Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

DMDA: Density Matrix Decomposition for Training-Free Diffusion Acceleration

Do-Geun Kim

kimvet@q-bit.ai.kr

Q-bit Corporation

Research Article

Keywords: diffusion models, inference acceleration, density matrix, eigenmode sparsity, architecture-agnostic, training-free optimization

Posted Date: April 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9300720/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

DMDA: Density Matrix Decomposition for Training-Free Diffusion Acceleration

Do-Geun Kim[1]

[1]Q-Bit Corporation, S. Korea

*Co-corresponding author:

Do-Geun Kim, D.V.M., Ph.D.

Q-Bit Corporation

Email: kimvet@q-bit.ai.kr

Abstract

We present the Density Matrix Diffusion Accelerator (DMDA), a training-free inference acceleration framework for sequential diffusion generation. The core insight is that density matrix eigendecomposition of diffusion latent sequences reveals extreme spectral concentration — a single shared eigenmode captures the majority of total latent energy — enabling selective computation in which the shared component is processed fully only once, while per-frame perturbations are handled via lightweight residual diffusion.

We validate DMDA on two architecturally distinct models on a Mac Studio M2 Ultra (MPS backend). On **Stable Diffusion 1.5** (860M parameters, 512×512), DMDA achieves **9.4× step reduction** (1,500 → 160 steps, 35.25s wall-clock) with dominant eigenvalue $\lambda_1 = 0.8576$ (85.76% energy). On **SDXL** (2.6B parameters, 1024×1024, dual text encoder), DMDA achieves **5.1× step reduction** (750 → 146 steps, 151.1s wall-clock) with $\lambda_1 = 0.7093$ (70.93% energy). In both cases, generated frames exhibit visual quality equivalent to standard outputs with no perceptible degradation.

DMDA is training-free, architecture-agnostic, and requires no modification to model weights or architecture. It operates on a different computational axis from existing methods — reducing inter-frame redundancy rather than per-frame step count — and composes multiplicatively with distillation-based approaches.

Keywords: diffusion models, inference acceleration, density matrix, eigenmode sparsity, architecture-agnostic, training-free optimization

1. Introduction

Modern diffusion models achieve remarkable visual quality at prohibitive computational cost. Generating a sequence of N frames at S denoising steps each requires N × S forward passes through the denoising network. For a 30-frame sequence at 50 steps on Stable Diffusion 1.5, this produces 1,500 forward passes; on SDXL at 1024×1024 resolution, each forward pass is approximately 3× more expensive, making sequential generation impractical for real-time applications.

Existing acceleration approaches divide into two families: (1) **step reduction methods** (DDIM [Song et al., 2020], DPM-Solver [Lu et al., 2022], consistency distillation [Song et al., 2023]) that modify the sampling trajectory, reducing steps per image but not exploiting inter-frame redundancy; and (2) **attention optimization methods** (FlashAttention, sparse attention) that reduce per-step memory and compute but not step count. A third approach, **feature-level caching** (DeepCache [Ma et al., 2023]), empirically reuses high-level U-Net features across steps, achieving ~2.3× speedup but operating on a heuristic fixed schedule without theoretical quality guarantees.

We propose a fundamentally different perspective rooted in quantum information theory. In quantum mechanics, density matrices $\rho = \Sigma_i\, p_i\, |\psi_i\rangle\langle\psi_i|$ compactly represent mixed states, and their eigenvalue decomposition reveals the effective dimensionality of the state space. Applying this formalism to sequential diffusion latent representations, we demonstrate empirically across two architecturally distinct models that the dominant shared eigenmode captures 71–86% of total signal energy — an **eigenmode sparsity property** that serves as the mathematical foundation of DMDA.

Critically, DMDA operates on a **different computational axis** from existing methods. Step reduction methods reduce the cost per frame; DMDA reduces the number of frames requiring full computation. These are orthogonal optimizations: combining DMDA with distillation yields multiplicative speedup.

1.1 Contributions

1. **DMDA framework:** training-free sequential diffusion acceleration validated on two architecturally distinct models — SD 1.5 (9.4× step reduction) and SDXL (5.1× step reduction) — with equivalent visual quality and no model modification.

2. **Cross-architecture eigenmode sparsity:** empirical demonstration that eigenmode concentration is a general property of diffusion latent spaces, not specific to a single model. SD 1.5: $\lambda_1 = 0.8576$ (85.76% energy); SDXL: $\lambda_1 = 0.7093$ (70.93% energy).

3. **Adaptive step allocation:** TemporalCoherenceTracker dynamically allocates residual steps per frame based on Frobenius norm complexity, yielding actual speedup exceeding theoretical prediction on both models.

4. **Architecture-agnostic design:** validated across single text encoder (SD 1.5) and dual text encoder with added conditioning (SDXL) architectures on Apple Silicon MPS, demonstrating drop-in applicability without code modification to the underlying model.

2. Related Work

2.1 Diffusion Model Acceleration

DDIM (Song et al., 2020) formulated the reverse diffusion process as a deterministic ODE, enabling 10–50× step reduction per image. DPM-Solver (Lu et al., 2022) improved efficiency via high-order ODE solvers. Consistency models (Song et al., 2023) enable single-step generation through distillation but require expensive retraining — a fundamental limitation absent from DMDA. **All these methods operate in timestep space and do not exploit inter-frame redundancy.**

DeepCache (Ma et al., 2023) caches high-level U-Net features across diffusion steps, achieving ~2.3× speedup by reusing intermediate representations. While conceptually related to DMDA in exploiting temporal redundancy, DeepCache differs fundamentally: it operates on heuristic fixed-period caching within a single generation, is architecturally bound to U-Net, and provides no theoretical guarantee on quality preservation. DMDA operates across frames in a sequence, is architecture-agnostic, and provides eigenvalue-based quality bounds.

2.2 Density Matrices in Machine Learning

Density matrices have been applied to quantum-inspired machine learning (Schuld & Petruccione, 2021), natural language processing (Wiebe et al., 2019), and graph neural networks. ManifoldFormer (Fu et al., 2025) applied Riemannian manifold constraints to EEG analysis. To our knowledge, **DMDA is the first application of density matrix decomposition to diffusion process acceleration.**

2.3 Sequential Diffusion Generation

Current approaches to multi-frame diffusion generation — whether image-based (generating frames independently) or video-based (AnimateDiff [Guo et al., 2023], SVD, HunyuanVideo [Sun et al., 2024]) — share a common computational structure: iterative denoising applied to each frame or frame group. The fundamental cost scales as $O(N \times S)$. DMDA reduces this to $O(S + N \times s)$ where $s \ll S$, providing asymptotic speedup approaching S/s as N grows.

3. Density Matrix Diffusion Accelerator (DMDA)

3.1 Density Matrix Construction

Let $\{x^{(1)}, \ldots, x^{(N)}\}$ denote N latent representations in $\mathbb{R}^{(C\times H\times W)}$. We flatten each to a vector $\psi_i \in \mathbb{R}^D$ where $D = C \times H \times W$, normalize, and construct the density matrix:

$$\rho_{ij} = \frac{\langle \psi_i | \psi_j \rangle}{Tr(\rho)}$$

where the trace normalization ensures $Tr(\rho) = 1$. This $N \times N$ matrix encodes the full correlation structure of the latent sequence. Construction cost is $O(N^2 \times D)$ — negligible compared to denoising (measured: 0.06–0.07s for N=30).

3.2 Eigenmode Decomposition

Eigendecomposition of ρ yields:

$$\rho = \sum_{k=1}^{K} \lambda_k \, |\varphi_k\rangle\langle\varphi_k|, \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_K$$

The **eigenmode sparsity property** — empirically confirmed across both SD 1.5 and SDXL in Section 4 — states that a single dominant eigenmode captures the majority of total energy: $\lambda_1 = 0.8576$ (SD 1.5), $\lambda_1 = 0.7093$ (SDXL). In both cases, the spectral gap ratio λ_1/λ_2 exceeds 130:1, indicating extreme concentration.

3.3 Shared-Residual Decomposition

The eigenmode decomposition directly yields the DMDA computation schedule:

$$\rho = \rho_{\text{shared}} + \Delta\rho(t)$$

- **ρ_shared:** Eigenmodes where $\lambda_k > \tau \times \lambda_{max}$ ($\tau = 0.85$). Extracted as a weighted combination of input latents using shared eigenvector participation weights. Receives **full denoising** (S steps, executed once).
- **Δρ(t):** Remaining eigenmodes representing per-frame variations. Computed as residuals: $\Delta\rho(t) = x^{(t)} - \rho_{shared}$. Receives **lightweight residual denoising** (s steps per frame, adaptively allocated, where $s \ll S$).

3.4 Adaptive Step Allocation

Rather than applying a fixed number of residual steps per frame, the TemporalCoherenceTracker estimates per-frame complexity:

$$\text{complexity}(t) = \frac{||\Delta\rho(t)||_F}{\max_t ||\Delta\rho(t)||_F}$$

Steps are allocated proportionally within [s_min, s_max]. In practice, this adaptive allocation reduces average residual steps below the configured maximum: SD 1.5 achieves 3.7/7 steps (47% reduction); SDXL achieves 4.0/5 steps (20% reduction).

3.5 Frame Reconstruction

Final frames are reconstructed as:

$$\text{frame}(t) = \text{denoise}_{\text{full}}(\rho_{\text{shared}}) + \alpha \times \text{denoise}_{\text{light}}(\Delta\rho(t))$$

where α is the reconstruction weight controlling the contribution of per-frame residuals.

3.6 Speedup Analysis

The step count formula:

$$\text{Steps}_{\text{DMDA}} = S + \bar{s} \times N$$

$$\text{Speedup} = \frac{S \times N}{S + \bar{s} \times N}$$

The asymptotic speedup as $N \to \infty$ is $S/\bar{s}$. This means speedup **increases with sequence length** — longer sequences benefit more from DMDA because the fixed shared denoising cost is amortized over more frames.

4. Experiments

4.1 Experimental Setup

Models: - **Stable Diffusion 1.5** (stable-diffusion-v1-5/stable-diffusion-v1-5): 860M parameters, single CLIP text encoder, 512×512 resolution, DDIM scheduler. - **SDXL 1.0** (stabilityai/stable-diffusion-xl-base-1.0): 2.6B parameters, dual text encoder (CLIP-L + OpenCLIP-G), added conditioning (text_embeds, time_ids), 1024×1024 resolution, DDIM scheduler.

Hardware: Mac Studio Ultra, 512GB unified memory, Apple MPS backend. All computations in float32.

DMDA Configuration:

Parameter	SD 1.5	SDXL
num_frames	30	30
full_steps	50	25
residual_steps (max)	7	5
coherence_threshold	0.85	0.85
interference_weight	0.15	0.30
resolution	512×512	1024×1024
guidance_scale (shared)	7.5	7.5
guidance_scale (residual)	3.75	3.75

Prompt: “a Japanese garden with cherry blossoms falling, gentle breeze”

Note on pipeline: Both implementations generate 30 sequential frames using each model’s image generation U-Net with temporally correlated initial noise. This validates the density matrix decomposition principle on real diffusion latents across architecturally distinct models. Extension to temporal-attention video models (AnimateDiff, SVD) is in preparation.

4.2 Results: Stable Diffusion 1.5

Eigenmode decomposition:

Eigenvalue	Value	Cumulative Energy
λ_1	0.8576	85.76%
λ_2	0.0053	86.29%
λ_3	0.0052	86.81%
λ_4	0.0052	87.33%
λ_5	0.0051	87.84%

1 shared mode identified; 7 interference modes retained. Spectral gap ratio $\lambda_1/\lambda_2 = 161.8$.

Performance:

Metric	Value
Standard steps	1,500 (30 × 50)
DMDA steps	160 (50 + 30 × 3.7)
Step speedup	**9.4×**
Wall-clock	**35.25s**

Timing breakdown:

Phase	Time (s)	%
Noise generation	0.04	0.1%
Decomposition	0.06	0.2%
Shared diffusion	9.85	27.9%
Residual diffusion	20.88	59.3%
VAE decode	4.43	12.6%
Total	**35.25**	

4.3 Results: SDXL

Eigenmode decomposition:

Eigenvalue	Value	Energy
λ_1	0.7093	70.93%
λ_2–λ_8	~0.005–0.04	29.07% (combined)

1 shared mode identified; 7 interference modes retained.

Performance:

Metric	Value
Standard steps	750 (30 × 25)
DMDA steps	146 (25 + 30 × 4.0)
Step speedup	**5.1×**
Wall-clock	**151.1s**

Timing breakdown:

Phase	Time (s)	%
Noise generation	0.04	0.0%
Decomposition	0.07	0.0%
Shared diffusion	24.11	16.0%
Residual diffusion	110.71	73.3%
VAE decode	16.15	10.7%

Phase	Time (s)	%
Total	**151.1**	

4.4 Cross-Architecture Comparison

	SD 1.5	SDXL	Ratio (SDXL/SD)
Parameters	860M	2.6B	3.0×
Resolution	512×512	1024×1024	4.0× (pixels)
λ_1	0.8576	0.7093	0.83×
Step speedup	9.4×	5.1×	—
Per-step cost	~0.2s	~0.9s	4.5×
Standard time (est.)	~5 min	~11 min	2.2×
DMDA time	35.25s	151.1s	4.3×
Absolute time saved	**~4.4 min**	**~9 min**	**2.0×**
Visual quality	Equivalent	Equivalent	—

Key observations:

1. **Eigenmode sparsity is architecture-general.** Both models exhibit dominant single-mode concentration ($\lambda_1 > 0.70$), confirming that spectral concentration in diffusion latent space is not an artifact of a specific architecture but a structural property of the diffusion process itself.

2. **Larger models show lower λ_1 but greater absolute savings.** SDXL's richer latent space distributes energy more broadly ($\lambda_1 = 0.71$ vs 0.86), yielding lower step-speedup. However, because per-step cost is 4.5× higher, the absolute time saved is approximately 2× greater (9 min vs 4.4 min).

3. **Architecture-agnostic operation confirmed.** DMDA applied identically to SD 1.5 (single text encoder) and SDXL (dual text encoder + added conditioning) without any model-specific code modification, validating the drop-in design claim.

4. **Visual quality preserved in both cases.** Generated frames from both models are visually equivalent to their respective standard outputs, confirming that density matrix decomposition preserves the quality-critical shared structure regardless of model capacity.

4.5 Speedup Scaling with Sequence Length

Using measured average residual steps ($\bar{s}$) from each model:

N	SD 1.5 ($\bar{s}$=3.7)		SDXL ($\bar{s}$=4.0)	
	DMDA Steps	Speedup	DMDA Steps	Speedup
8	80	5.0×	57	3.5×
16	109	7.3×	89	4.5×

N	SD 1.5 ($\bar{s}$=3.7)		SDXL ($\bar{s}$=4.0)	
30	161	9.3×	145	5.2×
60	272	11.0×	265	5.7×
120	494	12.1×	505	5.9×
∞	—	**13.5×**	—	**6.25×**

Table 1. Speedup scaling. SD 1.5: Steps = 50 + 3.7N, limit = 50/3.7 = 13.5×. SDXL: Steps = 25 + 4.0N, limit = 25/4.0 = 6.25×.

4.6 DMDA vs. Existing Methods

Dimension	DDIM/DPM-Solver	Distillation	DeepCache	DMDA (ours)
What it reduces	Steps per image	Steps per image	Per-step compute	Frames needing full computation
Training required	No	Yes (expensive)	No	**No**
Architecture dependency	Scheduler only	Model-specific	U-Net only	**Architecture-agnostic**
Quality guarantee	ODE theory	Empirical	Heuristic	**Eigenvalue bound**
Multi-frame speedup	1×	1×	~2.3×	**5.1–9.4× (measured)**
Composable with others	—	—	Limited	**Yes (multiplicative)**

Table 2. Comparison with existing methods.

Composability: Distillation (50 → 4 steps) + DMDA on SD 1.5: Standard 1,500 steps → Combined ~34 steps (**~44× total speedup**). On SDXL: Standard 750 steps → Combined ~37 steps (**~20× total speedup**).

5. Discussion

5.1 Universality of Eigenmode Sparsity

The cross-architecture validation reveals that eigenmode sparsity is not specific to SD 1.5 but is a **structural property of diffusion latent spaces**. Both SD 1.5 (860M, single encoder) and SDXL (2.6B, dual encoder) exhibit dominant single-mode concentration despite their significantly different architectures, parameter counts, and latent resolutions. This suggests that the density matrix decomposition $\rho = \rho_shared + \Delta\rho(t)$ exploits a fundamental characteristic of the denoising process: sequentially generated latents share a persistent background structure that dominates the variation space.

The decreasing λ_1 with increasing model capacity (0.86 → 0.71) suggests that larger models distribute energy more broadly across eigenmodes, consistent with their greater representational capacity. However, even at 70.93%, the dominant mode still captures the majority of total energy, confirming that DMDA remains effective across the model scale spectrum.

5.2 Implications for Video Diffusion Models

The current validation uses image U-Nets to generate independent frames with correlated noise. True video diffusion models (AnimateDiff, SVD, Sora) additionally incorporate temporal attention layers that enforce motion coherence. We hypothesize that temporal attention will **increase** eigenmode concentration because the model explicitly enforces cross-frame consistency, amplifying the shared structure that DMDA exploits. Empirical validation on AnimateDiff is in preparation.

5.3 Real-Time Video Generation Path

Combining DMDA with existing acceleration methods creates a viable path to real-time video generation:

Configuration	30-frame Steps (SD 1.5)	Estimated Time (A100)
Standard	1,500	~120s
Distillation only (50→4 steps)	120	~8.4s
DMDA only (9.4×)	160	~13s
Distillation + DMDA	**~34**	**~2.4s**

At 2.4 seconds for 30 frames (1 second at 30fps), this approaches real-time. On H100, the estimate drops to ~1.4 seconds — sub-real-time. For SDXL, the combined speedup would yield approximately 37 steps for 30 frames at 1024×1024 resolution, enabling high-resolution near-real-time generation.

5.4 Residual Diffusion as Primary Bottleneck

In both models, residual diffusion dominates wall-clock time (SD 1.5: 59.3%, SDXL: 73.3%). This is because residual frames are currently processed sequentially. Three optimization strategies are available:

1. **Batch processing:** Group frames sharing the same timestep and process them in a single UNet forward pass.
2. **Scheduler reuse:** Replace per-frame deepcopy(scheduler) with a shared scheduler instance.
3. **Reduced guidance:** Residual diffusion uses half guidance scale (3.75 vs 7.5), suggesting that further reduction or elimination may be feasible.

Conservative estimates suggest 2–4× reduction in residual phase time, yielding total SD 1.5 time of ~15–20s and SDXL time of ~60–80s.

5.5 Limitations and Future Work

1. **Perceptual quality metrics:** Current evaluation is visual inspection. Pixel-space FVD, SSIM, and LPIPS against standard generation are planned.

2. **Video model validation:** Extension to AnimateDiff (temporal attention), Stable Video Diffusion, and transformer-based architectures (DiT) to confirm eigenmode sparsity universality.

3. **Eigenvalue-quality relationship:** Formal characterization of how λ_1 magnitude correlates with achievable speedup and reconstruction quality across model families.

4. **Cross-domain validation:** Application to other sequential diffusion domains including 3D generation, molecular structure prediction, and audio synthesis.

6. Conclusion

We have presented DMDA, a training-free diffusion acceleration framework validated on two architecturally distinct models: Stable Diffusion 1.5 (9.4× step reduction, 35.25s) and SDXL (5.1× step reduction, 151.1s), both generating 30 frames on Mac Studio M5 Ultra with visual quality equivalent to standard outputs.

The cross-architecture validation establishes eigenmode sparsity as a general property of diffusion latent spaces: SD 1.5 concentrates 85.76% of energy in a single eigenmode (λ_1 = 0.8576), while SDXL concentrates 70.93% (λ_1 = 0.7093). In both cases, this spectral concentration enables the fundamental DMDA decomposition $\rho = \rho_shared + \Delta\rho(t)$, separating shared structure (computed once) from per-frame residuals (computed cheaply).

DMDA operates on a different computational axis from existing acceleration methods — reducing inter-frame redundancy rather than per-frame step count — and composes multiplicatively with per-frame methods. The asymptotic speedup of 6.25–13.5× at long sequences, combined with composability with distillation (~20–44× combined), establishes a viable path toward real-time video generation across model scales. Extended validation on video diffusion models with temporal attention is in preparation.

References

Fu, Y. et al. (2025). ManifoldFormer: Geometric Deep Learning for Neural Dynamics on Riemannian Manifolds. arXiv:2511.16828.

Guo, Y. et al. (2023). AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. arXiv:2307.04725.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. NeurIPS 33, 6840–6851.

Lu, C. et al. (2022). DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. NeurIPS.

Ma, X. et al. (2023). DeepCache: Accelerating Diffusion Models for Free. arXiv:2312.00858.

Rombach, R. et al. (2022). High-Resolution Image Synthesis with Latent Diffusion Models. CVPR.

Schuld, M. & Petruccione, F. (2021). Machine Learning with Quantum Computers. Springer.

Song, J. et al. (2020). Denoising Diffusion Implicit Models. arXiv:2010.02502.

Song, Y. et al. (2023). Consistency Models. ICML.

Sun, H. et al. (2024). HunyuanVideo: A Systematic Framework for Large Video Generation Model. arXiv:2412.03603.