Technical Report: MNIST Diffusion Experiments on Purdue Gautschi

PurdueHPC_Codex Workflow

March 1, 2026

1 Scope and Environment

This report summarizes end-to-end execution of diffusion model experiments on Purdue RCAC Gautschi, including environment setup, Slurm execution, model details, and resulting artifacts. All commands and scripts were executed under:

- Scratch workspace: `/scratch/gautschi/rmaulik/codex_test`
- Source repository: `PurdueHPC_Codex`
- Account: `rmaulik`
- Partition: `ai`

No credentials or authentication secrets are stored in this report.

2 Allocation and Job Accounting

Current allocation snapshot from `slist` on Gautschi:

- AI partition GPU-hour balance: **43,795.8**
- CPU partition balance: **0**

Completed follow-up experiment jobs (EDM + intrinsic-dimension studies):

Job ID	Name	State	Elapsed (s)	GPUs
8213207	mnist_edm	COMPLETED	28	1
8213218	mnist_edm20	COMPLETED	30	1
8213341	svd70	COMPLETED	32	1
8213342	svd20	COMPLETED	32	1
8213437	ssvd70	COMPLETED	27	1
8213438	ssvd20	COMPLETED	27	1
8213702	gid70	COMPLETED	30	1
8213703	gid20	COMPLETED	30	1
8214018	psvd	COMPLETED	27	1
8214200	rsvd20	COMPLETED	36	1

8214779	rsvd0	COMPLETED	43	1

Estimated direct GPU-hours from these accounted runs:

$$\text{GPU-hours} = \frac{342}{3600} \times 1 = 0.095 \text{ GPU-hours (approx.)}$$

3 Diffusion Model Formulation (EDM)

The training script now uses an Elucidated Diffusion Model (EDM) style preconditioned denoiser over MNIST ($x_0 \in [-1, 1]^{1\times 28\times 28}$), where the model approximates a score-consistent denoising function across continuous noise levels σ.

3.1 EDM Preconditioning and Objective

Given $x = x_0 + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ and $\sigma \sim \log \mathcal{N}(p_{\text{mean}}, p_{\text{std}}^2)$, the network is used in preconditioned form:

$$\hat{x}_0 = c_{\text{skip}}(\sigma)x + c_{\text{out}}(\sigma)F_\theta(c_{\text{in}}(\sigma)x, c_{\text{noise}}(\sigma)),$$

where

$$c_{\text{in}} = \frac{1}{\sqrt{\sigma^2 + \sigma_{\text{data}}^2}}, \quad c_{\text{skip}} = \frac{\sigma_{\text{data}}^2}{\sigma^2 + \sigma_{\text{data}}^2}, \quad c_{\text{out}} = \frac{\sigma\sigma_{\text{data}}}{\sqrt{\sigma^2 + \sigma_{\text{data}}^2}}.$$

Training minimizes weighted denoising MSE:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,\sigma,\epsilon}\left[w(\sigma)\|\hat{x}_0 - x_0\|_2^2\right], \quad w(\sigma) = \frac{\sigma^2 + \sigma_{\text{data}}^2}{(\sigma\sigma_{\text{data}})^2}.$$

3.2 EDM Sampling

Sampling follows a Karras σ-schedule from $\sigma_{\max}$ to $\sigma_{\min}$ (then 0) and integrates the probability flow ODE in σ-space:

$$\frac{dx}{d\sigma} = \frac{x - \hat{x}_0(x, \sigma)}{\sigma},$$

with Euler step plus second-order (Heun) correction.

4 Network Architecture

The current model is a residual U-Net style denoiser with sinusoidal time embeddings:

- Stem: Conv $1 \to 64$
- Down path: residual blocks at 64 and 128 channels with strided downsampling ($28 \to 14 \to 7$)
- Bottleneck: two residual blocks at 256 channels
- Up path: transposed convolutions + skip concatenation + residual blocks ($7 \to 14 \to 28$)
- Output: GroupNorm + Conv $64 \to 1$
- Optimizer: AdamW, gradient clipping (max norm 1.0)

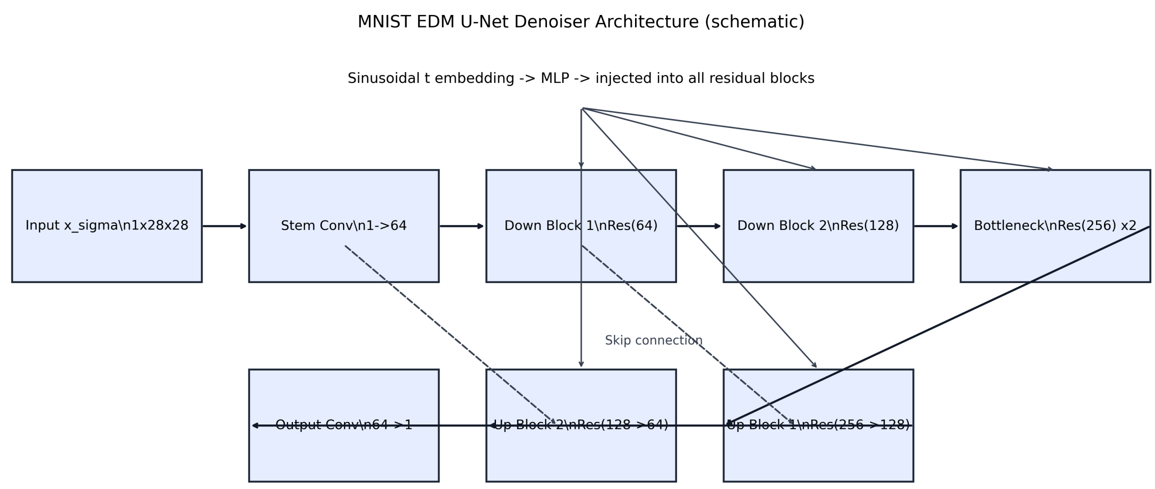

Figure 1: U-Net denoiser schematic generated by the EDM training pipeline.

5 Latest EDM Run Configuration

The latest EDM follow-up run (job `8213207`) used:

- Epochs: 1
- Karras sampling steps: 80
- Batch size: 128
- EDM parameters: $\sigma_{\text{min}} = 0.002$, $\sigma_{\text{max}} = 80.0$, $\rho = 7.0$, $\sigma_{\text{data}} = 0.5$
- Sigma training distribution: $p_{\text{mean}} = -1.2$, $p_{\text{std}} = 1.2$
- Slurm resources: 1 GPU, 14 CPUs, partition `ai`

From `metrics.json` (run `edm_dps_70pct_8213207`):

- Total steps: 469
- Final epoch mean loss: 0.273500
- Best epoch mean loss: 0.273500
- Device: CUDA

6 Results and Artifacts

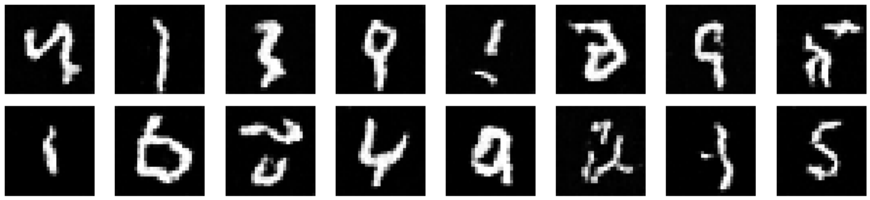

Figure 2: Generated MNIST samples from the EDM run (8213207).

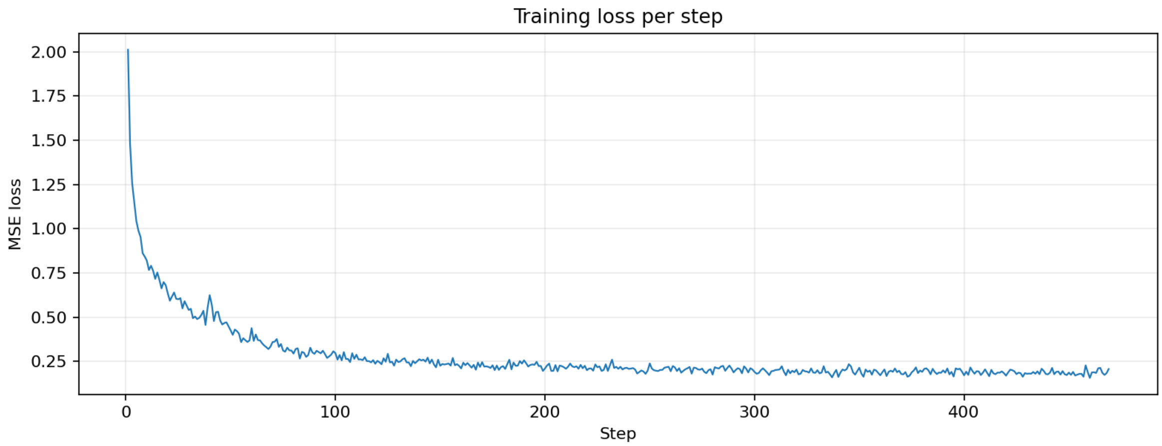

Figure 3: Per-step training loss trajectory for EDM run 8213207.

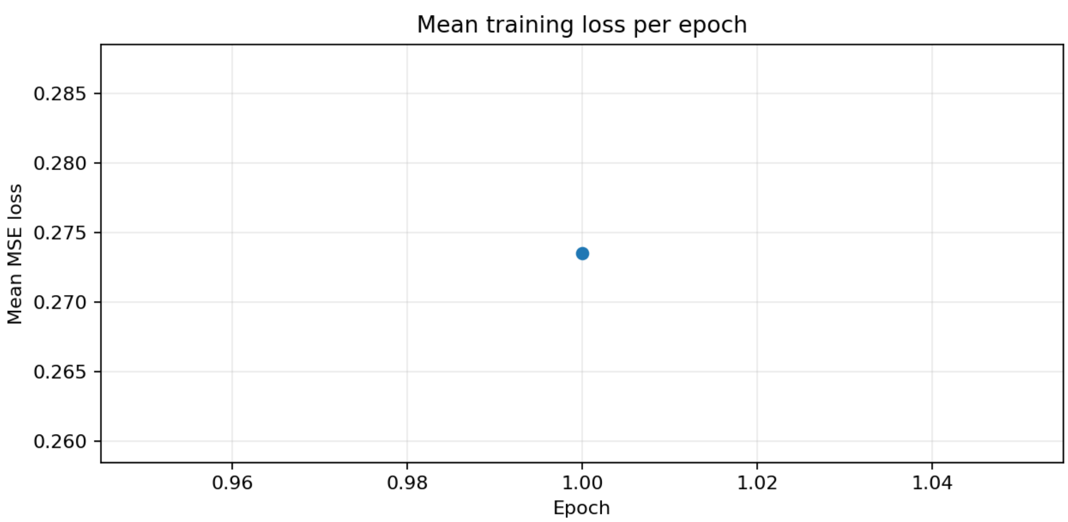

Figure 4: Per-epoch mean loss trajectory for EDM run 8213207.

7 Posterior Sampling with Partial Observations

We perform diffusion posterior sampling in EDM sigma-space. Let y denote observed pixels, $m \in \{0,1\}^{H\times W}$ the mask, and $\hat{x}_0(x,\sigma)$ the EDM denoised estimate.

Mathematical Formulation

Posterior guidance is applied through a masked Gaussian likelihood:

$$p(y \mid x_0) \propto \exp\left(-\frac{\|m \odot (y - x_0)\|_2^2}{2\sigma_t^2}\right), \quad \sigma_t^2 = \sigma_y^2 + c\sigma^2.$$

This yields the guidance gradient (used in code):

$$g_\sigma \approx \frac{m \odot (y - \hat{x}_0)}{\sigma_t^2}.$$

In EDM ODE form, we use:

$$\frac{dx}{d\sigma} \approx \frac{x - (\hat{x}_0 + \lambda_\sigma \sigma g_\sigma)}{\sigma},$$

with timestep-annealed guidance strength

$$\lambda_\sigma = \lambda_{\max}\left(\lambda_{\min} + (1-\lambda_{\min})\left(\frac{i}{N-1}\right)^p\right).$$

We additionally enforce projection-based data consistency:

$$\hat{x}_0 \leftarrow m \odot y + (1-m) \odot \hat{x}_0,$$

and during intermediate sigma steps:

$$x \leftarrow m \odot (y + \sigma\xi) + (1-m) \odot x.$$

The implemented strategy therefore combines:

- Noise-aware likelihood variance: $\sigma_t^2 = \sigma_y^2 + c\sigma^2$
- Annealed guidance schedule: weak guidance at high noise, stronger guidance near final denoise steps
- Data-consistency projection on observed pixels (in both $\hat{x}_0$ and sampled states)

Experiment A: 70% Pixels Revealed

- Observed fraction: 70% of pixels (30% occluded, random mask)
- Target digit class: 7
- Guidance scale: 1.5
- Guidance min fraction / power: 0.25 / 1.5
- Likelihood noise scale: 0.1

- Noise-aware coefficient: 0.05

Validated Gautschi posterior job:

- Job ID: `8213207`
- State: `COMPLETED`, exit code `0:0`
- Elapsed: `00:00:28`
- Run tag: `edm_dps_70pct_8213207`

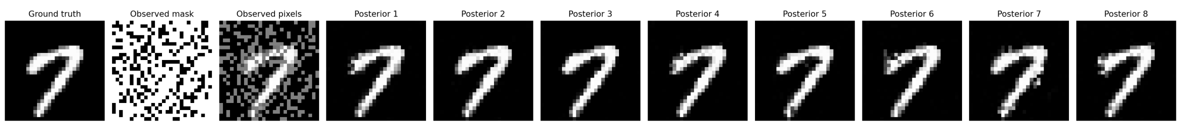

Figure 5: EDM posterior conditioning setup: ground truth, observed mask (70% observed), observed pixels, and posterior draws.

Figure 6: EDM posterior samples with noise-aware likelihood, guidance annealing, and projection-based data consistency (70% revealed).

Experiment B: 20% Pixels Revealed

- Observed fraction: 20% of pixels (80% occluded, random mask)
- Target digit class: 7
- Guidance scale: 1.5
- Guidance min fraction / power: 0.25 / 1.5
- Likelihood noise scale: 0.1
- Noise-aware coefficient: 0.05

Validated Gautschi posterior job:

- Job ID: `8213218`
- State: `COMPLETED`, exit code `0:0`
- Elapsed: `00:00:30`
- Run tag: `edm_dps_20pct_8213218`

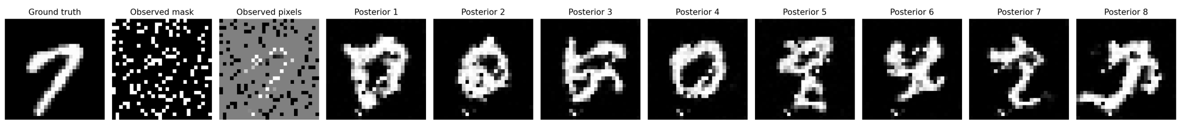

Figure 7: EDM posterior conditioning setup: ground truth, observed mask (20% observed), observed pixels, and posterior draws.

Figure 8: EDM posterior samples with noise-aware likelihood, guidance annealing, and projection-based data consistency (20% revealed).

8 Intrinsic Dimensionality of Posterior Samples

To analyze posterior sample complexity, we generated a large posterior ensemble for each setting and formed a snapshot matrix:

$$X = [x^{(1)}, x^{(2)}, \ldots, x^{(N)}] \in \mathbb{R}^{784 \times N},$$

where each column is a flattened 28×28 posterior sample and $N = 1024 > 784$. We computed

$$X = U \Sigma V^\top,$$

and inspected the singular value spectrum $\{\sigma_i\}$ and cumulative energy. For this SVD-based analysis, we report

$$d_{95} = \min \left\{ k : \frac{\sum_{i=1}^k \sigma_i^2}{\sum_j \sigma_j^2} \geq 0.95 \right\}, \qquad d_{99} = \min \left\{ k : \frac{\sum_{i=1}^k \sigma_i^2}{\sum_j \sigma_j^2} \geq 0.99 \right\}.$$

SVD jobs on Gautschi:

- 70% observed: job `8213341`, run `edm_svd_70pct_8213341`
- 20% observed: job `8213342`, run `edm_svd_20pct_8213342`
- Prior (unconditional): job `8214018`, run `edm_prior_svd_8214018`

Case	Pixels	Snapshots	d_{95}	d_{99}	Participation ratio
70%	784	1024	1	3	1.023
20%	784	1024	45	152	1.718
Prior (unconditional)	784	1024	47	158	1.812

Table 2: Singular-spectrum summary from sample snapshot matrices (posterior-conditioned and prior-unconditional).

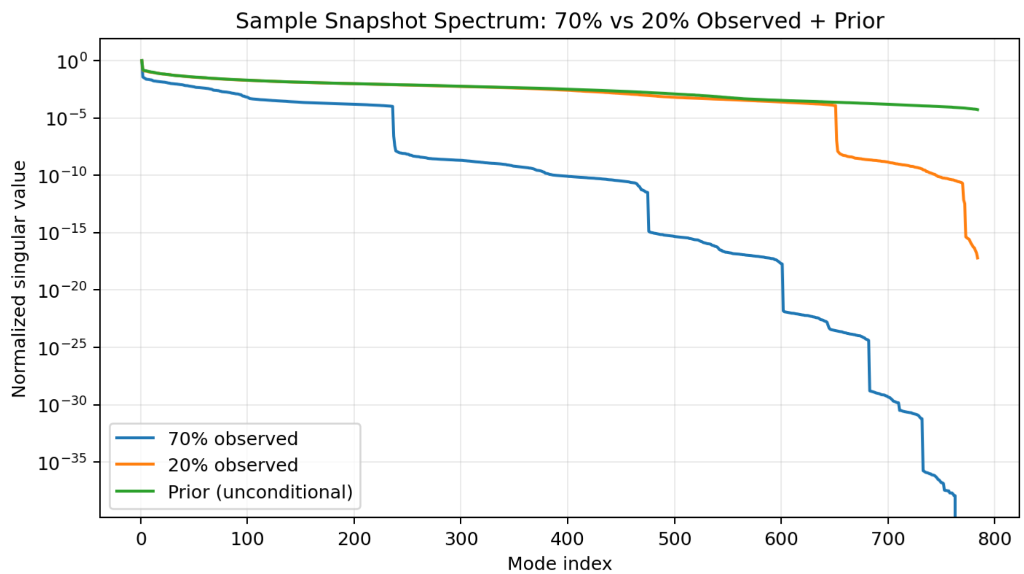

Figure 9: Normalized singular value spectrum for sample snapshots: 70% observed, 20% observed, and prior-unconditional.

Additional Subsection: Score-Snapshot SVD and Combined Spectra

We repeated the intrinsic dimensionality analysis by constructing snapshots from the *estimated score field* instead of the samples. For each posterior sample $x^{(j)}$, we evaluated the approximated score at fixed $\sigma_{\text{eval}} = 0.1$:

$$s^{(j)} \approx \frac{\hat{x}_0(x^{(j)}, \sigma_{\text{eval}}) - x^{(j)}}{\sigma_{\text{eval}}^2},$$

then formed

$$S = [s^{(1)}, s^{(2)}, \ldots, s^{(N)}] \in \mathbb{R}^{784 \times N},$$

with $N = 1024 > 784$, and computed its SVD.

Score-snapshot jobs on Gautschi:

- 70% observed: job `8213437`, run `edm_score_svd_70pct_8213437`
- 20% observed: job `8213438`, run `edm_score_svd_20pct_8213438`
- Prior (unconditional): job `8214018`, run `edm_prior_svd_8214018`

Case	Snapshot type	d_{95}	d_{99}	Participation ratio
70%	Sample (previous)	1	2	1.023
70%	Score (current)	63	137	2.566
20%	Sample (previous)	46	154	1.735
20%	Score (current)	293	464	8.786
Prior (unconditional)	Sample	47	158	1.812
Prior (unconditional)	Score	374	536	66.136

Table 3: Comparison of singular-spectrum intrinsic-dimensionality metrics for sample- vs score-based snapshots, including prior-only (no posterior sampling).

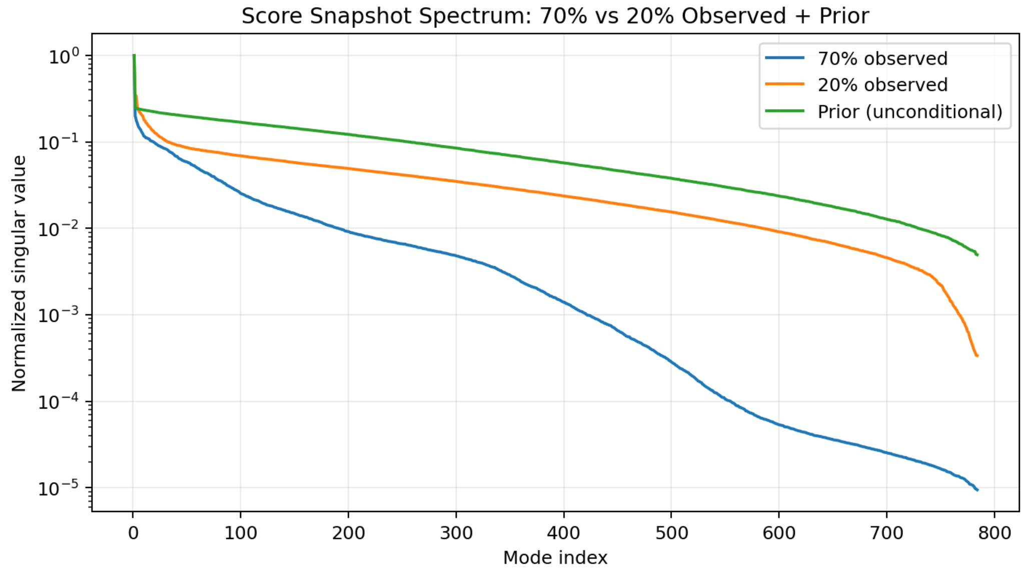

Figure 10: Normalized singular value spectrum for score snapshots: 70% observed, 20% observed, and prior-unconditional.

Additional Subsection: Randomized Top-20 Score Spectrum vs Visible Fraction

We ran an additional score-snapshot experiment for progressively increased visible-pixel fractions:

$$0\%,\ 20\%,\ 40\%,\ 60\%,\ 80\%.$$

For each fraction, we formed the score snapshot matrix

$$S = [s^{(1)}, \dots, s^{(N)}] \in \mathbb{R}^{784\times 1024},$$

and used a randomized SVD (instead of full SVD) to estimate only the leading 20 singular values. Given random test matrix Ω, we computed

$$Y = S\Omega, \quad Q = \mathrm{orth}(Y), \quad B = Q^\top S,$$

then took the leading singular values of B as approximations of the leading singular values of S.

Randomized sweep job on Gautschi:

- job 8214779, run `edm_randomized_score_svd_sweep_with_prior_8214779`

Because this is a truncated top-20 spectrum estimate, we use it here for comparative spectral shape only (not for intrinsic-dimension indicators).

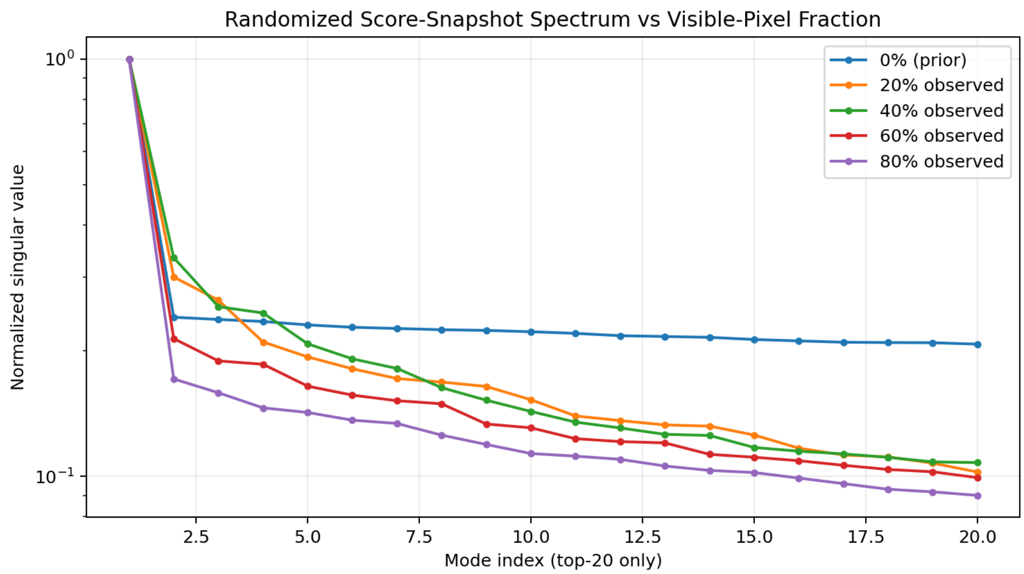

Figure 11: Randomized top-20 score-snapshot spectrum for visible fractions 0/20/40/60/80% (0% is prior-only sampling).

Additional Subsection: Sampling-Free Posterior Geometry ID

To quantify how partial observations change intrinsic dimensionality *without posterior sampling*, we used a local Gaussian approximation around a MAP point. For each observed mask, we estimated $\hat{x}_{\mathrm{MAP}}$ using score-guided ascent on:

$$\nabla_x \log p(x \mid y, M) \approx s_\theta(x, \sigma_{\mathrm{eval}}) \; + \; \frac{M^\top (y - Mx)}{\sigma_y^2},$$

then formed a local posterior precision proxy:

$$\Lambda(\hat{x}_{\mathrm{MAP}}) \approx -J_s(\hat{x}_{\mathrm{MAP}}, \sigma_{\mathrm{eval}}) \; + \; \frac{M^\top M}{\sigma_y^2},$$

where J_s is the Jacobian of the approximated score. Let the (nonnegative) precision eigenvalues be sorted in descending order:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{784} \geq 0.$$

We define normalized precision-energy weights

$$w_i = \frac{\lambda_i}{\sum_{j=1}^{784} \lambda_j},$$

and cumulative precision energy

$$E(k) = \sum_{i=1}^{k} w_i.$$

Then the codimension thresholds are computed as the *smallest* indices whose cumulative precision energy reaches the target level:

$$c_{95} = \min\{k : \; E(k) \geq 0.95\}, \qquad c_{99} = \min\{k : \; E(k) \geq 0.99\}.$$

Finally, we convert codimension to an intrinsic-dimension proxy by subtracting from ambient dimension 784:

$$d_{95}^{\text{proxy}} = 784 - c_{95}, \qquad d_{99}^{\text{proxy}} = 784 - c_{99}.$$

Geometry-ID jobs on Gautschi:

- 70% observed: job `8213702`, run `edm_geometry_id_70pct_8213702`
- 20% observed: job `8213703`, run `edm_geometry_id_20pct_8213703`

Observed fraction	c_{95}	d_{95}^{proxy}	c_{99}	d_{99}^{proxy}
70%	693	91	747	37
20%	654	130	724	60

Table 4: Sampling-free posterior-geometry intrinsic-dimension proxies from local precision spectra.

These sampling-free proxies show the expected trend: with fewer observed pixels (20%), the posterior is less constrained and has higher intrinsic dimensionality than the 70% observed case.

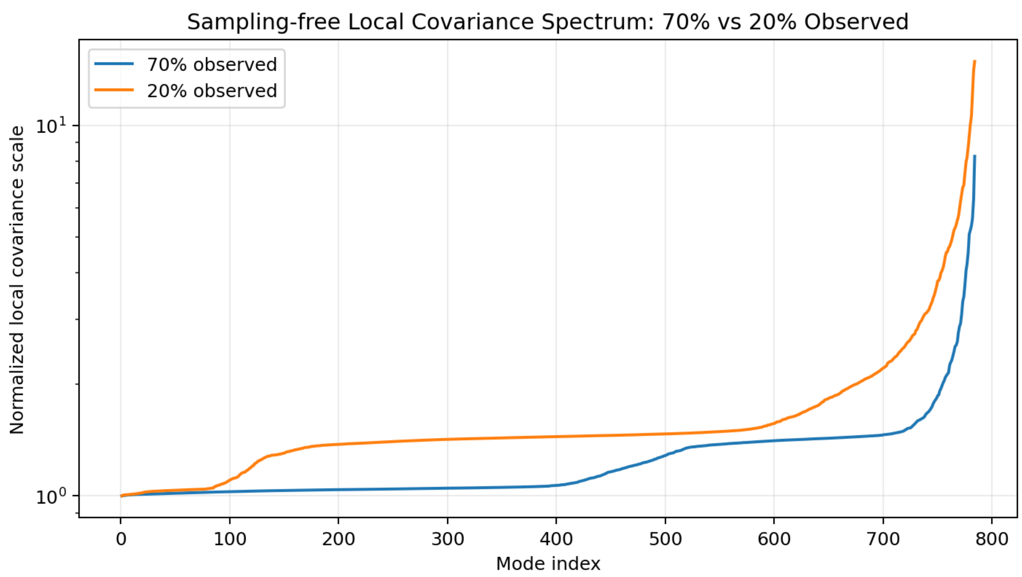

Figure 12: Sampling-free local covariance spectrum (MAP + score-Jacobian approximation): 70% observed vs 20% observed.

9 Live Dashboard Operation

The workflow now maintains a root live dashboard at:

- `outputs/dashboard.html`

This dashboard follows the latest run via:

- `outputs/LATEST_RUN.txt`

- `outputs/current` symlink

and supports:

- periodic refresh during active runs,
- automatic stop of refresh when status reaches `completed`,
- image zoom controls for plots/schematics.

10 Reproducibility Notes

Primary scripts used for these experiments:

- `mnist_diffusion.py`
- `posterior_svd_analysis.py`
- `posterior_score_svd_analysis.py`
- `posterior_score_randomized_svd_sweep.py`
- `prior_score_svd_analysis.py`
- `posterior_geometry_id_analysis.py`
- `submit_mnist_edm_posterior.slurm`
- `submit_mnist_edm_posterior_20pct.slurm`
- `serve_dashboard.sh`

Persistent SSH Connection Reuse

To avoid repeated password + 2FA prompts for each `ssh`/`scp` call, use SSH multiplexing with a persistent control socket:

```
ssh -M -S /tmp/gautschi_mux.sock -o ControlPersist=8h -N \
  rmaulik@gautschi.rcac.purdue.edu
```

Then reuse that authenticated channel:

```
ssh -S /tmp/gautschi_mux.sock rmaulik@gautschi.rcac.purdue.edu
scp -o ControlMaster=auto -o ControlPath=/tmp/gautschi_mux.sock \
  <src> <dst>
```

Check and close the master session:

```
ssh -S /tmp/gautschi_mux.sock -O check rmaulik@gautschi.rcac.purdue.edu
ssh -S /tmp/gautschi_mux.sock -O exit  rmaulik@gautschi.rcac.purdue.edu
```

To regenerate this report PDF locally:

```
cd PurdueHPC_Codex/report
latexmk -pdf -interaction=nonstopmode -halt-on-error \
  -output-directory=../output/pdf gautschi_diffusion_report.tex
```