Generating Images by Diffusion (continued)

ECE 57000 — November 11, 2026

David I. Inouye

Known corruption lets us learn a denoising task

- November 9 completed VAE training: reconstruction and prior regularization train both networks.
- Diffusion returns to **denoising**: predict a clean image from its corrupted version.
- At fixed noise scale, predicting the clean image or its added noise gives equivalent targets.

Today: learn noisy scores, connect several noise levels, then view generation as solving an SDE or ODE.

Predicting the clean image and predicting its noise are equivalent

For $\widetilde{\boldsymbol{x}}_\sigma = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}$ at a **fixed** $\sigma > 0$:

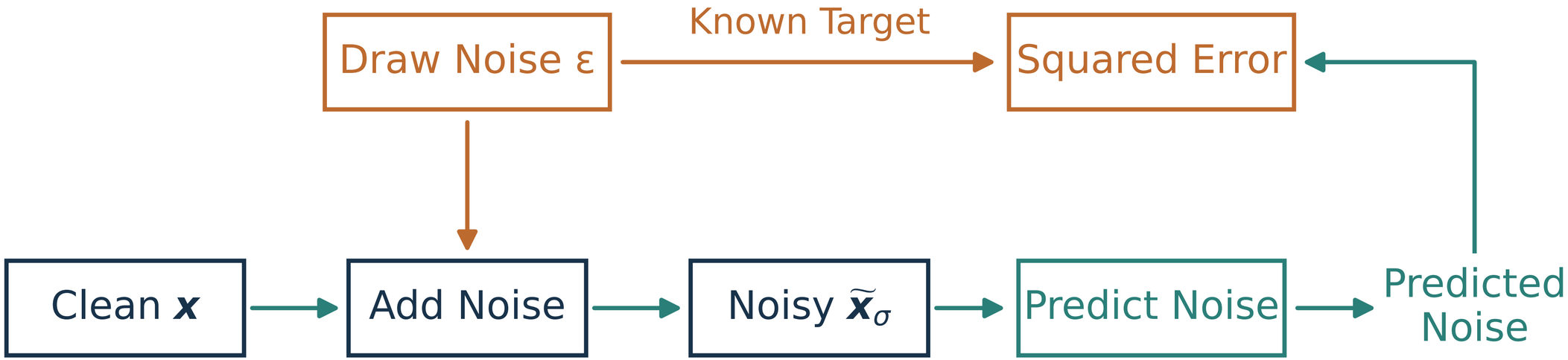
$$\widehat{\boldsymbol{x}} = \widetilde{\boldsymbol{x}}_\sigma - \sigma\epsilon_{\theta,\sigma}(\widetilde{\boldsymbol{x}}_\sigma), \qquad \hat{\boldsymbol{\epsilon}} = \frac{\widetilde{\boldsymbol{x}}_\sigma - D_{\theta,\sigma}(\widetilde{\boldsymbol{x}}_\sigma)}{\sigma}.$$

The squared errors differ by a constant:

$$\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2 = \sigma^2\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}\|^2.$$

Same prediction problem; a different output parameterization.

The noise we draw becomes a supervised target



$$J(\theta) = \mathbb{E}_{\mathbf{x},\boldsymbol{\epsilon}} \left[\|\boldsymbol{\epsilon} - \epsilon_{\theta,\sigma}(\boldsymbol{x} + \sigma\boldsymbol{\epsilon})\|^2 \right].$$

No noise labels need to be collected: **we constructed the corruption**.

One minibatch update learns noise prediction at one level

Fix $\sigma > 0$. For each batch of B clean images:

Step	Computation, $i = 1, \dots, B$
Draw noise	$\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, I)$ independently
Corrupt	$\widetilde{\boldsymbol{x}}_{\sigma,i} = \boldsymbol{x}_i + \sigma\boldsymbol{\epsilon}_i$
Predict	$\hat{\boldsymbol{\epsilon}}_i = \epsilon_{\theta,\sigma}(\widetilde{\boldsymbol{x}}_{\sigma,i})$
Average loss	$\widehat{J} = \frac{1}{B}\sum_i \lVert\boldsymbol{\epsilon}_i - \hat{\boldsymbol{\epsilon}}_i\rVert^2$
Update	$\theta \leftarrow \theta - \eta\nabla_\theta \widehat{J}$

Can this one denoiser also turn random noise into a clean image?

From denoising to noisy scores

What the predictor learns—and why one noise level is insufficient

Denoising learns the score of the noisy distribution

$$\widetilde{\boldsymbol{x}}_\sigma = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}, \qquad p_\sigma = p_{\text{data}} * \mathcal{N}(\mathbf{0}, \sigma^2 I).$$

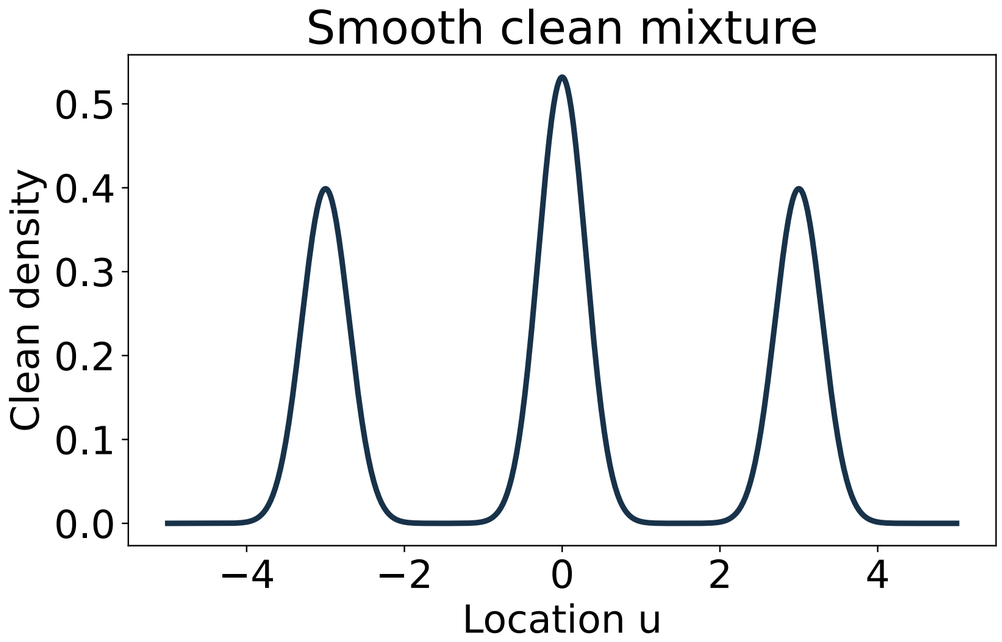
$*$ denotes **convolution**: adding independent Gaussian noise smooths the clean-data distribution.

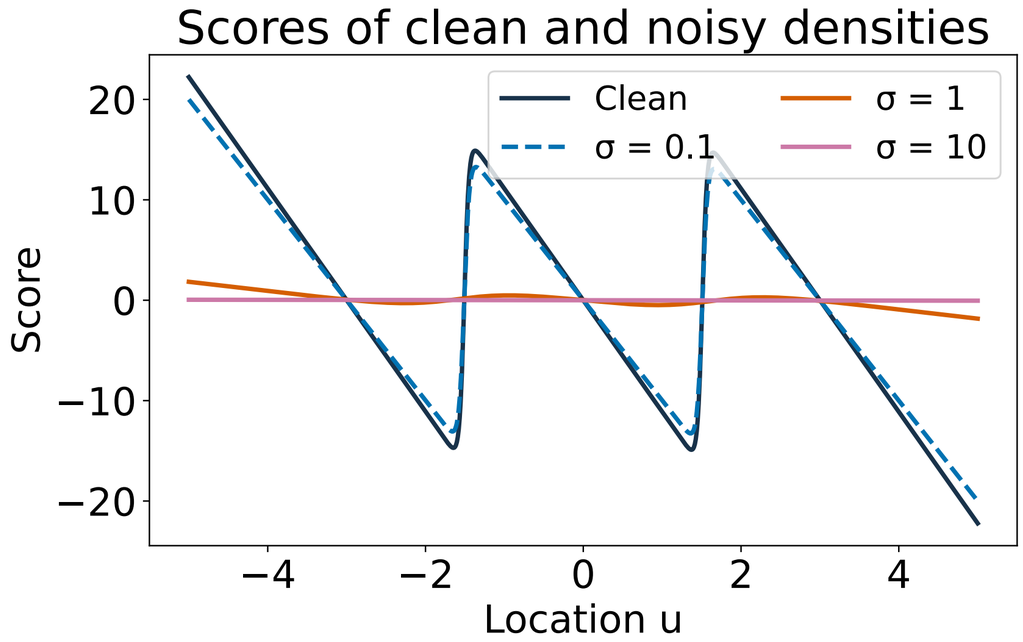
At the noisy input $\widetilde{\boldsymbol{x}}_\sigma$, the score points toward increasing **noisy-data log density**:

$$s_\sigma(\widetilde{\boldsymbol{x}}_\sigma) = \nabla_{\widetilde{\boldsymbol{x}}_\sigma} \log p_\sigma(\widetilde{\boldsymbol{x}}_\sigma) = -\frac{\epsilon^*_\sigma(\widetilde{\boldsymbol{x}}_\sigma)}{\sigma} \quad \approx \quad -\frac{\epsilon_{\theta,\sigma}(\widetilde{\boldsymbol{x}}_\sigma)}{\sigma}.$$

Exact for the ideal squared-error predictor; approximate for the trained network.

Small noise approaches the clean score where that score exists

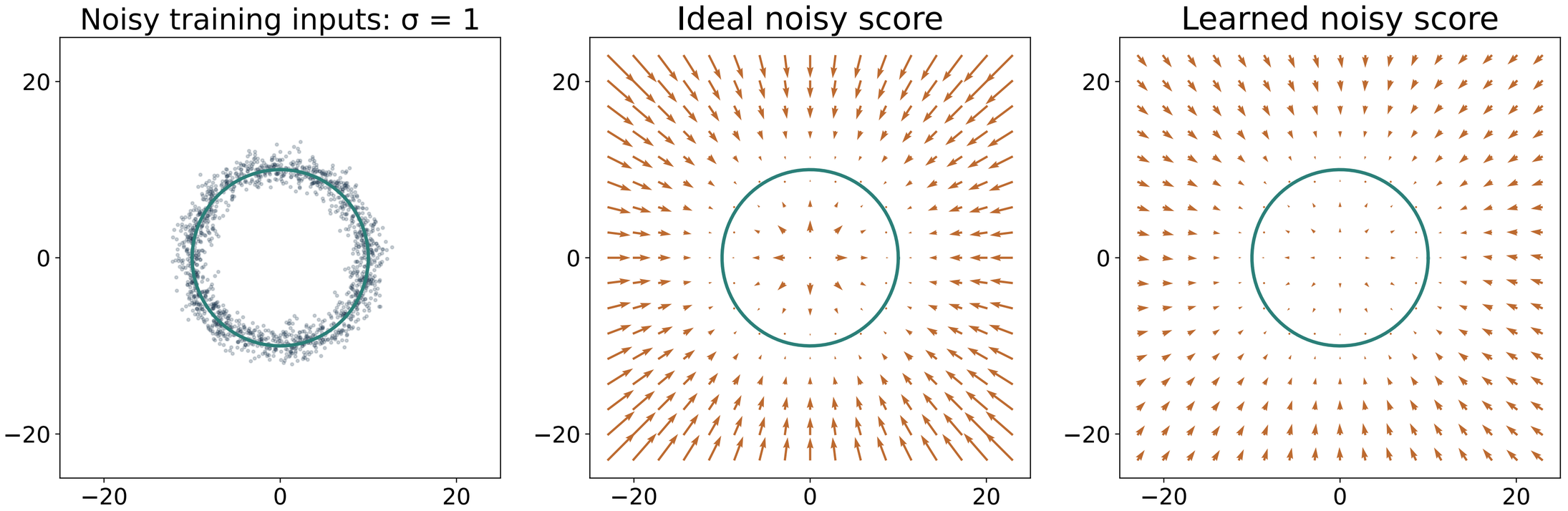




For a smooth positive clean density, under suitable regularity: $\nabla \log p_\sigma(\widetilde{\boldsymbol{x}}_\sigma) \to \nabla \log p_{\text{data}}(\widetilde{\boldsymbol{x}}_\sigma)$ as $\sigma \to 0$.

Small-noise training leaves poorly constrained scores away from data

Clean data: uniform on a radius-10 circle. Train a small MLP at $\sigma = 1$.



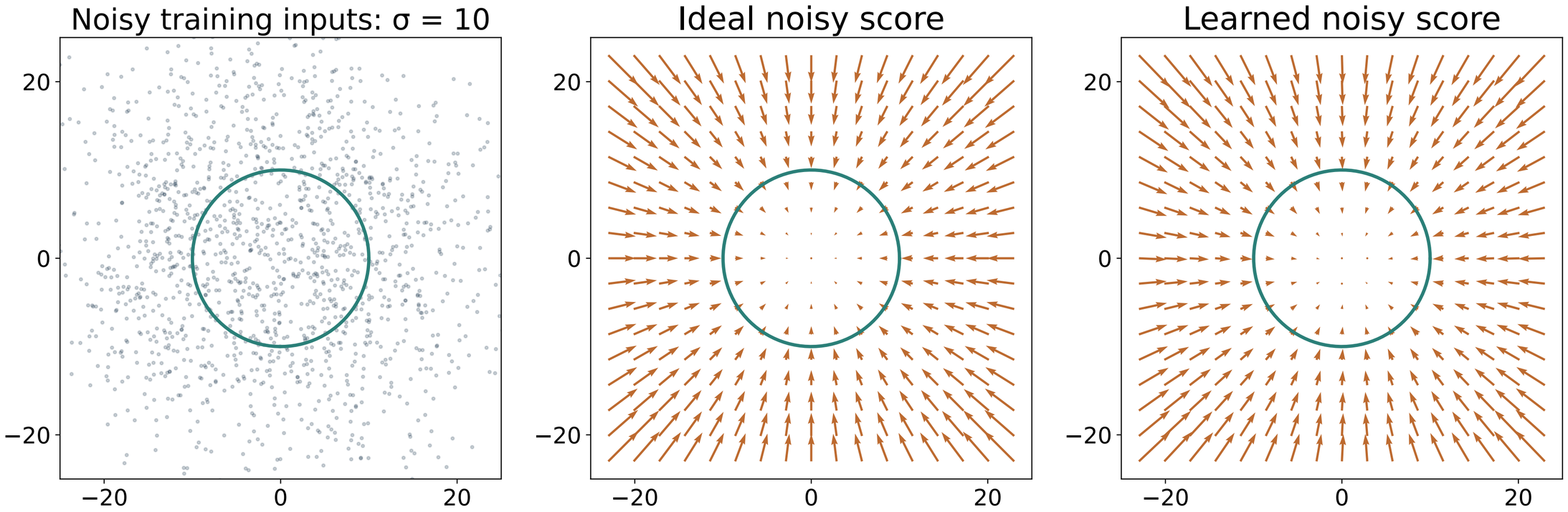
Relative score error: ring band 6% • interior 60% • exterior 53%

Arrow length = score magnitude × 0.2 (both panels) • Teal circle: clean data • Seed 5720

Measured failure here: wrong magnitudes off the ring; directions mostly remain radial.

Large noise improves coverage but changes the score being learned

Same architecture and training budget, now $\sigma = 10$.



Relative score error: ring band 12% • interior 29% • exterior 4%

Arrow length = score magnitude × 20 (both panels) • Teal circle: clean data • Seed 5720

A good estimate of $\nabla \log p_{10}$ is still not the clean-data score.

What would perfect single-level sampling produce?

Suppose the $\sigma = 10$ score is exact and the sampler correctly targets its density.
What would more accurate sampling recover?
A. The clean radius-10 circle
B. The broad noisy distribution p_{10}
C. One conditional-mean point

B. Better approximation cannot change the distribution the score describes.

Learning across noise levels

Broad coverage first; clean-data detail at lower noise

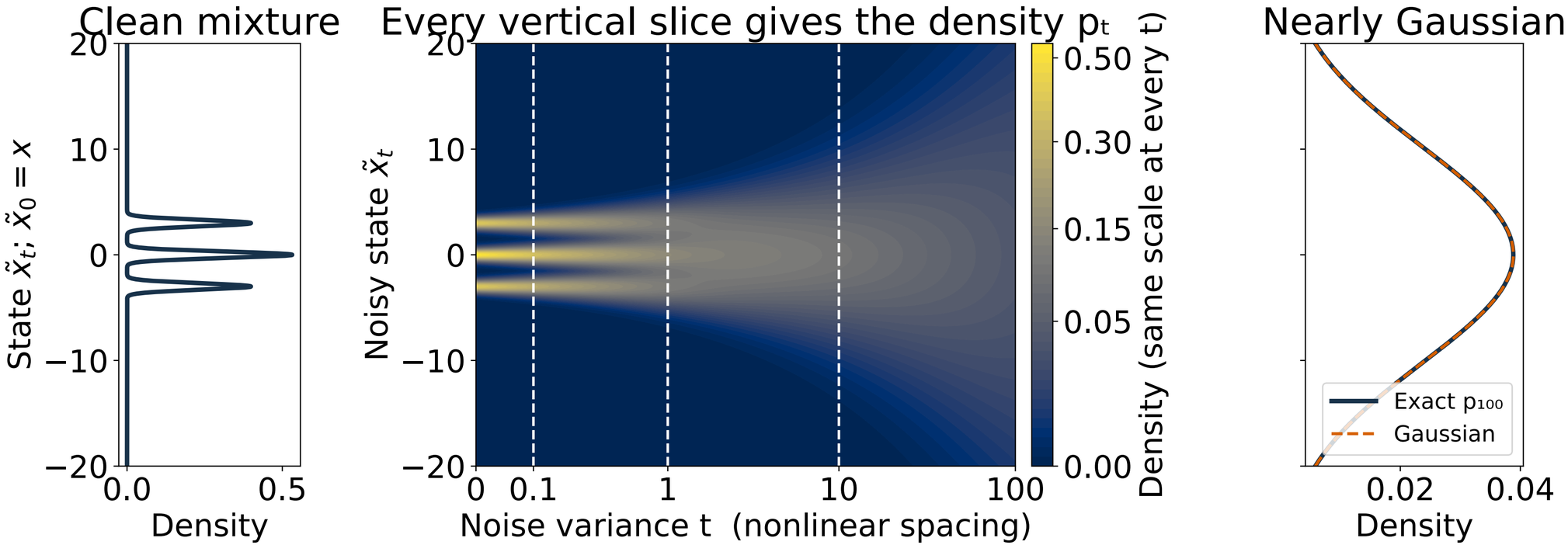
Multiple noise levels connect broad coverage to fine detail

Noisy scores → discrete noise levels → continuous SDE/ODE sampling

Now t denotes noise variance: $\sigma_t = \sqrt{t}, \widetilde{\boldsymbol{x}}_t = \boldsymbol{x} + \sqrt{t}\,\boldsymbol{\epsilon}$, and $\widetilde{\boldsymbol{x}}_0 = \boldsymbol{x}$.

1. Large noise: learn directions across a broad region.
2. Intermediate noise: change the target toward cleaner structure.
3. Small noise: resolve detail near the clean data.

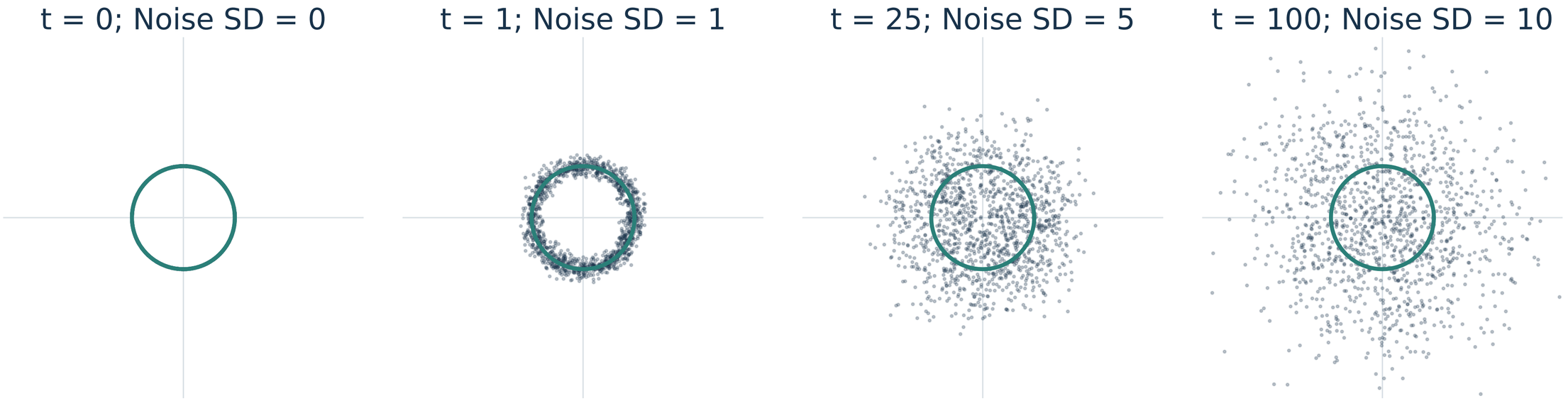
Every time slice is a distribution over possible noisy samples



$$p_t(\tilde{x}_t) = \sum_{j=1}^{3} w_j \, \mathcal{N}(\tilde{x}_t; \mu_j, \, 0.09 + t), \qquad (\mu_1, \mu_2, \mu_3) = (-3, 0, 3).$$

Independent standard-normal steps make noise variance add

$$\widetilde{\boldsymbol{x}}_t = \widetilde{\boldsymbol{x}}_{t-1} + \boldsymbol{\eta}_t, \qquad \boldsymbol{\eta}_t \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, I).$$



Same Axis Scale in Every Panel; Each Step Adds Independent N(0, I) Noise

After t steps: **noise variance t; noise standard deviation $\sqrt{t}$.**

Any noisy training input needs only one Gaussian draw

$$p_t(\widetilde{\mathbf{x}}_t \mid \boldsymbol{x}) = \mathcal{N}(\widetilde{\mathbf{x}}_t; \boldsymbol{x}, tI), \qquad \widetilde{\boldsymbol{x}}_t = \boldsymbol{x} + \sqrt{t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I).$$

- Independent Gaussian variances add: $I + \cdots + I = tI$.
- Select t and construct its noisy input directly.
- For large T, the noise dominates: start generation approximately from $\mathcal{N}(\mathbf{0}, TI)$.

One network learns noise prediction at every level

Add the noise level to the input: $\hat{\boldsymbol{\epsilon}}_i = \epsilon_\theta(\widetilde{\boldsymbol{x}}_{t_i}, t_i)$.

Step	Computation for each image in a batch
Draw level and noise	$t_i \sim \mathrm{Uniform}\{1, \dots, T\}, \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, I)$
Construct input	$\widetilde{\boldsymbol{x}}_{t_i} = \boldsymbol{x}_i + \sqrt{t_i}\,\boldsymbol{\epsilon}_i$
Predict and score	$\widehat{J} = \frac{1}{B}\sum_i \lVert \boldsymbol{\epsilon}_i - \epsilon_\theta(\widetilde{\boldsymbol{x}}_{t_i}, t_i)\rVert^2$
Update shared parameters	$\theta \leftarrow \theta - \eta \nabla_\theta \widehat{J}$

One sampled level per image; the same parameters across levels.

Reverse generation needs small conditional steps, not one clean guess

A clean-image estimate uses the whole predicted noise:

$$\widehat{\boldsymbol{x}} = \widetilde{\boldsymbol{x}}_t - \sqrt{t}\,\epsilon_\theta(\widetilde{\boldsymbol{x}}_t, t).$$

For the additive walk, the **previous-step conditional mean** instead uses

$$\widehat{\boldsymbol{\mu}}_{t-1} = \widetilde{\boldsymbol{x}}_t - \frac{1}{\sqrt{t}}\epsilon_\theta(\widetilde{\boldsymbol{x}}_t, t).$$

Generation follows $T \to T-1 \to \cdots \to 0$ with fixed learned parameters.

A mean is not a full distribution: a reverse sampler must also handle uncertainty.

A stochastic reverse step denoises, then adds fresh noise

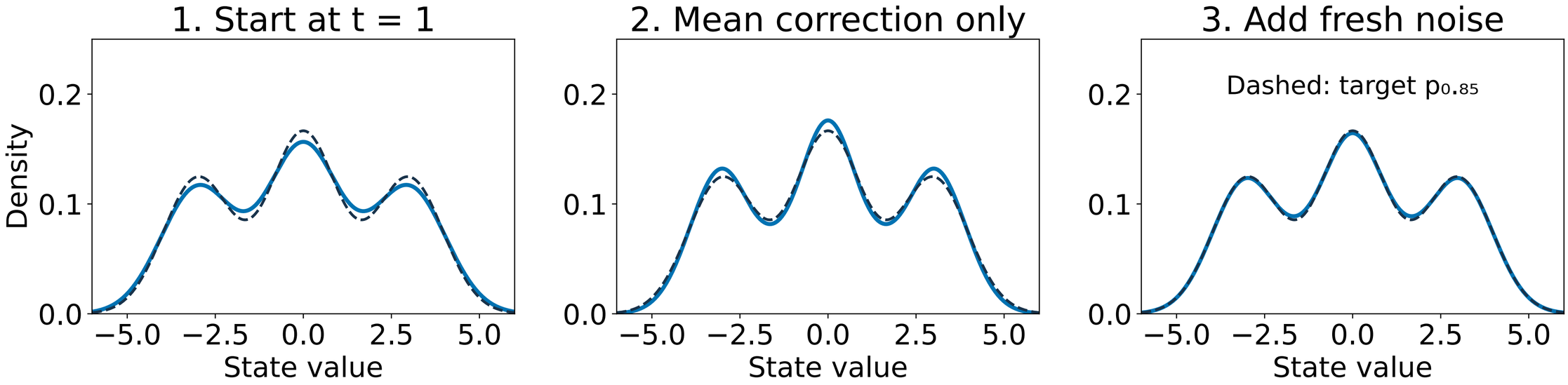
Move from current noise variance t to a lower variance $s \geq 0$. The **step size** is $h = t - s > 0$: how much we reduce the noise variance.

A **DDPM-style step in our additive units** is

$$\tilde{x}_s = \underbrace{\tilde{x}_t - \frac{h}{\sqrt{t}}\epsilon_\theta(\tilde{x}_t, t)}_{m\text{: previous-level mean}} + \underbrace{\sqrt{\frac{sh}{t}}\,\xi}_{\text{fresh noise}}, \qquad \xi \sim \mathcal{N}(0, 1).$$

- The correction uses h/t of the predicted clean-data correction.
- For small h relative to t, the added variance is approximately h.
- This Gaussian step **approximates** the reverse conditional distribution.

Mean correction narrows the distribution; noise spreads it again



Local Gaussian intuition: variance $v \to v - 2h \to v - h$ (first order in h)

“Denoise two, add back one” describes local variance change—not a time reversal.

Diffusion as a continuous process

From Gaussian steps to SDEs and continuous-time training

Continuous noising is the small-step limit of the Gaussian walk

Discrete diffusion → continuous SDE → probability-flow ODE → solver choices

$$d\widetilde{\boldsymbol{x}}_t = d\mathbf{w}_t, \qquad \widetilde{\boldsymbol{x}}_0 = \boldsymbol{x}.$$

- An **SDE** adds random increments: $\Delta\mathbf{w} \sim \mathcal{N}(\mathbf{0}, hI)$ over an interval h.
- Our time label is noise variance: $\sigma(t) = \sqrt{t}$.
- Its marginal remains $\widetilde{\boldsymbol{x}}_t = \boldsymbol{x} + \sqrt{t}\,\boldsymbol{\epsilon}$ in distribution.