Forty-first International Conference on Machine Learning*, 2024.

Haoxuan Wang, Yuzhang Shang, Zhihang Yuan, Junyi Wu, Junchi Yan, and Yan Yan. Quest: Low-bit diffusion model quantization via efficient selective finetuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15542–15551, 2025.

Chun-Han Yao, Yiming Xie, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d 2.0: Enhancing spatio-temporal consistency in multi-view video diffusion for high-quality 4d generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13248–13258, 2025.

Zichao Yu, Zhen Zou, Guojiang Shao, Chenwei Zhang, Shengze Xu, Jie Huang, Feng Zhao, Xiaodong Cun, and Wenyi Zhang. Ab-cache: Training-free acceleration of diffusion models via adams-bashforth cached feature reuse. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 10408–10417, 2025.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hDBrQ4DApF.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: This can be verified by the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the Appendix

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: In the appendix

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details added

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Datasets are public, settings and splits are provided, code will be released upon acceptance

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Details provided in the main paper

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: All existing methods in this category report numbers without error bars, so we followed them.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Provided in main paper

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms to the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have referenced the required papers.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: There are no new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Used for writing and editing etc.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Appendix

A Theoretical Analysis

A.1 Notation for Proofs

We work with the two-mode Gaussian mixture of Assumption (A3). Let $\mu_1, \mu_2 \in \mathcal{M}_c$ be the two mode centres with separation $\delta_c := \|\mu_1 - \mu_2\| > 0$, unit intermode axis $u := (\mu_2 - \mu_1)/\delta_c$, and midpoint $\bar{\mu} := (\mu_1 + \mu_2)/2$. Adopting centred coordinates, $\langle \mu_1, u\rangle = -\delta_c/2$ and $\langle \mu_2, u\rangle = \delta_c/2$. The within-mode covariance is $\Sigma_c^{\text{in}} := P_u^{\perp} + \sigma_u^2 P_u$, where $P_u := uu^{\top}$ and $P_u^{\perp} := \mathbf{I}_{d_c} - P_u$, so Σ_c^{in} has eigenvalue σ_u^2 along u and eigenvalue 1 in the remaining $d_c - 1$ tangential directions. The total marginal covariance of the equal-weight mixture is $\Sigma_c := P_u^{\perp} + \Lambda_c P_u$ with $\Lambda_c := \sigma_u^2 + \delta_c^2/4$, so $\operatorname{tr}\Sigma_c = (d_c - 1) + \Lambda_c$ and the eigenvalues of Σ_c are 1 (multiplicity $d_c - 1$) and Λ_c (multiplicity 1).

We use the following quantities throughout the proofs. $V_\tau^u := (1-\bar{\alpha}_\tau)+\bar{\alpha}_\tau\sigma_u^2$ is the marginal variance of $x_{\tau,u} := \langle x_\tau, u\rangle$ given either mode, arising from $x_\tau = \sqrt{\bar{\alpha}_\tau}x_0 + \sqrt{1-\bar{\alpha}_\tau}\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$. Φ denotes the standard Gaussian CDF. $M_t := \widehat{x}_0(x_t, t; c) = \mathbb{E}[x_0 \mid x_t, c]$ is the Tweedie posterior mean. $\pi_k(x_\tau) := \Pr[\text{mode } k \mid x_\tau]$ is the posterior mode probability. $\Pi_{\mathcal{M}_c}(x) \in \{\mu_1, \mu_2\}$ denotes nearest-mode projection. In centred coordinates $\|x - \mu_1\|^2 - \|x - \mu_2\|^2 = 2\delta_c\langle x, u\rangle$, so $\Pi_{\mathcal{M}_c}(x) = \mu_2$ if $\langle x, u\rangle \geq 0$ and $\Pi_{\mathcal{M}_c}(x) = \mu_1$ otherwise (ties, which have probability zero, are assigned to μ_2). $K \in \{1, 2\}$ denotes the mode from which x_0 is drawn.

A.2 Standing Assumptions

(A1) Network regularity. Each block f_ℓ is C^2 along the sampling trajectory with $\|J_\ell\|_{\text{op}} \leq K_\ell$ and $\|\nabla^2 h_\ell\|_{\text{op}} \leq M_\ell$ a.s., where $J_\ell := \nabla_x h_\ell$.

(A2) Manifold regularity. $\mathcal{M}_c$ is a closed C^2 submanifold of $\mathcal{X}$ with positive reach $\rho_c > 0$ (guaranteed by Definition 1 when g is C^2 with the submersion condition holding).

(A3) Two-mode Gaussian mixture. $p(x_0 \mid c) = \frac{1}{2}\mathcal{N}(\mu_1, \Sigma_c^{\text{in}}) + \frac{1}{2}\mathcal{N}(\mu_2, \Sigma_c^{\text{in}})$ (notation as in §A.1), with the following conditions: (i) $0 < \sigma_u \leq 1$; (ii) $\Lambda_c > 1$ (equivalently $\delta_c > 2\sqrt{1-\sigma_u^2}$), ensuring all eigenvalues of Σ_c satisfy $\lambda_i \geq 1$; (iii) $\delta_c > 0$.

(A4) ODE remainder. $\mathbb{E}\|R(\Delta t)\|^4 \leq C_R \Delta t^8$ for the Euler remainder $R(\Delta t)$ of the probability-flow ODE (the local error of an Euler step is $O(\Delta t^2)$).

(A5) Directional Lipschitz constants and depth ordering. For each block ℓ and $t \leq T^*$, there exist $L_\ell^{\perp}, L_\ell^{\parallel} > 0$ with $\|J_\ell P_x^{\perp} v\| \leq L_\ell^{\perp}\|v\|$ for $v \in N_{\Pi(x)}\mathcal{M}_c$ and $\|J_\ell P_x^{\parallel} w\| \leq L_\ell^{\parallel}\|w\|$ for $w \in T_{\Pi(x)}\mathcal{M}_c$. Under unit-gain normalisation $L_\ell^{\perp} + L_\ell^{\parallel} = 1$, set $\beta_\ell := L_\ell^{\perp}$ and $L_\ell^{\parallel} = 1 - \beta_\ell$. The depth-ordering condition $\beta_{\ell_1} < \beta_{\ell_2}$ for $\ell_1 < \ell_2$ holds throughout the safe phase.

(A5′) Transversality (for Theorem 2, two-sided bound only). There exist $\theta_* \in (0, \pi/2]$, $\zeta \in (0, \cos\theta_*]$, and $\gamma_* > 0$ such that for all non-zero $J_\ell v$, $J_\ell w$ with $v \in N_{\Pi(x)}\mathcal{M}_c$, $w \in T_{\Pi(x)}\mathcal{M}_c$: $\langle J_\ell v, J_\ell w\rangle/(\|J_\ell v\|\|J_\ell w\|) \geq -\cos\theta_* + \zeta$, so $\alpha := \cos\theta_* - \zeta \geq 0$. Additionally: $\|J_\ell v\| \geq \underline{L}_\ell^{\perp}\|v\|$ and $\|J_\ell w\| \geq \underline{L}_\ell^{\parallel}\|w\|$, with $\gamma_* := \min(\underline{L}_\ell^{\perp}/L_\ell^{\perp}, \underline{L}_\ell^{\parallel}/L_\ell^{\parallel}) > 0$. The constraint $\zeta \leq \cos\theta_*$ (i.e., $\alpha \geq 0$) is necessary for the AM-GM step in the lower bound (Step 6).

(A6) Temporal regularity. $\mathbb{E}\,\delta_\ell(t_1)^2 \leq \eta\,\mathbb{E}\,\delta_\ell(t_2)^2$ for all t_1, t_2 within any cache stride window, for some $\eta \geq 1$.

A.3 Lemma 1: Exact Posterior Variance

Lemma 1 (MMSE upper bound)**.** *For any prior $p(x_0 \mid c)$ with mean zero and covariance Σ_c having eigenvalues $\{\lambda_i\}_{i=1}^{d_c}$, and the VP forward kernel $q(x_\tau \mid x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_\tau}x_0, (1-\bar{\alpha}_\tau)\mathbf{I})$, the Tweedie estimate $M_\tau = \mathbb{E}[x_0 \mid x_\tau]$ satisfies*

$$\mathbb{E}\|M_\tau - x_0\|^2 \;\leq\; \sum_{i=1}^{d_c} \frac{(1-\bar{\alpha}_\tau)\lambda_i}{\bar{\alpha}_\tau\lambda_i + 1 - \bar{\alpha}_\tau}.$$

Proof. The nonlinear MMSE estimator $M_\tau = \mathbb{E}[x_0 \mid x_\tau]$ achieves lower mean-squared error than any linear estimator. Work in the eigenbasis of Σ_c. The linear MMSE estimator of $x_0^{(i)}$ from $x_\tau^{(i)} = \sqrt{\bar{\alpha}_\tau}x_0^{(i)} + \sqrt{1-\bar{\alpha}_\tau}\varepsilon^{(i)}$ has MSE $\operatorname{Cov}(x_0^{(i)}) - \operatorname{Cov}(x_0^{(i)}, x_\tau^{(i)})^2/\operatorname{Var}(x_\tau^{(i)}) = \lambda_i - \bar{\alpha}_\tau\lambda_i^2/(\bar{\alpha}_\tau\lambda_i + 1 - \bar{\alpha}_\tau) = (1-\bar{\alpha}_\tau)\lambda_i/(\bar{\alpha}_\tau\lambda_i + 1 - \bar{\alpha}_\tau)$, using $\operatorname{Cov}(x_0^{(i)}, x_\tau^{(i)}) = \sqrt{\bar{\alpha}_\tau}\lambda_i$ and $\operatorname{Var}(x_\tau^{(i)}) = \bar{\alpha}_\tau\lambda_i + 1 - \bar{\alpha}_\tau$. Since $\mathbb{E}\|M_\tau - x_0\|^2 \leq \mathbb{E}\|\hat{x}_{0,\text{lin}} - x_0\|^2 = \sum_i (1-\bar{\alpha}_\tau)\lambda_i/(\bar{\alpha}_\tau\lambda_i + 1 - \bar{\alpha}_\tau)$, the result follows. □

A.4 Proof of Theorem 1

Setup. Freezing the sampler at cache time τ yields output $x_0^\tau = M_\tau := \widehat{x}_0(x_\tau, \tau; c)$. Since $M_t = \mathbb{E}[x_0 \mid x_t, c]$ is a reverse-time martingale (by the tower property: $\mathbb{E}[M_s \mid x_t] = \mathbb{E}[x_0 \mid x_t] = M_t$ for $s < t$) with $M_0 = x_0$ a.s., the cache error is $\mathcal{E}(\tau) = \mathbb{E}\|M_\tau - x_0\|^2$.

Part (i): Safe regime $\bar{\alpha}_\tau \geq \frac{1}{2}$. By Lemma 1 (MMSE $\leq$ linear MMSE, valid for any prior with covariance Σ_c): $\mathcal{E}(\tau) \leq \sum_i (1-\bar{\alpha}_\tau)\lambda_i/(\bar{\alpha}_\tau\lambda_i + 1 - \bar{\alpha}_\tau)$. Since $\lambda_i \geq 1$ (A3.ii), we have $\bar{\alpha}_\tau\lambda_i + 1 - \bar{\alpha}_\tau \geq 1$, so each term is bounded by $(1-\bar{\alpha}_\tau)\lambda_i$, giving

$$\mathcal{E}(\tau) \;\leq\; (1-\bar{\alpha}_\tau)\,\mathrm{tr}\,\Sigma_c \;\leq\; \tfrac{1}{2}\,\mathrm{tr}\,\Sigma_c, \tag{7}$$

where the last step uses $1 - \bar{\alpha}_\tau \leq \frac{1}{2}$.

Part (ii): Mode-flip probability. Fix $\tau \geq 1$, so that $\bar{\alpha}_\tau \in (0,1)$. In centred coordinates ($\mu_1 = -\frac{\delta_c}{2}u$, $\mu_2 = \frac{\delta_c}{2}u$), write $x_{\tau,u} := \langle x_\tau, u\rangle$. The marginal distribution of $x_{\tau,u}$ given mode k is $\mathcal{N}(\sqrt{\bar{\alpha}_\tau}\langle\mu_k, u\rangle, V_\tau^u)$.

Step 1: Log-posterior ratio. Under (A3), Σ_c^{in} is block-diagonal with respect to u and its orthogonal complement, the component of x_0 orthogonal to u has the same distribution under both modes, and the forward noise is isotropic. Hence the two mode likelihoods differ only through $x_{\tau,u}$. Since the two mode-conditional marginals of $x_{\tau,u}$ are Gaussian with the same variance V_τ^u and means differing by $\sqrt{\bar{\alpha}_\tau}\delta_c$:

$$\log\frac{\pi_2(x_\tau)}{\pi_1(x_\tau)} \;=\; \log\frac{p(x_{\tau,u} \mid \text{mode } 2)}{p(x_{\tau,u} \mid \text{mode } 1)} \;=\; \frac{\sqrt{\bar{\alpha}_\tau}\delta_c}{V_\tau^u}\,x_{\tau,u},$$

so $\pi_2 \geq \frac{1}{2}$ if and only if $x_{\tau,u} \geq 0$.

Step 2: Tweedie flip (Lemma 2).

Lemma 2 (Tweedie mode flip)**.** *$\langle M_\tau, u\rangle \geq 0$ if and only if $x_{\tau,u} \geq 0$. Hence $\Pi_{\mathcal{M}_c}(M_\tau) = \mu_2$ if $x_{\tau,u} \geq 0$ and $\Pi_{\mathcal{M}_c}(M_\tau) = \mu_1$ otherwise.*

Proof. The Tweedie formula gives $M_\tau = \pi_1 m_1 + \pi_2 m_2$, where $m_k := \mathbb{E}[x_0 \mid x_\tau, \text{mode } k]$. By the independence noted in Step 1, $\langle m_k, u\rangle = \mathbb{E}[\langle x_0, u\rangle \mid x_{\tau,u}, \text{mode } k]$, and the Gaussian regression formula ($\mathrm{Cov}(x_{0,u}, x_{\tau,u}) = \sqrt{\bar{\alpha}_\tau}\sigma_u^2$, $\mathrm{Var}(x_{\tau,u}) = V_\tau^u$) gives

$$\langle m_k, u\rangle \;=\; \langle\mu_k, u\rangle + \frac{\sqrt{\bar{\alpha}_\tau}\sigma_u^2}{V_\tau^u}(x_{\tau,u} - \sqrt{\bar{\alpha}_\tau}\langle\mu_k, u\rangle) \;=\; \frac{\sqrt{\bar{\alpha}_\tau}\sigma_u^2\,x_{\tau,u} + (1-\bar{\alpha}_\tau)\langle\mu_k, u\rangle}{V_\tau^u}.$$

Hence $\langle m_2 - m_1, u\rangle = (1-\bar{\alpha}_\tau)\delta_c/V_\tau^u > 0$, $\langle m_1, u\rangle = [\sqrt{\bar{\alpha}_\tau}\sigma_u^2\,x_{\tau,u} - (1-\bar{\alpha}_\tau)\delta_c/2]/V_\tau^u$, and $\langle m_2, u\rangle = [\sqrt{\bar{\alpha}_\tau}\sigma_u^2\,x_{\tau,u} + (1-\bar{\alpha}_\tau)\delta_c/2]/V_\tau^u$. On $\{x_{\tau,u} \geq 0\}$, we have $\pi_2 \geq \frac{1}{2}$, so:

$$\begin{aligned}\langle M_\tau, u\rangle &= \langle m_1, u\rangle + \pi_2\langle m_2 - m_1, u\rangle \\ &\geq \frac{\sqrt{\bar{\alpha}_\tau}\sigma_u^2\,x_{\tau,u} - (1-\bar{\alpha}_\tau)\delta_c/2}{V_\tau^u} + \frac{(1-\bar{\alpha}_\tau)\delta_c}{2V_\tau^u} \;=\; \frac{\sqrt{\bar{\alpha}_\tau}\sigma_u^2\,x_{\tau,u}}{V_\tau^u} \;\geq\; 0.\end{aligned}$$

On $\{x_{\tau,u} < 0\}$, we have $\pi_1 > \frac{1}{2}$, so:

$$\begin{aligned}\langle M_\tau, u\rangle &= \langle m_2, u\rangle - \pi_1\langle m_2 - m_1, u\rangle \\ &< \frac{\sqrt{\bar{\alpha}_\tau}\sigma_u^2\,x_{\tau,u} + (1-\bar{\alpha}_\tau)\delta_c/2}{V_\tau^u} - \frac{(1-\bar{\alpha}_\tau)\delta_c}{2V_\tau^u} \;=\; \frac{\sqrt{\bar{\alpha}_\tau}\sigma_u^2\,x_{\tau,u}}{V_\tau^u} \;<\; 0.\end{aligned}$$

The last statement follows from the characterisation of $\Pi_{\mathcal{M}_c}$ in §A.1. □

Step 3: Closed form. By Lemma 2, the frozen estimate is closer to the wrong mode exactly when $x_{\tau,u}$ falls on the wrong side of zero. Given mode 1, $x_{\tau,u} \sim \mathcal{N}(-\sqrt{\bar{\alpha}_\tau}\delta_c/2,\, V_\tau^u)$, so

$$\Pr[x_{\tau,u} \geq 0 \mid K = 1] \;=\; \Phi\Big(-\frac{\sqrt{\bar{\alpha}_\tau}\,\delta_c}{2\sqrt{V_\tau^u}}\Big),$$

and by symmetry $\Pr[x_{\tau,u} < 0 \mid K = 2]$ takes the same value, which proves (3). Since $V_\tau^u = 1 - \bar{\alpha}_\tau(1-\sigma_u^2)$, the derivative of $\bar{\alpha}_\tau/V_\tau^u$ with respect to $\bar{\alpha}_\tau$ is $(V_\tau^u)^{-2} > 0$, so p_{flip} is continuous and strictly decreasing in $\bar{\alpha}_\tau$.

Safe regime ($\bar{\alpha}_\tau \geq \frac{1}{2}$). Since $\sigma_u \leq 1$ (A3.i), $V_\tau^u \leq 1$, hence

$$\sqrt{\bar{\alpha}_\tau / V_\tau^u} \;\geq\; \sqrt{\bar{\alpha}_\tau} \;\geq\; 1/\sqrt{2} \quad \text{and} \quad p_{\text{flip}}(\tau) \;\leq\; \Phi(-\delta_c \sqrt{\bar{\alpha}_\tau}/2) \;\leq\; \Phi(-\delta_c/(2\sqrt{2})).$$

Unsafe regime ($\bar{\alpha}_\tau < \frac{1}{2}$). Here $V_\tau^u - \bar{\alpha}_\tau = 1 - 2\bar{\alpha}_\tau + \bar{\alpha}_\tau \sigma_u^2 > 0$, so $\sqrt{\bar{\alpha}_\tau / V_\tau^u} < 1$ and $p_{\text{flip}}(\tau) > \Phi(-\delta_c/2) = p_{\text{mix}}$. This bound does not depend on $\bar{\alpha}_\tau$, so it does not vanish as $\bar{\alpha}_\tau \uparrow \frac{1}{2}$.

Part (iii): Role of T^*. Since Φ is strictly increasing, $p_{\text{flip}}(\tau) \geq p_{\text{mix}}$ if and only if $\bar{\alpha}_\tau \leq V_\tau^u$, that is, $\bar{\alpha}_\tau(2 - \sigma_u^2) \leq 1$, or $\bar{\alpha}_\tau \leq 1/(2 - \sigma_u^2)$. For $\sigma_u \in (0, 1]$ we have $1/(2-\sigma_u^2) > \frac{1}{2}$, and $1/(2-\sigma_u^2) \to \frac{1}{2}$ as $\sigma_u \to 0$. Hence the condition holds for every $\sigma_u \in (0, 1]$ if and only if $\bar{\alpha}_\tau \leq \frac{1}{2}$. □

A.5 Proof of Theorem 2

Step 1: Linearise block drift. By Taylor's theorem and (A1): $h_\ell(x_{t'}, t, c) - h_\ell(x_t, t, c) = J_\ell \Delta x + E_\ell$, with $\Delta x := x_{t'} - x_t$ and $\|E_\ell\| \leq \frac{M_\ell}{2} \|\Delta x\|^2 = O(\Delta t^2)$. Squaring, taking expectations, and bounding the cross term via Cauchy–Schwarz (which gives $O(\Delta t^3)$):

$$\mathbb{E}\, \delta_\ell(t)^2 = \mathbb{E}\|J_\ell \Delta x\|^2 + O(\Delta t^3). \tag{8}$$

Step 2: Decompose the Euler step. The probability-flow ODE advances as $x_{t'} = x_t + \Delta t\, A_t s_t(x_t; c) + R(\Delta t)$, so $\Delta x = \Delta t A_t s_t + R(\Delta t)$. Using $s_t = s_t^\perp + s_t^\parallel$ and the projectors: $P_x^\perp \Delta x = \Delta t A_t s_t^\perp + P_x^\perp R$ and $P_x^\parallel \Delta x = \Delta t A_t s_t^\parallel + P_x^\parallel R$. With $\Theta_t := A_t^2 \mathbb{E}\|s_t^\parallel\|^2 > 0$ and $\kappa_t := \mathbb{E}\|s_t^\perp\|^2 / \mathbb{E}\|s_t^\parallel\|^2$, Assumption (A4) gives $\mathbb{E}\|P_x^\perp R\|^2, \mathbb{E}\|P_x^\parallel R\|^2 = O(\Delta t^4)$, so:

$$\mathbb{E}\|P_x^\perp \Delta x\|^2 = \Delta t^2 \Theta_t \kappa_t + O(\Delta t^3), \quad \mathbb{E}\|P_x^\parallel \Delta x\|^2 = \Delta t^2 \Theta_t + O(\Delta t^3). \tag{9}$$

Step 3: Upper bound on drift. By (A5) and the triangle inequality: $\|J_\ell \Delta x\| \leq L_\ell^\perp \|P_x^\perp \Delta x\| + L_\ell^\parallel \|P_x^\parallel \Delta x\|$. Using $(a+b)^2 \leq 2(a^2 + b^2)$, taking expectations, and substituting (9) into (8):

$$\mathbb{E}\, \delta_\ell(t)^2 \;\leq\; \overline{B}_\ell\, \Delta t^2 \Theta_t + O(\Delta t^3), \quad \overline{B}_\ell := 2[(L_\ell^\perp)^2 \kappa_t + (L_\ell^\parallel)^2] = 2B(\beta_\ell; \kappa_t). \tag{10}$$

Step 4: Certified strides. Let $\overline{D}_\ell := \big(\overline{B}_\ell \Delta t^2 \Theta_t + O(\Delta t^3)\big)^{1/2} = \Delta t \sqrt{\overline{B}_\ell \Theta_t}\,\big(1 + O(\Delta t)\big)$ be the square root of the right-hand side of (10). For every step $t - k\Delta t$ in a cache window starting at t, Jensen's inequality and (A6) give $\mathbb{E}\, \delta_\ell(t - k\Delta t) \leq \big(\mathbb{E}\, \delta_\ell(t - k\Delta t)^2\big)^{1/2} \leq \sqrt{\eta}\, \overline{D}_\ell$. Hence every stride S with $(S-1)\sqrt{\eta}\, \overline{D}_\ell \leq \varepsilon$ is ε-cacheable, and the certified stride $\widehat{S}_\ell := 1 + \varepsilon/(\sqrt{\eta}\, \overline{D}_\ell)$ satisfies $\lfloor \widehat{S}_\ell \rfloor \leq S_\ell^\star$, where $S_\ell^\star$ is the largest ε-cacheable stride.

Step 5: Depth-monotone comparison. For $\ell_1 < \ell_2$ with $\beta_{\ell_1} < \beta_{\ell_2}$ (A5):

$$\frac{\widehat{S}_{\ell_2} - 1}{\widehat{S}_{\ell_1} - 1} = \frac{\overline{D}_{\ell_1}}{\overline{D}_{\ell_2}} = \sqrt{\frac{B(\beta_{\ell_1}; \kappa_t)}{B(\beta_{\ell_2}; \kappa_t)}}\,(1 + O(\Delta t)).$$

Its leading factor exceeds one iff $B(\beta_{\ell_1}; \kappa_t) > B(\beta_{\ell_2}; \kappa_t)$, i.e., $(\beta_{\ell_1}^2 - \beta_{\ell_2}^2)\kappa_t > (1-\beta_{\ell_2})^2 - (1-\beta_{\ell_1})^2$. Dividing both sides by $(\beta_{\ell_2}^2 - \beta_{\ell_1}^2) > 0$ and negating: $\kappa_t < \kappa^* := [(1 - \beta_{\ell_1}) + (1 - \beta_{\ell_2})]/(\beta_{\ell_1} + \beta_{\ell_2})$, which is (5). This establishes the envelope comparison and the certified-stride comparison of Theorem 2. □ (Theorem 2a)

Step 6: Actual drift under transversality. Write $v := P_x^\perp \Delta x$ and $w := P_x^\parallel \Delta x$. From (A5′) with $\alpha := \cos\theta_* - \zeta \geq 0$: $2\langle J_\ell v, J_\ell w \rangle \geq -2\alpha \|J_\ell v\| \|J_\ell w\| \geq -\alpha(\|J_\ell v\|^2 + \|J_\ell w\|^2)$, where the last step applies AM-GM and uses $\alpha \geq 0$. Hence:

$$\|J_\ell \Delta x\|^2 = \|J_\ell v\|^2 + \|J_\ell w\|^2 + 2\langle J_\ell v, J_\ell w \rangle \;\geq\; (1 - \alpha)(\|J_\ell v\|^2 + \|J_\ell w\|^2).$$

Applying the lower-Lipschitz bounds ($\|J_\ell v\| \geq \underline{L}_\ell^\perp \|v\| \geq \gamma_* \beta_\ell \|v\|$ and similarly for w):

$$\mathbb{E}\, \delta_\ell^2(t) \;\geq\; \underline{B}_\ell\, \Delta t^2 \Theta_t - O(\Delta t^3), \quad \underline{B}_\ell := (1 - \alpha)\gamma_*^2[\beta_\ell^2 \kappa_t + (1 - \beta_\ell)^2] = \tfrac{(1-\alpha)\gamma_*^2}{2} \overline{B}_\ell. \tag{11}$$

Combining (10) for ℓ_2 with (11) for ℓ_1,

$$\mathbb{E}\, \delta_{\ell_2}(t)^2 \leq 2B(\beta_{\ell_2}; \kappa_t)\, \Delta t^2 \Theta_t + O(\Delta t^3),$$
$$\mathbb{E}\, \delta_{\ell_1}(t)^2 \geq (1 - \alpha)\gamma_*^2 B(\beta_{\ell_1}; \kappa_t)\, \Delta t^2 \Theta_t - O(\Delta t^3).$$

If $B(\beta_{\ell_1}; \kappa_t)/B(\beta_{\ell_2}; \kappa_t) > 2/\big((1 - \alpha)\gamma_*^2\big)$, the leading term of the first bound is strictly smaller than that of the second, so $\mathbb{E}\, \delta_{\ell_2}(t)^2 < \mathbb{E}\, \delta_{\ell_1}(t)^2$ for small enough Δt, completing Theorem 2b. □

A.6 Discussion

Theorem 1 is proved under a two-mode Gaussian mixture (A3), the simplest setting in which the mode-flip probability can be computed in closed form. We use it as a simplified model: the data distributions of the evaluated models are multimodal and non-Gaussian, and we do not claim that they satisfy (A3). Theorem 1 is related to analyses of critical noise levels at which the modes of a mixture separate during sampling [Raya and Ambrogioni, 2023, Biroli et al., 2024, Li and Chen, 2024], and more broadly to theoretical studies of diffusion models on simple mixture distributions [Shah et al., 2023, Chidambaram et al., 2024]. Two parts of the analysis do not depend on (A3): the MMSE bound of Lemma 1 holds for any prior with covariance Σ_c, and Theorem 2 does not use (A3). Theorem 1 also does not use the smoothness or the reach of $\mathcal{M}_c$; only Theorem 2 needs the tubular neighbourhood of (A2), and the runtime schedule never evaluates g. Assumption (A3.ii) ($\lambda_i \geq 1$) is a coordinate normalisation used only for the simplified bound $\frac{1}{2}\operatorname{tr}\Sigma_c$.

Exact versus learned score. Both theorems are stated for the exact score. For a learned score $\hat{s}_\tau$, the learned Tweedie estimate $\hat{x}_0^\tau := (x_\tau + (1-\bar{\alpha}_\tau)\hat{s}_\tau(x_\tau))/\sqrt{\bar{\alpha}_\tau}$ satisfies $\hat{x}_0^\tau - x_0^\tau = \frac{1-\bar{\alpha}_\tau}{\sqrt{\bar{\alpha}_\tau}}(\hat{s}_\tau - s_\tau)$, so by the triangle inequality $\left(\mathbb{E}\|\hat{x}_0^\tau - x_0\|^2\right)^{1/2} \leq \mathcal{E}(\tau)^{1/2} + \frac{1-\bar{\alpha}_\tau}{\sqrt{\bar{\alpha}_\tau}}\left(\mathbb{E}\|\hat{s}_\tau - s_\tau\|^2\right)^{1/2}$. For $\tau \leq T^*$ the factor $(1-\bar{\alpha}_\tau)/\sqrt{\bar{\alpha}_\tau}$ is at most $1/\sqrt{2}$, so score errors are damped rather than amplified in the regime where we cache.

Regime condition of Theorem 2. Theorem 2 is conditional on the depth ordering (A5) and on $\kappa_t < \kappa^*$ (5); the latter is a regime condition with an explicit form that we assume rather than derive from the noise schedule.

Codimension. The safe-side bound scales with $\operatorname{tr}\Sigma_c$, the total variance along the $d_c = D - k$ tangent directions, so for a fixed variance per direction a higher codimension k gives a smaller bound. In practice, outputs with many constraints, such as multi-view video, are also less tolerant of small errors: a sub-pixel misalignment between views already lowers PSNR and raises LPIPS. We believe this is why the conservative setting $(2, 3)$ worked best for SV4D 2.0.

Other samplers. Extensions to flow-matching models require redefining T^* via the corresponding SNR parameterisation; for stochastic samplers, injected noise adds $O(\sqrt{1-\bar{\alpha}_t})$ per cached step and is left to future work.

B CMDM Membership of the Evaluated Models

Table 9 lists, for each evaluated model, the constraint encoded by $g(x; c)$ and how the model keeps its samples on $\mathcal{M}_c$: statistically (learned from data), structurally (built into the parameterisation or architecture), or through learned geometry. As discussed in §3.1, this mechanism does not affect how MANIFOLDCACHE is applied.

Table 9: Constraint encoded by $g(x; c)$ and how each evaluated model keeps its samples on $\mathcal{M}_c$.

Model	**Constraint encoded by $g(x; c)$**	**How it is maintained**
Med-DDPM	anatomical intensity and continuity	statistical
DiffCSP++	space-group and Wyckoff symmetry	structural
Proteina	Ramachandran validity and steric-clash avoidance	learned geometry
SV4D 2.0, Lyra	multi-view photometric consistency	statistical
DiT-MC	bond-geometry validity	learned geometry
UniMoMo	$\mathrm{E}(3)$-equivariant block-graph validity	structural
DemoDiff	discrete motif-graph validity (one category per token)	structural

C Quality–Speed Trade-off on All Eight Models

Figure 4 shows the quality–speed trade-off for all eight models, complementing the Med-DDPM view in Figure 1.

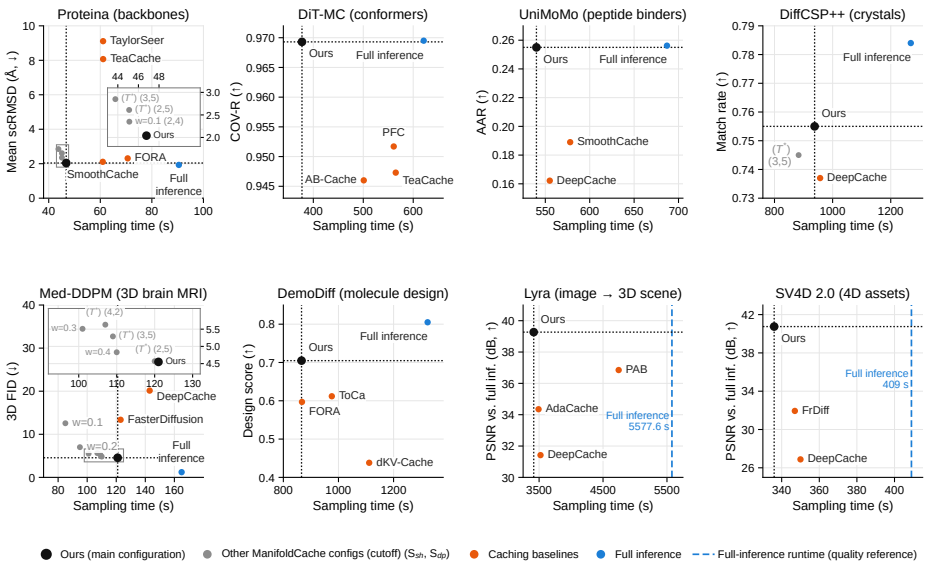

Figure 4: Quality–speed trade-off on all eight CMDMs. Each panel plots sampling time against the model's primary quality metric: mean scRMSD (Proteina), COV-R (DiT-MC), AAR (UniMoMo), Match Rate (DiffCSP++), FID (Med-DDPM), design Score (DemoDiff), and PSNR (Lyra, SV4D 2.0); the axis labels show whether higher or lower is better. "Ours" is the main MANIFOLDCACHE configuration in each table, the dotted lines pass through it, and grey points are the other MANIFOLDCACHE configurations and ablations. For Lyra and SV4D 2.0, quality is measured against full inference, so full inference appears only as a dashed line at its runtime. All values are taken from Tables 1–8.

D Computational Cost

Table 10 reports inference FLOPs for full inference, for the caching baseline with the fewest FLOPs, and for MANIFOLDCACHE. MANIFOLDCACHE needs the fewest FLOPs on five of the six models, up to 18.4% fewer than the best baseline (DemoDiff); on SV4D 2.0 it needs 1.6% more.

Table 10: Inference FLOPs (lower is better). "Best baseline" is the caching baseline with the fewest FLOPs for each model.

Model	Full Inference	Best baseline	MANIFOLDCACHE
Med-DDPM	1.06×10^{17}	7.01×10^{16}	$\mathbf{6.64 \times 10^{16}}$
Proteina	7.20×10^{14}	6.06×10^{14}	$\mathbf{5.80 \times 10^{14}}$
UniMoMo	3.24×10^{14}	2.04×10^{14}	$\mathbf{2.03 \times 10^{14}}$
SV4D 2.0	2.137×10^{15}	$\mathbf{1.945 \times 10^{15}}$	1.977×10^{15}
DiT-MC	1.81×10^{14}	1.65×10^{14}	$\mathbf{1.46 \times 10^{14}}$
DemoDiff	1.52×10^{16}	1.241×10^{16}	$\mathbf{1.013 \times 10^{16}}$

Table 11 reports the number of function evaluations (NFE) on two models from different domains, Med-DDPM (3D medical imaging) and Proteina (protein design). MANIFOLDCACHE needs the fewest NFE on both models: 142 instead of 250 on Med-DDPM (188 for the best baseline) and 164 instead of 200 on Proteina (171 for the best baseline).

Table 11: Number of function evaluations (NFE; lower is better). "Best baseline" is the caching baseline with the fewest NFE for each model.

Model	Full Inference	Best baseline	MANIFOLDCACHE
Med-DDPM	250	188	**142**
Proteina	200	171	**164**



Figure 5: From left to right: MANIFOLDCACHE, PAB, and AdaCache. It can be seen that MANIFOLDCACHE achieves the best visual quality on Lyra.

E Comparison with Speculative Sampling

Speculative methods and MANIFOLDCACHE reduce cost in different ways. Speculative methods reduce the number of expensive model calls: a cheap proposal drafts future states, and the target model then accepts or rejects them, which keeps the samples exactly distributed as those of the target model [De Bortoli et al., 2025, Agrawal et al., 2025]. MANIFOLDCACHE has no draft model, verifier, or rejection step. It keeps every denoising step and lowers the cost of each one by reusing block outputs according to a fixed schedule that depends only on the noise level and the block depth.

This leads to three practical differences. First, speculative methods need either a separate draft model, which is costly to obtain for 3D MRI or protein generation, or additional machinery. The training-free drafts of De Bortoli et al. [2025] require a stochastic sampler (the authors note that their method does not apply directly to deterministic samplers), and Spiffy [Agrawal et al., 2025] calibrates its draft graphs offline for discrete diffusion language models. De Bortoli et al. [2025] also note that, unlike large language models, diffusion models are not necessarily memory-bound, so parallel verification may bring smaller wall-clock gains. MANIFOLDCACHE needs no calibration and runs on all eight backbones. Second, the guarantees differ: speculative sampling preserves the output distribution of the target model, whereas MANIFOLDCACHE bounds the error of reuse and the probability of landing in the wrong mode (Theorem 1). Third, the two act on different parts of the trajectory: the analysis of De Bortoli et al. [2025] indicates that the acceptance rate is high at the beginning of denoising, which is the high-noise phase in which Theorem 1 advises against caching. In principle the two could therefore be combined, with speculation reducing the number of steps and caching reducing the cost of each remaining step; we leave this to future work.

Table 12 compares both speculative methods with MANIFOLDCACHE on Proteina. Neither method was designed for continuous protein backbone generation, so both had to be adapted. In our adaptations, drafts are verified token by token, which captures the continuous dynamics poorly; drafts were therefore often rejected and rolled back, both runs were slower than full inference, and the resulting scRMSD was much higher. These numbers reflect our adaptations rather than the performance of either method in the setting it was designed for.

Table 12: Speculative sampling and MANIFOLDCACHE on Proteina. Time and scRMSD for full inference and MANIFOLDCACHE are taken from Table 1.

Method	**Time (sec)** $\downarrow$	**Mean scRMSD** $\downarrow$	**FLOPs** $\downarrow$
Full Inference	90.47	1.93	7.20×10^{14}
Speculative sampling [De Bortoli et al., 2025]	281	16.2	2.8×10^{15}
Spiffy [Agrawal et al., 2025]	377	18.2	3.8×10^{15}
MANIFOLDCACHE $(T^*)\,(2, 4)$	**46.78**	**2.04**	5.80×10^{14}

F Qualitative Results

Figure 5 shows that MANIFOLDCACHE achieves the best visual quality on Lyra, particularly in preserving fine details.

G Limitations and Future Work

Despite strong empirical and theoretical performance, MANIFOLDCACHE has several important limitations that motivate future work. First, our theory relies on smooth closed manifold assumptions

with well-defined tangent and normal decompositions. While these assumptions cover a broad range of scientific diffusion models, some structured generation problems involve non-smooth, discrete, combinatorial, or dynamically changing constraint sets where the local geometric decomposition may not exist or may vary discontinuously across the trajectory. **Extending manifold-aware caching to hybrid discrete–continuous diffusion processes, topology-changing manifolds, or non-Euclidean latent spaces remains an open direction**. Second, the theoretical guarantees are derived under simplified regularity and mixture assumptions designed to make the analysis tractable. Real CMDMs exhibit highly multimodal, non-Gaussian, and hierarchical distributions whose geometry may evolve across denoising stages. Although the ablations support the choice of T^*, **deriving sharper guarantees under more realistic score models, heavy-tailed distributions, or learned manifold geometries would strengthen the theoretical foundations of geometry-aware acceleration**. In addition, the depth ordering of Theorem 2 relies on the regime condition $\kappa_t < \kappa^*$, which we assume rather than derive. Third, the current schedule is analytically fixed from the noise schedule and block depth and does not adapt online to individual samples, prompts, or trajectory difficulty. Certain trajectories may approach the manifold substantially earlier or later than predicted by the global schedule, especially in highly multimodal settings. **Future work could investigate sample-adaptive or uncertainty-aware caching policies that dynamically estimate manifold proximity, score curvature, or feature drift during inference while retaining the low-overhead nature of the current framework**. Fourth, although MANIFOLDCACHE is backbone-agnostic, the present implementation focuses on inference-time feature reuse and does not jointly optimize with other acceleration mechanisms such as learned schedulers, consistency distillation, speculative denoising, low-rank adaptation, or mixed-precision execution. **Combining geometry-aware caching with solver acceleration, quantization, or token sparsification could potentially yield multiplicative speedups beyond those observed here**. Fifth, our experiments focus on single-device inference on existing pretrained checkpoints. Large-scale distributed generation systems introduce additional communication and synchronization bottlenecks that may alter the trade-off between recomputation and cache reuse. **Investigating geometry-aware caching under tensor parallelism, pipeline parallelism, or streaming generation settings is an important systems-level extension**. Finally, our work studies geometry primarily as a mechanism for safe acceleration. **A broader future direction is to treat manifold structure as a unifying principle connecting controllability, robustness, interpretability, and efficiency in scientific generative models**. We believe CMDMs provide a promising foundation for studying these properties jointly, particularly in emerging multimodal settings combining physical simulation, language conditioning, molecular reasoning, and embodied 3D generation.