Accelerating Diffusion via Compressed Sensing: Applications to Imaging and Finance

Zhengyi Guo Jiatu Li Wenpin Tang David D. Yao
Columbia University
{zg2525, jl6969, wt2319, ddy1}@columbia.edu

Abstract

We integrate compressed sensing with diffusion models to accelerate synthetic data generation. Our pipeline, *Compressed-Space Diffusion Modeling (CSDM)*, first projects data from the ambient space to a latent space and trains a diffusion model in that space, then apply a compressed sensing algorithm to the latent samples to decode them back to the original space, with the goal of improving the efficiency of both training and inference. Under certain sparsity assumptions on the data, our approach achieves provably faster convergence by combining diffusion inference with sparse recovery, and it sheds light on the choice of the latent-space dimension. To illustrate the effectiveness of this approach, we present experiments on medical imaging data and financial time series for stress testing.

1 Introduction

Diffusion models have driven recent advances in text-to-image and text-to-video generation (e.g., DALL·E 2 Ramesh et al. [2022], Stable Diffusion Rombach et al. [2022], Sora OpenAI [2024], Make-A-Video Singer et al. [2023], Veo Google [2024]), and are increasingly used beyond vision Nie et al. [2025], Khanna et al. [2025]. However, in operations research and management settings, training and inference often require prohibitively many function evaluations in high-dimensional ambient spaces, creating memory and runtime bottlenecks that hinder real-time or on-device use.

Many datasets admit low-dimensional structure Donoho [2006], Pope et al. [2021], motivating diffusion in compressed or latent spaces Karras et al. [2022], Chen et al. [2023, 2025]. We pursue this direction and develop *Compressed-Space Diffusion Modeling (CSDM)*, which integrates compressed sensing with diffusion: (i) compress data from $\mathbb{R}^d$ to a low-dimensional space $\mathbb{R}^m$ ($m \ll d$); (ii) train/sample a diffusion model in $\mathbb{R}^m$; (iii) reconstruct to $\mathbb{R}^d$ via sparse recovery algorithms (e.g. FISTA Beck and Teboulle [2009b,a]). This compute-aware pipeline reduces the cost of score evaluation, backpropagation, and sampling, while maintaining task-relevant fidelity.

Contributions. (1) *Method.* We embed a sparse-recovery decoder into the diffusion workflow via the above three-step pipeline. (2) *Theory.* We analyze compute–accuracy tradeoffs with statements that separate diffusion and recovery errors and that guide the choice of the compressed dimension m; in very sparse regimes, DDIM-style VP sampling with FISTA yields overall complexity $\mathcal{O}(d^{1/3})$ and suggests $m = \mathcal{O}(d^{2/3})$. (3) *Evidence.* In the main text we report results on **OCTMNIST (medical imaging)** and **financial time series**, showing substantial wall-clock savings while preserving summary statistics and tail behavior critical for decision-making; additional results are deferred to the appendix.

Our target applications are decision-centric workflows that rely on scenario generation under tight compute budgets (e.g., medical analysis or portfolio construction). Preliminaries and full proofs are kept in the appendix; a compact schematic of the pipeline is also provided there.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

2 Method and Main Statements

Diffusion models. A diffusion model learns a data distribution $p_{\text{data}}(\cdot)$ through a forward SDE

$$dX_t = f(t, X_t)dt + g(t)dW_t, \quad X_0 \sim p_{\text{data}}(\cdot),$$

and samples new data via a reversed SDE:

$$d\widetilde{X}_t = \left(-f(T-t, \widetilde{X}_t) + g^2(T-t)\nabla \log p(T-t, \widetilde{X}_t)\right) dt + g(T-t)dB_t, \quad \widetilde{X}_0 \sim p(T, \cdot)$$

In practice, the score $\nabla \log p(t, x)$ is approximated by neural nets $s_\theta(t, x)$ using *denoising score matching* (DSM). In practice, we use discretizations of the reversed process to do training and inference. Recent results Gao et al. [2025], Gao and Zhu [2025] show that the Variance-Preserving (VP) model requires $n_{\text{diff}} = \mathcal{O}(d/\epsilon^2)$ (SDE Sampler) or $\mathcal{O}(\sqrt{d}/\epsilon)$ (Probabilistic ODE Sampler) steps to achieve accuracy ϵ.

Compressed sensing. For a sketching matrix $A \in \mathbb{R}^{m\times d}$, compressed sensing recovers an S-sparse signal $x \in \mathbb{R}^d$ from $y = Ax + e \in \mathbb{R}^m$ ($m \ll d$), where e is some unknown perturbation with $\|e\|_2 \le \sigma$ (σ is known). The robust recovery is given by the convex program

$$\min \|x\|_1 \quad \text{s.t.} \quad \|Ax - y\|_2 \le \sigma,$$

which can be reformulated as the Lasso problem

$$\min \tfrac{1}{2}\|Ax - y\|_2^2 + \lambda \|x\|_1.$$

We solve it via the *Fast Iterative Shrinkage-Thresholding Algorithm* (FISTA) Beck and Teboulle [2009a], whose convergence rate is $\mathcal{O}(1/k^2)$ in function value and $\mathcal{O}(1/k)$ in iterates. To reach error ϵ, FISTA requires $\mathcal{O}(s_{\max}^2(A)/\epsilon)$ iterations, where $s_{\max}(A)$ is the largest singular value of A. More details on FISTA and its convergence rate can be found in Appendix A.

Now we introduce our CSDM algorithm. The idea is to compress data to low dimension using a sketching matrix A, train a diffusion model there, and then recover full signals via FISTA.

Algorithm 1 Compressed-Space Diffusion Modeling (CSDM)

Require: Sketching matrix A, training data distribution $p_{\text{data}}(\cdot)$

1: Compress data: $\widetilde{p}_{\text{data}}(\cdot) = A_{\#}p_{\text{data}}(\cdot)$, where $A_{\#}p_{\text{data}}(\cdot)$ denotes the pushforward of $p_{\text{data}}(\cdot)$ by A.
2: Train a diffusion model using samples drawn from $\widetilde{p}_{\text{data}}(\cdot)$ and generate signals.
3: Reconstruct signals by solving the Lasso problem with FISTA using the sampled data.

Theorem 1 *Let $(x, \widetilde{y})$ be defined on the same probability space such that $x \sim p_{data}(\cdot)$, and $\widetilde{y}$ is output by the diffusion model in algorithm 1. Assume that $\|Ax - \widetilde{y}\|_2 \le \sigma$ with high probability. Also let $p_{data}(\cdot)$ enjoys S-sparsity and A satisfies the restricted isometry property (see Appendix B). For $\{x_k\}_{k\ge 0}$ the FISTA iterates relative to $\widetilde{y}$, we have with high probability,*

$$\|x_k - x\|_2 \le C\left(\sigma + \frac{s_{\max}^2(A) + \sqrt{S}}{k}\right), \quad \textit{for } k \textit{ sufficiently large}. \tag{1}$$

Proof is provided in Appendix C.

It is common to choose the sketching matrix $A \in \mathbb{R}^{m\times d}$ to be random, e.g., each entry of A is a Gaussian variable with mean 0 and variance $\frac{1}{m}$. By extreme value theory of random matrices Rudelson and Vershynin [2010], the largest singular value $s_{\max}(A) \lesssim \sqrt{\frac{d}{m}}$ with high probability.

Corollary 1 *Let the assumptions in Theorem 1 hold. Assume that $p_{data}(\cdot)$ is strongly log-concave, the score $\nabla \log p(t, x)$ is Lipschitz, and the score matching error $\mathbb{E}_{X\sim p(t,\cdot)}\|s_{\theta_*}(t, X) - \nabla \log p(t, X)\|_2^2 < \epsilon^2$ holds, with $\widetilde{y}$ be the output of the discretized VP model in k' steps, and $\{x_{k',k}\}_{k\ge 0}$ be the FISTA iterates as to $\widetilde{y}$. Then:*

$$\|x_{k,k'} - x\|_2 \lesssim \begin{cases} \sqrt{\frac{m}{k'}} + \frac{1}{k}(\frac{d}{m} + \sqrt{S}) & \textit{for the stochastic sampler}, \\ \frac{\sqrt{m}}{k'} + \frac{1}{k}(\frac{d}{m} + \sqrt{S}) & \textit{for the deterministic sampler}. \end{cases} \tag{2}$$

Complexity tradeoff. The error has two parts: diffusion sampling and compressed-sensing optimization. Balancing them yields the overall complexity

$$\mathcal{O}\Big(\max\{m,\ d/m + \sqrt{S}\}\Big).$$

Considering the very sparse case $S = \mathcal{O}(1)$, the optimal dimension $m = \mathcal{O}(\sqrt{d})$ (stochastic VP) gives $\mathcal{O}(\sqrt{d})$ complexity, and $m = \mathcal{O}(d^{2/3})$ (deterministic VP) gives $\mathcal{O}(d^{1/3})$. A summary is in Table 1.

Table 1: Optimal m and complexity of CSDM under different samplers

Sampling	VP (Det.)	VP (Stoch.)	VE (Det.)	VE (Stoch.)
m	$d^{2/3}$	$d^{1/2}$	$d^{2/5}$	$d^{2/3}$
Complexity	$d^{1/3}$	$d^{1/2}$	$d^{3/5}$	$d^{1/3}$

3 Experiments

3.1 OCTMNIST (Medical Imaging Data)

Setup. We use OCTMNIST retinal B-scans from MedMNIST. Due to the dataset's inherently low resolution, we adopt an upscaling strategy: we resize images to several larger ambient resolutions, $d \in \{32^2, 40^2, 48^2\}$, while preserving their inherent sparsity structure, and we fix the sketch dimension at $m = 28^2 = 784$. This allows us to evaluate our method under different levels of compression. We train a VP diffusion model in $\mathbb{R}^m$ and decode to $\mathbb{R}^d$ via FISTA.

Metrics and results. We report wall-clock time for low-dimensional diffusion ($\mathrm{Diffuse}_m$) and recovery ($\mathrm{Recover}$). We then compute overall speedup against a baseline that performs diffusion directly in $\mathbb{R}^d$ (stochastic sampler) as

$$\mathrm{Speedup} = 1 - \frac{\mathrm{Diffuse}_m + \mathrm{Recover}}{\mathrm{Diffuse}_d}. \tag{3}$$

As shown in table 2, with growing d (lower retention m/d), $\mathrm{Diffuse}_m$ stays roughly constant and recovery adds a small overhead, while the $\mathrm{Diffuse}_d$ baseline grows with d, yielding larger net gains.

Table 2: Comparison of generation time on OCTMNIST

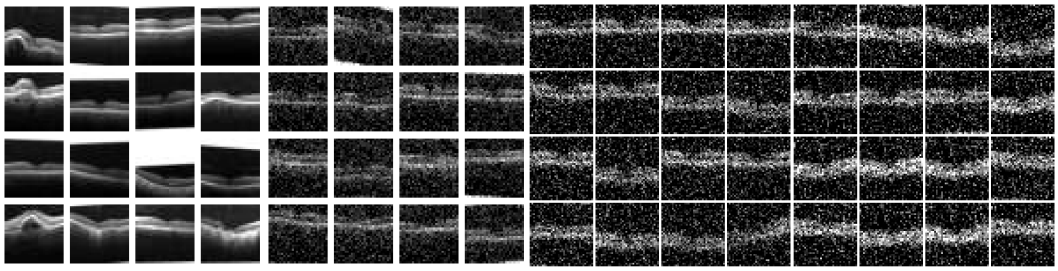
Compression	Original Dim.	Original Dim. Inference Time	Low Dim. Inference Time	Recovery Time	Speedup
77%	$1024 \to 784$	1.6465s / pic	1.2606s / pic	0.1519s / pic	**4.99%**
49%	$1600 \to 784$	1.9243s / pic	0.9429s / pic	0.1541s / pic	**42.99%**
34%	$2304 \to 784$	2.8735s / pic	1.1076s / pic	0.1556s / pic	**56.04%**

Fidelity (decision-relevant structure). Across compression levels, the retinal band remains contiguous and well localized. At aggressive compression (low retention), artifacts manifest primarily as mild intra-band speckle and slight softening of sharp transitions; background remains largely quiescent.

Takeaway. CSDM attains substantial wall-clock savings on OCT while preserving band continuity/localization that are critical for downstream medical analysis. The gains align with the compute–accuracy trade-offs in Sec. 2: increasing d (fixed m) improves speedup as $\mathrm{Diffuse}_d$ scales with dimension whereas $\mathrm{Diffuse}_m + \mathrm{Recover}$ remains stable.

3.2 Time Series Data

Setup. We further study integrating diffusion models with dimension reduction for financial time series. While prior work Aghapour et al. [2025], Chen et al. [2025] applies diffusion to portfolio optimization, we target stress testing. Rather than random projections (and compressed sensing), we use principal component analysis (PCA) to extract the dominant variance directions of macroeconomic factors, train a diffusion model in the principal-component (PC) space to generate synthetic PC data, and treat these PCs as "informative" factors for portfolio backtesting and stress testing via regression analysis.



(a) Original OCT samples (b) 77% dims retained (c) 49% dims retained (d) 34% dims retained

Figure 1: OCTMNIST generations at three compression levels.

Data and model. We train a diffusion model in a low-dimensional macro-factor space given by the first 6 PCs from 126 FRED-MD factors McCracken and Ng [2016], capturing over 90% cumulative variance. We then generate synthetic PC paths and map them to selected equities (AAPL, AMZN, COST, CVX, GOOGL, JPM, KO, MCD, NVDA, UNH) for portfolio construction.

Metrics. Evaluation uses 6-month cumulative log-returns under a Risk-Parity portfolio. If the PCs retain key risk directions, the generated data should reproduce distributional properties (center, dispersion, tails).

Table 3: Risk-Parity portfolio weights comparison (real vs. generated).

Source	AAPL	AMZN	COST	CVX	GOOGL	JPM	KO	MCD	NVDA	UNH
Real (%)	7.90	8.37	11.56	9.05	9.23	8.23	14.67	14.27	5.42	11.30
Generated (%)	7.31	8.62	12.29	9.04	9.43	8.50	14.71	13.50	5.14	11.46

Results. Table 3 reports the risk-parity weights, which are similar in both settings. Figure 2 provides the histograms of real and generated log-returns. The summary statistics (Table 4) shows that mean and volatilities match, and the left-tail quantiles differ by only 0.01–0.04pp, indicating that tail risk is effectively captured when portfolios are constructed from risk-balanced exposures.

Table 4: Risk-Parity: real vs. generated 6M log-return statistics.

Statistics	Real RP	Predicted RP
Mean	8.55%	8.54%
Median	8.85%	8.71%
Std Dev	7.43%	7.41%
1% Quantile	-11.33%	-11.32%
5% Quantile	-3.47%	-3.51%

Takeaway. Overall, training a diffusion model in a low-dimensional macro-PC space reliably reproduces real-data distributions for risk-parity backtests. Crucially, left-tail quantiles closely match empirical values. Risk-parity weight patterns are likewise consistent, indicating the PCs retain the covariance structure that drives risk-aware portfolio construction.

4 Conclusion

We proposed *Compressed-Space Diffusion Modeling* (CSDM), which integrates compressed sensing with diffusion to accelerate data generation by sampling in a low-dimensional sketch and reconstructing via sparse recovery. Under sparsity assumptions, we provided error/compute statements that imply faster convergence and guide the choice of the compressed dimension. Empirically, we demonstrated substantial wall-clock savings while preserving decision-relevant structure on OCTMNIST and selected financial tasks; additional results appear in the appendix.

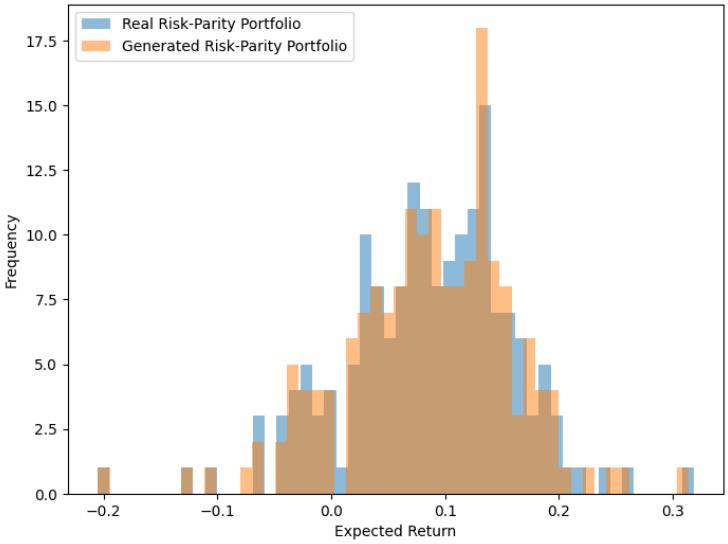


Figure 2: Risk-parity comparison.

Future work includes deriving sharper convergence rates for FISTA with explicit dimension dependence, extending CSDM to conditional/guided generation and diffusion model alignment Dhariwal and Nichol [2021], Ho and Salimans [2021], Karras et al. [2024], Black et al. [2024], Fan et al. [2023], Zhao et al. [2025], and developing theory for PCA+diffusion variants.

References

Ahmad Aghapour, Erhan Bayraktar, and Fengyi Yuan. Solving dynamic portfolio selection problems via score-based diffusion models. 2025. arXiv:2507.09916.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *ICASSP*, pages 693–696, 2009a.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009b.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.

Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W. Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Math. Program.*, 165(2):471–507, 2017.

Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1207–1223, 2006.

Antonin Chambolle, Ronald A. DeVore, Nam-yong Lee, and Bradley J. Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. Image Process.*, 7(3):319–335, 1998.

Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *ICML*, pages 4672–4712, 2023.

Minshuo Chen, Renyuan Xu, Yumin Xu, and Ruixun Zhang. Diffusion factor models: Generating high-dimensional returns with factor structure. 2025. arXiv:2504.06566.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–1457, 2004.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Neurips*, volume 34, pages 8780–8794, 2021.

David L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, 2006.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Neurips*, volume 36, pages 79858–79885, 2023.

Xuefeng Gao and Lingjiong Zhu. Convergence analysis for general probability flow odes of diffusion models in wasserstein distances. In *AISTATS*, pages 1009–1017, 2025.

Xuefeng Gao, Hoang M Nguyen, and Lingjiong Zhu. Wasserstein convergence guarantees for a general class of score-based generative models. *J. Mach. Learn. Res.*, 26(43):1–54, 2025.

Google. State-of-the-art video and image generation with Veo 2 and Imagen 3. 2024. Available at `https://blog.google/technology/google-labs/video-image-generation-update-december-2024/`.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Patrick R Johnstone and Pierre Moulin. A lyapunov analysis of fista with local linear convergence for sparse optimization. 2015. arXiv:1502.02281.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Neurips*, volume 35, pages 26565–26577, 2022.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Neurips*, volume 37, pages 52996–53021, 2024.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, and Stefano Ermon. Mercury: Ultra-fast language models based on diffusion. 2025. arXiv:2506.17298.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.

Michael W McCracken and Serena Ng. Fred-md: A monthly database for macroeconomic research. *J. Bus. Econ. Stat.*, 34(4):574–589, 2016.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. 2025. arXiv:2502.09992.

OpenAI. Sora: Creating video from text. 2024. Available at `https://openai.com/sora`.

Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *ICLR*, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. 2022. arXiv:2204.06125.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

Mark Rudelson and Roman Vershynin. Non-asymptotic theory of random matrices: extreme singular values. In *Proceedings of the International Congress of Mathematicians. Volume III*, pages 1576–1602. Hindustan Book Agency, New Delhi, 2010.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, and Oran Gafni. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023.

Shaozhe Tao, Daniel Boley, and Shuzhong Zhang. Local linear convergence of ISTA and FISTA on the LASSO problem. *SIAM J. Optim.*, 26(1):313–336, 2016.

Hanyang Zhao, Haoxian Chen, Ji Zhang, David Yao, and Wenpin Tang. Score as Action: Fine tuning diffusion generative models by continuous-time reinforcement learning. In *ICML*, 2025.

A Detail of FISTA and its Convergence Rate

We denote $f(x) := \frac{1}{2}|Ax - y|_2^2$ and $g(x) := \lambda|x|_1$. Note that $\nabla f(x) = A^T(Ax - y)$, so

$$|\nabla f(x) - \nabla f(x')|_2 \leq L|x - x'|_2 \quad \text{for all } x, x' \in \mathbb{R}^d, \tag{4}$$

where $L := \lambda_{\max}(A^T A)$ is the largest eigenvalue of $A^T A$. Define

$$Q_L(x, x') := f(x') + \nabla f(x') \cdot (x - x') + \frac{L}{2}|x - x'|_2^2 + g(x),$$

and

$$\begin{aligned} p_L(x') &= \arg\max_x Q_L(x, x') \\ &= \arg\max_x \left\{ \frac{L}{2} \left| x - \left(x' - \frac{\nabla f(x')}{L} \right) \right|_2^2 + g(x) \right\} \\ &= \text{SoftThreshold}\left(x' - \frac{\nabla f(x')}{L}, \frac{\lambda}{L} \right), \end{aligned} \tag{5}$$

where the soft-thresholding operator is applied coordinate-wise Chambolle et al. [1998], Daubechies et al. [2004]:

$$\text{SoftThreshold}(x, a)_i := \begin{cases} x_i - a & \text{if } x_i > a, \\ 0 & \text{if } |x_i| \leq a, \\ x_i + a & \text{if } x_i < -a. \end{cases}$$

FISTA is a proximal gradient method by incorporating the Nesterov acceleration.

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)
Input: L (Lipschitz constant of ∇f).
Step 0. Take $y_1 = x_0 \in \mathbb{R}^d$, $t_1 = 1$.
Step k. Compute

$$\begin{aligned} x_k &= p_L(y_k), \\ t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \\ y_{k+1} &= x_k + \left(\frac{t_k - 1}{t_{k+1}} \right)(x_k - x_{k-1}). \end{aligned}$$

FISTA is a proximal gradient method by incorporating Nesterov acceleration. The convergence result is as follows.

Theorem 2 *Beck and Teboulle [2009b], Bolte et al. [2017] Let x_* be the solution to the problem* (2)*, and $\{x_k\}_{k \geq 0}$ the FISTA iterates. We have for k sufficiently large,*

$$F(x_k) - F(x^*) \leq \frac{CL}{k^2} \quad \textit{and} \quad |x_k - x_*|_2 \leq \frac{C(L + |y|_2)}{k},$$

for some $C > 0$.

To ensure that $|x_k - x_*|_2 \leq \epsilon$, it requires the number of iterations $n_{\text{CS}} = \mathcal{O}\left(\frac{L}{\epsilon}\right) = \mathcal{O}(\frac{s_{\max}^2(A)}{\epsilon})$, where $s_{\max}(A)$ is the largest singular value of A. Also refer to Johnstone and Moulin [2015], Tao et al. [2016] for sharper convergence results of FISTA (but implicit in the dependence on dimension).

B Restricted Isometry Property

Let A be the matrix with the finite collection of vectors $(v_j)_{j \in J} \in \mathbb{R}^m$ as columns. For each $1 \leq S \leq |J|$, we define the S-restricted isometry constant δ_S to be the smallest quantity such that A_T obeys

$$(1 - \delta_S)|c|_2^2 \leq |A_T c|_2^2 \leq (1 + \delta_S)|c|_2^2,$$

for all subsets $T \subset J$ of cardinality at most S, and all real coefficients $(c_j)_{j \in T}$.

The numbers δ_S measure how close the vectors v_j behave like an orthonormal system, but only when restricting to sparse linear combinations involving no more than S vectors. The following theorem concerns sparse recovery for robust compressed sensing.

Theorem 3 *(Restricted Isometry Property)Candès et al. [2006] Let S be such that $\delta_{3S} + 3\delta_{4S} < 2$. Then for any signal x supported on T with $|T| \leq S$ (referred to as S-sparse), and any perturbation e with $|e|_2 \leq \sigma$,*

$$|x_* - x|_2 \leq C_S \, \sigma,$$

where x_* is the solution to the problem (2)*, and the constant C_S only depends on δ_{4S}.*

C Proof of Theorem 1

Let x_* be the solution to the problem:

$$\min |x|_1 \quad \text{subject to} \quad |Ax - \widetilde{y}|_2 \leq \sigma.$$

By Theorem 3, we have $|x - x_*|_2 \leq C\sigma$ for some $C > 0$. Further by Theorem 2, we have for k sufficiently large,

$$|x_k - x_*|_2 \leq \frac{C(L + |\widetilde{y}|_2)}{k} \leq \frac{C(L + \sigma + |Ax|_2)}{k}.$$

Under the assumption of Theorem 3, the term $|Ax|_2$ is of order $\mathcal{O}(\sqrt{S})$. Thus, we get $|x_k - x_*|_2 \leq \frac{C'(L+\sigma+\sqrt{S})}{k}$ for some $C' > 0$ and for k sufficiently large. By triangle inequality, we have $|x_k - x|_2 \leq |x_k - x_*|_2 + |x_* - x|_2$, which yields the desired result.

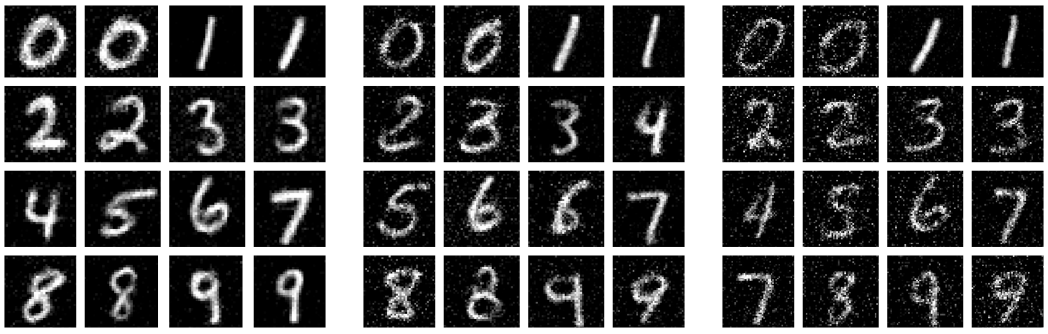
D MNIST Experiment Results

The MNIST dataset LeCun et al. [1998] consists of images of handwritten digits, and on average, over 80% of the pixels in each image have intensity values equal or very close to zero. As mentioned, we resize the images to the ambient resolutions $d \in \{32 \times 32, 40 \times 40, 48 \times 48\}$, and fix the compressed dimension at $m = 28 \times 28$. We train a VP model in $\mathbb{R}^m$ for diffusion inference, and decode the generated sample to $\mathbb{R}^d$ by FISTA.

Table 5 reports per-image wall-clock for (i) diffusion inference in $\mathbb{R}^m$ and (ii) FISTA recovery in $\mathbb{R}^d$, along with the speedup. As the ambient dimension d increases (or the retention m/d drops), the diffusion inference time in the latent space $\mathbb{R}^m$ stays roughly constant with the recovery adding a small overhead, while the diffusion inference in the ambient space grows with d. This leads to increasing net speedups (from 4.39% up to 61.13%).

Table 5: Comparison of generation time on MNIST

Compression	Original Dim.	Original Dim. Inference Time	Low Dim. Inference Time	Recovery Time	Speedup
76%	$1024 \to 784$	0.4463s / pic	0.3417s / pic	0.0852s / pic	**4.39%**
49%	$1600 \to 784$	1.1103s / pic	0.5441s / pic	0.0741s / pic	**44.32%**
34%	$2304 \to 784$	1.5987s / pic	0.5440s / pic	0.0774s / pic	**61.13%**



(a) 77% dimensions retained (b) 49% dimensions retained (c) 34% dimensions retained

Figure 3: MNIST generations at three compression levels.

Figure 3 illustrates CSDM generations at each compression level. With low compression/high retention (76%), digits are crisp and legible with thin strokes largely intact. But with high compression/low

retention (34%), we observe a higher background grain and occasional breaks in tight curves, with loop digits ($0/6/8$) and multi-segment (5) the first to degrade. Nevertheless, class identity remains visible in most samples, with the strong speedup at this compression. Overall, CSDM achieves substantial wall-clock savings while preserving digit identity over a wide range of compression; artifacts concentrate in thin/curved strokes at aggressive compression.

E ERA5 Reanalysis Experiment Results

ERA5 Reanalysis dataset is provided by ECMWF on the large-scale precipitation fraction (LSPF) field (see Figure 4 for illustration). In contrast with the previous two subsections, each snapshot is resized to a fixed ambient resolution of 80×80, and then compressed to the retention levels 64% (64×64), 49% (56×56), and 36% (48×48).

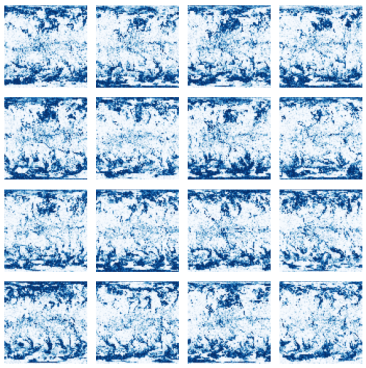


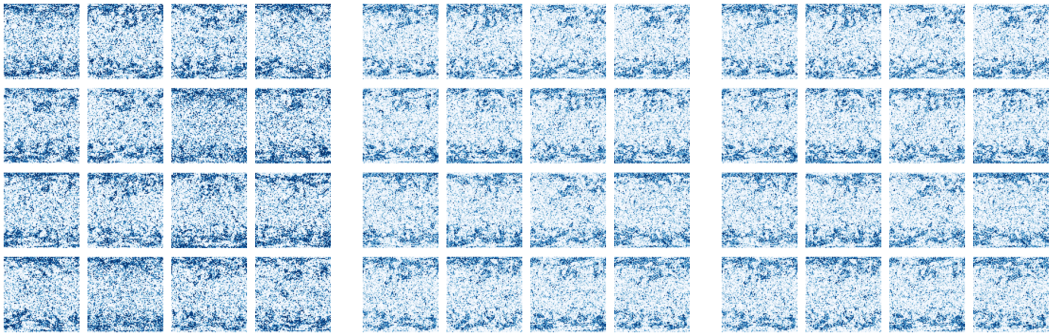
Figure 4: Original LSPF samples in Year 2023

Table 6 reports the per-sample wall-clock for diffusion inference in $\mathbb{R}^m$, FISTA recovery in $\mathbb{R}^d$, along with the speedup. As the level of compression increases, the diffusion inference time in the latent space shortens significantly with the recovery remaining lightweight; the net speedup increases steadily from 4.22% to 59.31%.

Table 6: Comparison of generation time on LSPF

Compression	Original Dim.	Original Dim. Inference Time	Low Dim. Inference Time	Recovery Time	Speedup
64%	6400 $\rightarrow$ 4096	13.7545s / pic	8.8029s / pic	4.3712s / pic	**4.22%**
49%	6400 $\rightarrow$ 3136	12.8296s / pic	6.2865s / pic	1.7669s / pic	**37.23%**
36%	6400 $\rightarrow$ 2304	12.2049s / pic	4.3938s / pic	0.5721s / pic	**59.31%**

Figure 5 illustrates CSDM generations at different compression levels. With low compression/high retention (64%), the generations are nearly indistinguishable from the full-resolution fields. Fine-scale precipitation patterns are well preserved, with only minor smoothing in localized regions. With high compression/low retention (36%), large-scale structures are still visible, but finer details are partially lost, and small patches may merge or vanish. Overall, CSDM achieves significant wall-clock savings, while retaining essential spatial patterns on a complex and low-sparsity climate dataset. As the level of compression increases, artifacts manifest primarily in the loss of local variability, but the large-scale precipitation dynamics remain intact for downstream geophysical and risk analysis.



(a) 64% dimensions retained (b) 49% dimensions retained (c) 36% dimensions retained

Figure 5: LSPF generations at three compression levels.