On Diffusion Posterior Sampling

Matias G. Delgadino
August 6, 2026

j.w.w. Will Porteous (UT Math), Sanjay Shakkottai (UT EE),
Advait Parulekar (UT EE), Sebastian Motsch (ASU Math)

Motivation

Diffusion models are one of the most successful algorithms to generate high quality samples from complex distributions.

Motivation

Diffusion models are one of the most successful algorithms to generate high quality samples from complex distributions.

We are all aware of image generation, but these advances have also been applied to protein synthesis, text generation, and many other fields.

Set up

We have access to data:

$$\{X_i\}_{i=1}^{N} \subset \mathbb{R}^K.$$

The internet provides us with a huge amount of high resolution cat images.

Set up

We have access to data:

$$\{X_i\}_{i=1}^{N} \subset \mathbb{R}^K.$$

The internet provides us with a huge amount of high resolution cat images.

How do we create a model that can generate new cat images?

Set up

We have access to data:

$$\{X_i\}_{i=1}^N \subset \mathbb{R}^K.$$

The internet provides us with a huge amount of high resolution cat images.

How do we create a model that can generate new cat images?

We assume that the data are i.i.d. samples of some unknown underlying **target distribution**

$$X_i \sim \rho_* \in \mathcal{P}(\mathbb{R}^K).$$

Dynamic interpretation of sampling

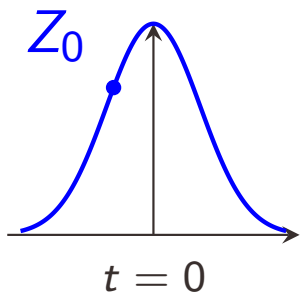
We are interested in finding a dynamic process that converts a sample

$$Z \sim \gamma = \mathcal{N}(0, I)$$

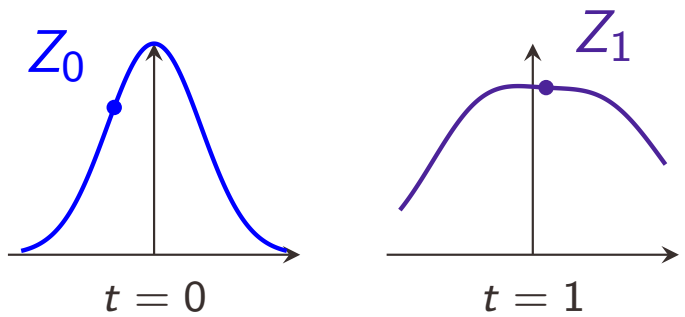
into a sample from the target distribution

$$X \sim \rho_*.$$

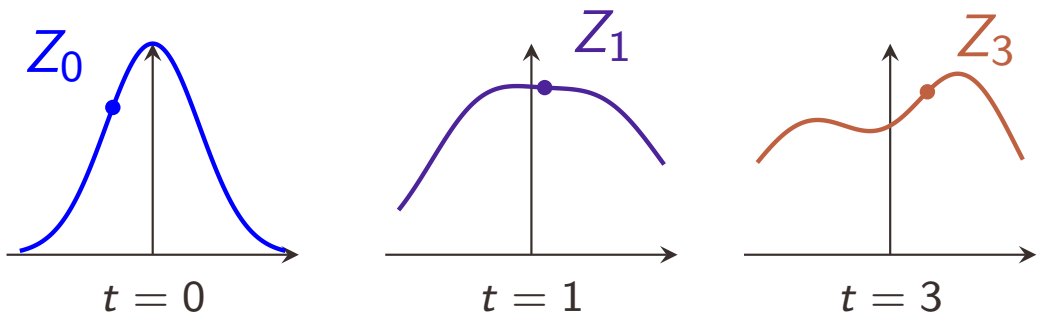
Time-Evolution of the distribution



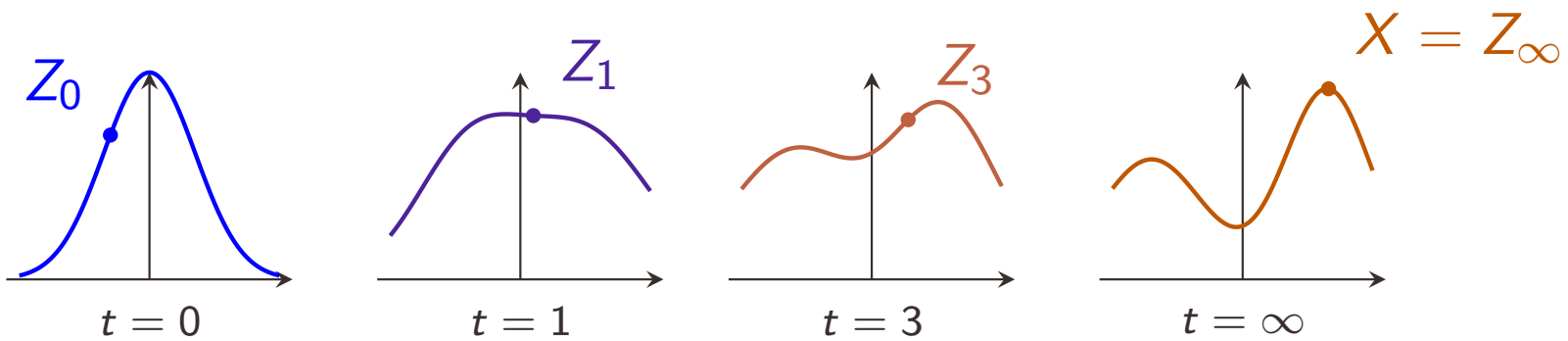
Time-Evolution of the distribution



Time-Evolution of the distribution



Time-Evolution of the distribution



The Sampling Problem

The problem is to find a vector field

$$V : [0, \infty) \times \mathbb{R}^K \to \mathbb{R}^K$$

such that Z_t evolves according to an SDE of the form

$$dZ_t = V(t, Z_t)dt + \sqrt{2}dB_t,$$

The Sampling Problem

The problem is to find a vector field

$$V : [0, \infty) \times \mathbb{R}^K \to \mathbb{R}^K$$

such that Z_t evolves according to an SDE of the form

$$dZ_t = V(t, Z_t)dt + \sqrt{2}dB_t,$$

and

$$Law(Z_\infty) \sim \rho_*.$$

The forward process: Ornstein-Uhlenbeck

It is much easier to go the other way. Namely, noise ρ_* into a Gaussian.

The forward process: Ornstein-Uhlenbeck

It is much easier to go the other way. Namely, noise ρ_* into a Gaussian.

The Ornstein-Uhlenbeck (O-U) process:

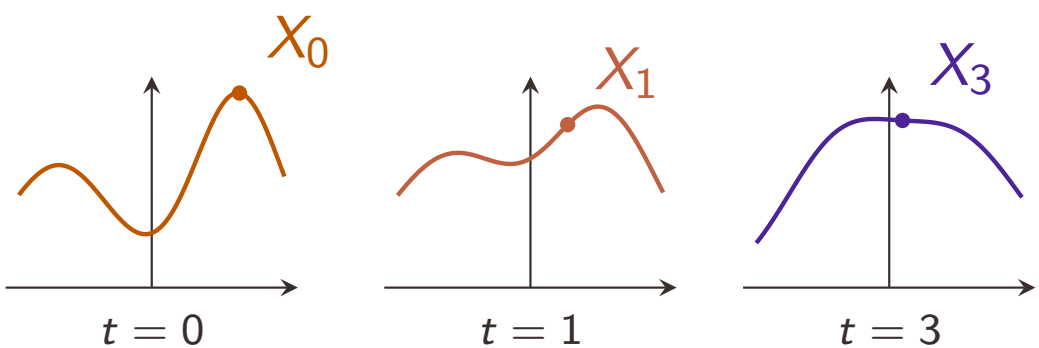
$$dX_t = -X_t dt + \sqrt{2} dB_t,$$

The forward process: Ornstein-Uhlenbeck

It is much easier to go the other way. Namely, noise ρ_* into a Gaussian.

The Ornstein-Uhlenbeck (O-U) process:

$$dX_t = -X_t dt + \sqrt{2} dB_t,$$

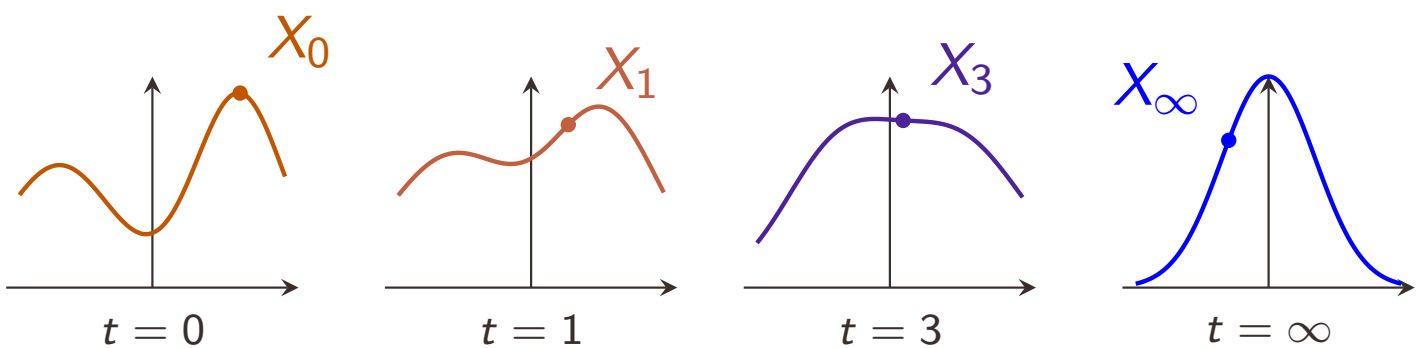


The forward process: Ornstein-Uhlenbeck

It is much easier to go the other way. Namely, noise ρ_* into a Gaussian.

The Ornstein-Uhlenbeck (O-U) process:

$$dX_t = -X_t dt + \sqrt{2} dB_t,$$



Fokker-Planck equation

More precisely, the evolution of the law of $X_t \sim \overrightarrow{\rho}_t$ is given by the Fokker-Planck equation

$$\begin{cases} \partial_t \overrightarrow{\rho}_t = \nabla \cdot (\overrightarrow{\rho}_t x) + \Delta \overrightarrow{\rho}_t, \\ \overrightarrow{\rho}_0 = \rho_*. \end{cases}$$

Convergence Estimate

Exponential Convergence

$$\mathcal{H}(\overrightarrow{\rho}_t \mid \gamma) \leq Ce^{-t}.$$

for some constant C depending on the second moment of ρ_*, where $\mathcal{H}$ is the relative entropy.

Convergence Estimate

Exponential Convergence

$$\mathcal{H}(\overrightarrow{\rho}_t \mid \gamma) \leq Ce^{-t}.$$

for some constant C depending on the second moment of ρ_*, where $\mathcal{H}$ is the relative entropy.

This estimate is independent of the regularity of ρ_*!

The backward process

We chose a time horizon $T > 0$, such that

$$\overrightarrow{\rho}_T \approx \gamma.$$

The backward process

We chose a time horizon $T > 0$, such that

$$\overrightarrow{\rho}_T \approx \gamma.$$

In practice $T = 5$.

The backward process

We chose a time horizon $T > 0$, such that

$$\overrightarrow{\rho}_T \approx \gamma.$$

In practice $T = 5$.
We consider the equation satisfied by the time-reversal of the law:

$$\overleftarrow{\rho}_t = \overrightarrow{\rho}_{T-t}.$$

The wrong backward process

WARNING

If we are not careful, using the standard Fokker-Planck equation

$$\begin{cases} \partial_t \overrightarrow{\rho}_t = \Delta \overrightarrow{\rho}_t + \nabla \cdot (\overrightarrow{\rho}_t x), \\ \overrightarrow{\rho}_0 = \rho_*, \end{cases}$$

we end up with the reverse heat equation

$$\begin{cases} \partial_t \overleftarrow{\rho}_t = -\Delta \overleftarrow{\rho}_t - \nabla \cdot (\overleftarrow{\rho}_t x), \\ \overleftarrow{\rho}_0 = \overrightarrow{\rho}_T. \end{cases}$$

The trick

We can re-write the Laplacian as a transport term:

$$\begin{aligned}\partial_t \overrightarrow{\rho} &= \Delta \overrightarrow{\rho} + \nabla \cdot (x \overrightarrow{\rho}) \\ &= -\Delta \overrightarrow{\rho} + 2\Delta \overrightarrow{\rho} + \nabla \cdot (x \overrightarrow{\rho}) \\ &= -\Delta \overrightarrow{\rho} + 2\nabla \cdot (\overrightarrow{\rho} \nabla \log \overrightarrow{\rho}) + \nabla \cdot (x \overrightarrow{\rho})\end{aligned}$$

The trick

We can re-write the Laplacian as a transport term:

$$\begin{aligned}\partial_t \overrightarrow{\rho} &= \Delta \overrightarrow{\rho} + \nabla \cdot (x \overrightarrow{\rho}) \\ &= -\Delta \overrightarrow{\rho} + 2\Delta \overrightarrow{\rho} + \nabla \cdot (x \overrightarrow{\rho}) \\ &= -\Delta \overrightarrow{\rho} + 2\nabla \cdot (\overrightarrow{\rho} \nabla \log \overrightarrow{\rho}) + \nabla \cdot (x \overrightarrow{\rho})\end{aligned}$$

Reversing time, we get

$$\begin{cases} \partial_t \overleftarrow{\rho}_t = \Delta \overleftarrow{\rho}_t - 2\nabla \cdot (\overleftarrow{\rho}_t \nabla \log \overrightarrow{\rho}_{T-t}) - \nabla \cdot (x \overleftarrow{\rho}_t), \\ \overleftarrow{\rho}_0 = \overrightarrow{\rho}_T \sim \gamma. \end{cases}$$

Score Matching

Hence, we are left with finding a vector field V such that

$$\inf_{\Theta} \int_0^T \int_{\mathbb{R}^K} |V_\Theta(t,x) - \nabla \log \overrightarrow{\rho}_{T-t}(x)|^2 d\overrightarrow{\rho}_t dt \approx 0.$$

Why Score Matching Suffices

In fact, we have the quantitative bound

Error Estimate

Differentiating the relative entropy, we can show that

$$\mathcal{H}(\rho_*|\overleftarrow{\rho}^V_T) \leq \mathcal{H}(\overrightarrow{\rho}_T|\gamma) + \int_0^T \int_{\mathbb{R}^K} |V(t,x) - \nabla \log \overrightarrow{\rho}_{T-t}(x)|^2 d\overrightarrow{\rho}_t dt,$$

where $\overleftarrow{\rho}^V_t$ is the solution of the Fokker-Planck equation

$$\begin{cases} \partial_t \overleftarrow{\rho}^V_t = \Delta \overleftarrow{\rho}^V_t + 2\nabla \cdot (V \overleftarrow{\rho}^V_t) + \nabla \cdot (x \overleftarrow{\rho}^V_t) \\ \overleftarrow{\rho}^V_0 = \gamma. \end{cases}$$

The score function

In our set up, we have access to samples $\{X_0^{(i)}\}_{i=1}^N$ from ρ_*.

The score function

In our set up, we have access to samples $\{X_0^{(i)}\}_{i=1}^N$ from ρ_*.

No functional representation of ρ_* is known a-priori!

The score function

In our set up, we have access to samples $\{X_0^{(i)}\}_{i=1}^N$ from ρ_*.

No functional representation of ρ_* is known a-priori!

However, we can produce samples from $\overrightarrow{\rho}_t$ cheaply. In particular, we have the explicit formula

$$X_t = e^{-t} X_0^i + \sqrt{1 - e^{-2t}} Z \sim \overrightarrow{\rho}_t$$

where

$$i \sim \text{uniform}\{1, \dots, N\}, \quad Z \sim \gamma.$$

Tweedie's formula

Using linearity of the equation, we can guess where we came from by looking at the current position and the time. More precisely, we have the formula

Tweedie's formula

$$\mathbb{E}[X_0|X_t = x] = e^t x + (e^t - e^{-t})\nabla \log \overrightarrow{\rho}_t(x).$$

Tweedie's formula

Using linearity of the equation, we can guess where we came from by looking at the current position and the time. More precisely, we have the formula

Tweedie's formula

$$\mathbb{E}[X_0|X_t = x] = e^t x + (e^t - e^{-t})\nabla \log \overrightarrow{\rho}_t(x).$$

We have a stochastic estimator of the score function $\nabla \log \overrightarrow{\rho}_t$ given by

$$\nabla \log \overrightarrow{\rho}_t(X_t) \approx \frac{1}{e^t - e^{-t}}(X_0 - e^t X_t).$$

Tweedie's formula

Using linearity of the equation, we can guess where we came from by looking at the current position and the time. More precisely, we have the formula

Tweedie's formula

$$\mathbb{E}[X_0|X_t = x] = e^t x + (e^t - e^{-t})\nabla \log \overrightarrow{\rho}_t(x).$$

We have a stochastic estimator of the score function $\nabla \log \overrightarrow{\rho}_t$ given by

$$\nabla \log \overrightarrow{\rho}_t(X_t) \approx \frac{1}{e^t - e^{-t}}(X_0 - e^t X_t).$$

Sampling the initial condition X_0 from the data, the noise Z, and times, we can get an unbiased estimator of the score function.

Conditional Sampling

Now that we have understood how to sample a new picture of a cat, how do we sample a cat with specific features?

Conditional Sampling

Now that we have understood how to sample a new picture of a cat, how do we sample a cat with specific features? For instance, how do we sample a black cat?

Conditional Sampling

Now that we have understood how to sample a new picture of a cat, how do we sample a cat with specific features? For instance, how do we sample a black cat?

An option is always to do rejection sampling, but this can take very long depending on the rarity of the feature.

For instance, calico (three-colored) cats are common, but only about 1 in 3000 calicos is male.

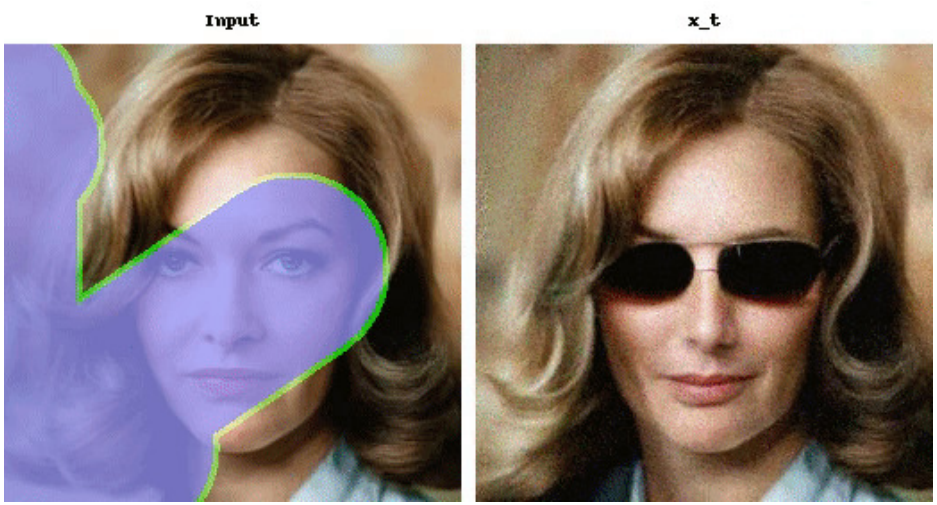
Conditional Sampling

Now that we have understood how to sample a new picture of a cat, how do we sample a cat with specific features? For instance, how do we sample a black cat?

An option is always to do rejection sampling, but this can take very long depending on the rarity of the feature.

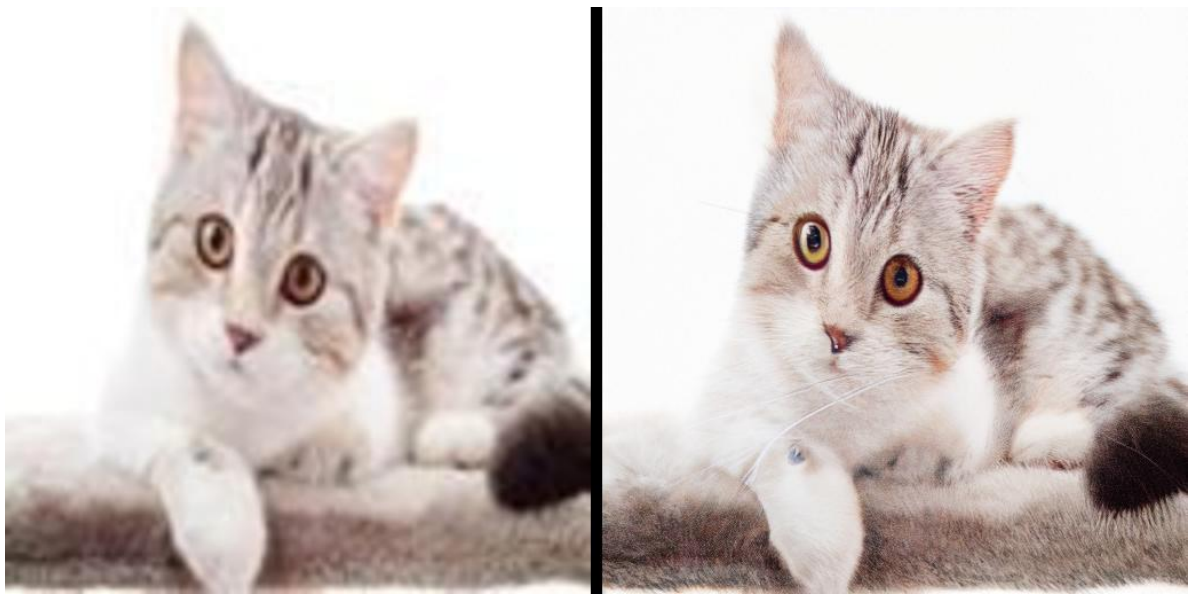
For instance, calico (three-colored) cats are common, but only about 1 in 3000 calicos is male. Finding a male calico by rejection sampling would take tens of thousands of samples.

Inpainting

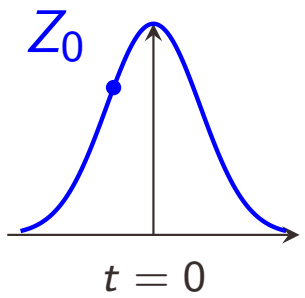


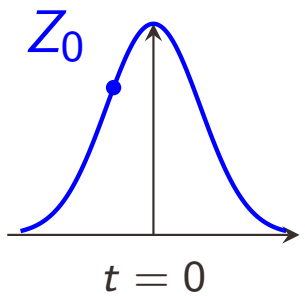
Super-resolution

Star Trek: The Next Generation style "Enhance!".



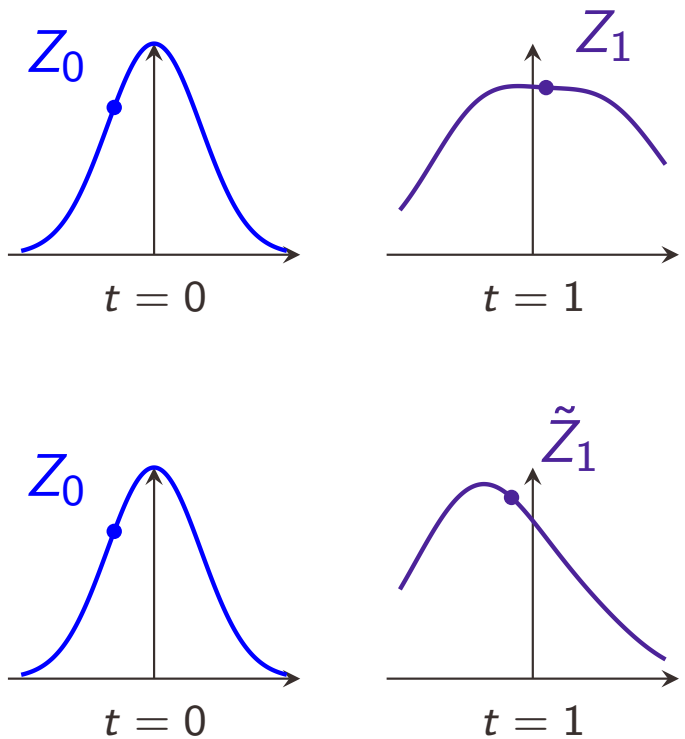
Conditional Dynamics: Mode Selection





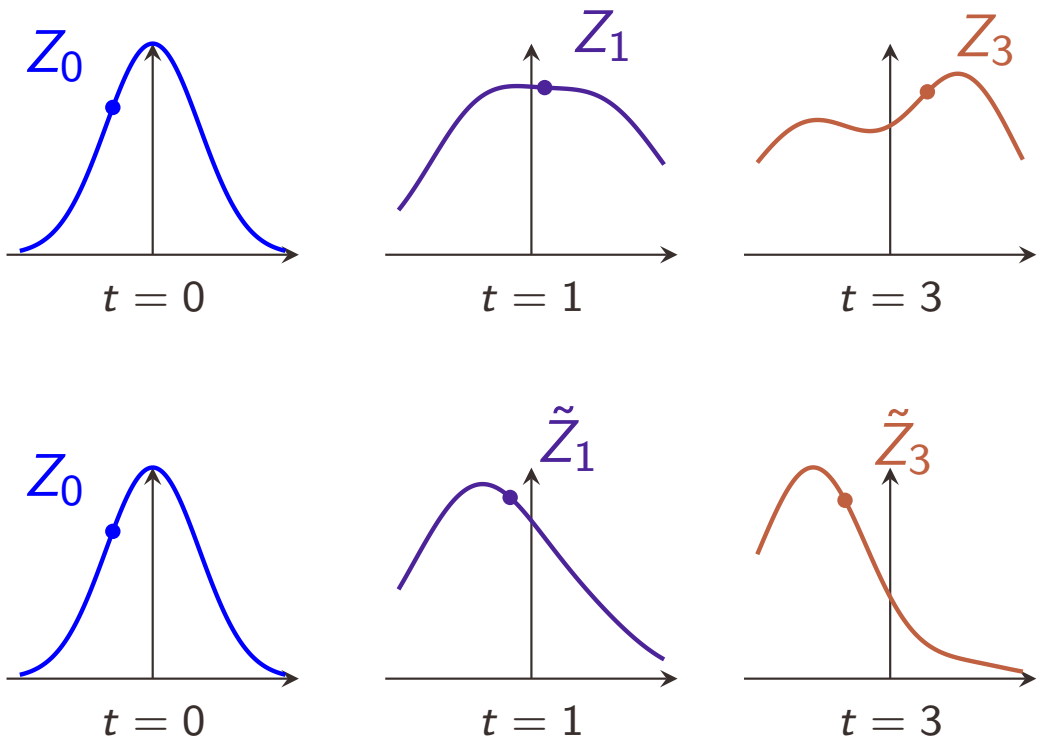
$$\tilde{Z}_\infty \sim \rho_*(\cdot \mid \text{left well})$$

Conditional Dynamics: Mode Selection



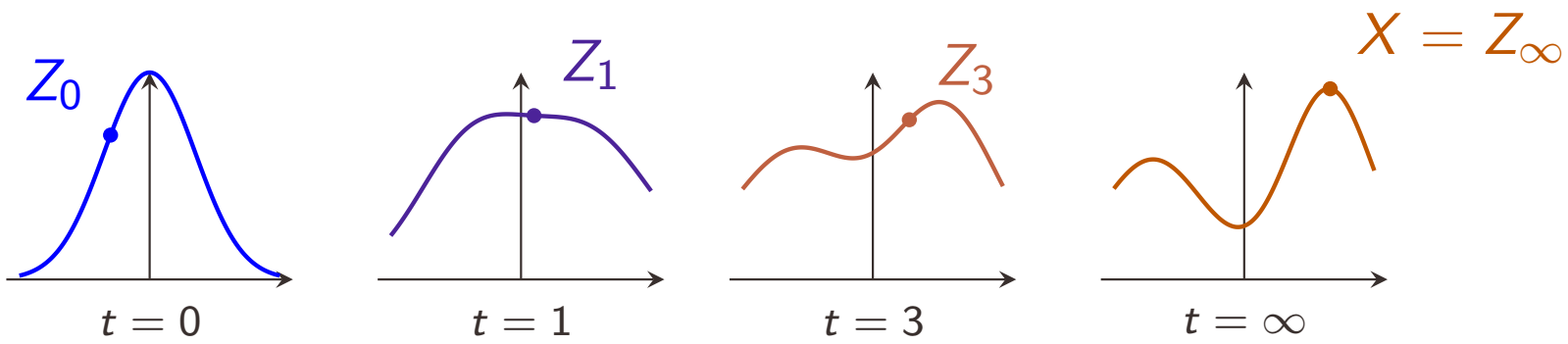
$$\tilde{Z}_\infty \sim \rho_*(\cdot \mid \text{left well})$$

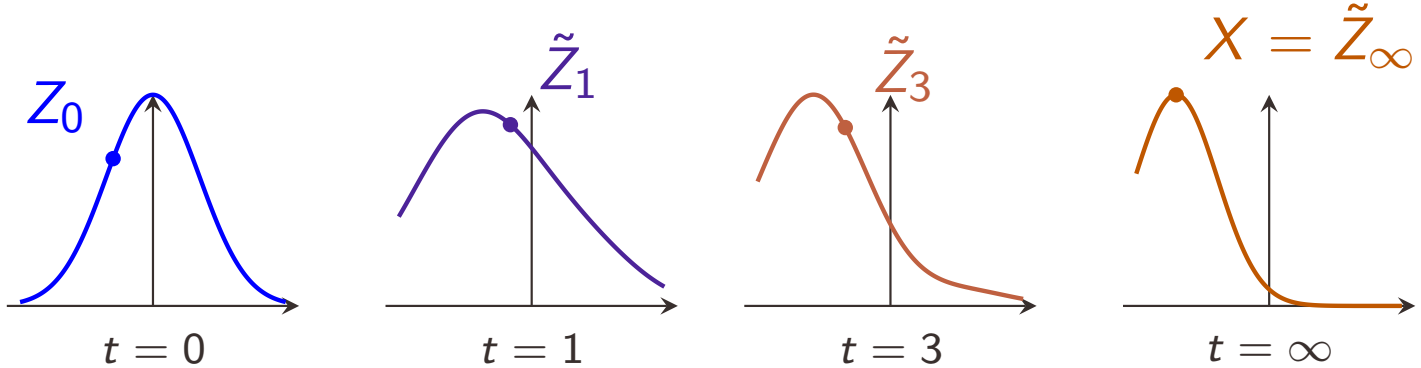
Conditional Dynamics: Mode Selection



$$\tilde{Z}_\infty \sim \rho_*(\cdot \mid \text{left well})$$

Conditional Dynamics: Mode Selection





$$\tilde{Z}_\infty \sim \rho_*(\cdot \mid \text{left well})$$

Mathematical Set-Up

We assume we have a measurement operator $\mathcal{A} : \mathbb{R}^K \to \mathbb{R}^L$

$$\mathcal{A}(x) = y$$

Mathematical Set-Up

We assume we have a measurement operator $\mathcal{A} : \mathbb{R}^K \to \mathbb{R}^L$

$$\mathcal{A}(x) = y$$

For the inpainting problem, $\mathcal{A}(x) = Px$ where P is a projection on the observed pixels. For example, dropping the second pixel:

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Mathematical Set-Up

We assume we have a measurement operator $\mathcal{A} : \mathbb{R}^K \to \mathbb{R}^L$

$$\mathcal{A}(x) = y$$

For the inpainting problem, $\mathcal{A}(x) = Px$ where P is a projection on the observed pixels. For example, dropping the second pixel:

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In super-resolution, $\mathcal{A}(x) = Ax$ is a projection on a lower resolution grid. For example, averaging neighboring pixels:

$$A = \frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Mathematical Set-Up

We are interested in sampling from a measure that is consistent with the observation and close to the prior ρ_*:

Mathematical Set-Up

We are interested in sampling from a measure that is consistent with the observation and close to the prior ρ_*:

$$\mu_y = \operatorname*{arg\,min}_{q \in \mathcal{P}(\mathbb{R}^K)} \frac{1}{2}\mathbb{E}_{x\sim q}[\|\mathcal{A}(x) - y\|^2_{\mathbb{R}^L}] + \lambda \mathcal{H}(q|\rho_*)$$

Mathematical Set-Up

We are interested in sampling from a measure that is consistent with the observation and close to the prior ρ_*:

$$\mu_y = \underset{q \in \mathcal{P}(\mathbb{R}^K)}{\arg\min} \frac{1}{2}\mathbb{E}_{x\sim q}[\|\mathcal{A}(x) - y\|^2_{\mathbb{R}^L}] + \lambda \mathcal{H}(q|\rho_*)$$

Then, we can compute

$$\mu_y(x) = \frac{e^{-\frac{\|\mathcal{A}(x)-y\|^2_{\mathbb{R}^L}}{2\lambda}} \rho_*(x)}{Z}$$

In inverse problems, the set up is usually written as

$$y = \mathcal{A}(x) + \zeta,$$

where $\zeta \in \mathcal{N}(0, \lambda I)$.

In inverse problems, the set up is usually written as

$$y = \mathcal{A}(x) + \zeta,$$

where $\zeta \in \mathcal{N}(0, \lambda I)$.

So, we are interested in sampling from the measure induced by conditioning

$$\text{Prob}(x|y) = \mu_y(x) = \frac{e^{-\frac{\|\mathcal{A}(x)-y\|^2_{\mathbb{R}^L}}{2\lambda}} \rho_*(x)}{Z}$$

In inverse problems, the set up is usually written as

$$y = \mathcal{A}(x) + \zeta,$$

where $\zeta \in \mathcal{N}(0, \lambda I)$.

So, we are interested in sampling from the measure induced by conditioning

$$\text{Prob}(x|y) = \mu_y(x) = \frac{e^{-\frac{\|\mathcal{A}(x)-y\|^2_{\mathbb{R}^L}}{2\lambda}} \rho_*(x)}{Z}$$

The relative entropy or the Gaussian noise can be replaced by other sensible choices.

Avoiding Heavy Lifting

In theory, we could create a new dataset that is consistent with the measurement operator and learn the gradient of logarithm for O-U process with μ_y as initial condition.

Avoiding Heavy Lifting

In theory, we could create a new dataset that is consistent with the measurement operator and learn the gradient of logarithm for O-U process with μ_y as initial condition.

The main point is to try avoid doing expensive computations again, even if we pay an acceptable bias. Can we recycle the score we have now precomputed to find a path from

$$\gamma \to \mu_y?$$

Surrogate Path

Problem: Find a path

$$\mu_t = g_t \overrightarrow{\rho}_t \qquad \text{(Surrogate Path)}$$

such that

$$\mu_0 = \mu_y \qquad \text{and} \qquad \mu_\infty = \gamma.$$

Surrogate Path

Problem: Find a path

$$\mu_t = g_t \overrightarrow{\rho}_t \qquad \text{(Surrogate Path)}$$

such that

$$\mu_0 = \mu_y \qquad \text{and} \qquad \mu_\infty = \gamma.$$

That we can approximately invert.

First Try: OU

We can apply the same reasoning for the noising process:

$$\begin{cases} \partial_t \mu_t^{OU} = \Delta \mu_t^{OU} + \nabla \cdot (x \mu_t^{OU}) \\ \mu_0^{OU} = \mu_y. \end{cases}$$

First Try: OU

We can apply the same reasoning for the noising process:

$$\begin{cases} \partial_t \mu_t^{OU} = \Delta \mu_t^{OU} + \nabla \cdot (x \mu_t^{OU}) \\ \mu_0^{OU} = \mu_y. \end{cases}$$

$$\mu_t^{OU}(x) = P(X_t = x|y) = \frac{\overbrace{P(y|X_t = x)}^{g_t(x)} P(X_t = x)}{P(y)}$$

First Try: OU

We can apply the same reasoning for the noising process:

$$\begin{cases} \partial_t \mu_t^{OU} = \Delta \mu_t^{OU} + \nabla \cdot (x \mu_t^{OU}) \\ \mu_0^{OU} = \mu_y. \end{cases}$$

$$\mu_t^{OU}(x) = P(X_t = x|y) = \frac{\overbrace{P(y|X_t = x)}^{g_t(x)} P(X_t = x)}{P(y)}$$

To invert the path, as before, we need to have an approximation of the vectorfield

$$\nabla \log \mu_t^{OU}(x) = \underbrace{\nabla \log P(y|X_t = x)}_{\nabla \log g_t(x)} + \underbrace{\nabla \log \overrightarrow{\rho}_t(x)}_{\text{Diffusion Oracle}}.$$

Exact solution

We know μ_t^{OU} and $\overrightarrow{\rho}_t$ solves O-U and

$$\mu_t^{OU} = \frac{g_t \overrightarrow{\rho}_t}{Z},$$

then

Exact solution

We know μ_t^{OU} and $\overrightarrow{\rho}_t$ solves O-U and

$$\mu_t^{OU} = \frac{g_t \overrightarrow{\rho}_t}{Z},$$

then

$$\begin{cases} \partial_t g_t = \Delta g_t + x \nabla g_t + 2 \nabla \log \overrightarrow{\rho}_t g_t \\ g_0 = P(y|X_0 = x) = e^{-R_y(x)}. \end{cases}$$

Stochastic Characteristics

Moreover, we can use Stochastic Characteristics to solve

$$g_t(x) = \mathbb{E}\left[g_0(X_0) \middle| X_t = x\right].$$

Stochastic Characteristics

Moreover, we can use Stochastic Characteristics to solve

$$g_t(x) = \mathbb{E}\left[g_0(X_0) \middle| X_t = x\right].$$

Computational Complexity

This is too computationally heavy to be done in practice. Also, we need access to $\nabla \log g_t(x)$ which adds another layer of complication.

Taking approximations ML style

The original paper of Diffusion Posterior Sampling (DPS), just says that we can pass the expectation inside and hope for the best:

$$\mathbb{E}\left[g_0(X_0)|\, X_t = x\right] = g_t(x) \neq h_t(x) = g_0(\mathbb{E}\left[X_0|\, X_t = x\right])$$

Taking approximations ML style

The original paper of Diffusion Posterior Sampling (DPS), just says that we can pass the expectation inside and hope for the best:

$$\mathbb{E}\left[g_0(X_0)|\, X_t = x\right] = g_t(x) \neq h_t(x) = g_0(\mathbb{E}\left[X_0|\, X_t = x\right])$$

The advantage is that the right hand side can be computed using the diffusion oracle

$$\mathbb{E}\left[X_0|\, X_t = x\right] = e^t x + (e^t - e^{-t})\nabla \log \overrightarrow{\rho}_t(x).$$

(Tweedie's formula)

Ode to simplicity

We can consider

$$\mu_t^{OU} \neq \mu_t^{DPS} = \frac{\overbrace{e^{-R_y(\mathbb{E}[X_0|X_t=x])}}^{\text{Explicitly known}} \overrightarrow{\rho}_t}{Z_t}$$

Ode to simplicity

We can consider

$$\mu_t^{OU} \neq \mu_t^{DPS} = \frac{\overbrace{e^{-R_y(\mathbb{E}[X_0|X_t=x])}}^{\text{Explicitly known}} \overrightarrow{\rho}_t}{Z_t}$$

Using the properties of the conditional expectation

$$\lim_{t\to 0^+} \mu_t^{DPS} = \mu_y \qquad \text{and} \qquad \lim_{t\to\infty} \mu_t^{DPS} = \gamma.$$

Ode to simplicity

We can consider

$$\mu_t^{OU} \neq \mu_t^{DPS} = \frac{\overbrace{e^{-R_y(\mathbb{E}[X_0|X_t=x])}}^{\text{Explicitly known}} \overrightarrow{\rho}_t}{Z_t}$$

Using the properties of the conditional expectation

$$\lim_{t\to 0^+} \mu_t^{DPS} = \mu_y \qquad \text{and} \qquad \lim_{t\to\infty} \mu_t^{DPS} = \gamma.$$

and we have direct access to compute

$$\nabla \log \mu_t^{DPS} = -\nabla(R_y(\mathbb{E}[X_0|\, X_t = x])) + \nabla \log \overrightarrow{\rho}_t$$

by knowing R_y and the diffusion oracle.

Property of the expected value

Let's understand the conditional expectation $\mathbb{E}[X_0|X_t = x]$ a bit better. We have the following property:

$$\mathbb{E}[X_0|X_t = x] \in \text{conv}(\text{Supp}\,\rho_*).$$

Property of the expected value

Let's understand the conditional expectation $\mathbb{E}[X_0|X_t = x]$ a bit better. We have the following property:

$$\mathbb{E}[X_0|X_t = x] \in \text{conv}(\text{Supp}\,\rho_*).$$

So, for instance if $\rho_* = \delta_0$, then $\mathbb{E}[X_0|X_t = x] = 0$ for all $t \geq 0$ and $x \in \mathbb{R}^K$.

Property of the expected value

Let's understand the conditional expectation $\mathbb{E}[X_0|X_t = x]$ a bit better. We have the following property:

$$\mathbb{E}[X_0|X_t = x] \in \text{conv}(\text{Supp}\,\rho_*).$$

So, for instance if $\rho_* = \delta_0$, then $\mathbb{E}[X_0|X_t = x] = 0$ for all $t \geq 0$ and $x \in \mathbb{R}^K$.

If $\rho_* = \frac{1}{2}(\delta_{-1} + \delta_1)$, then $\mathbb{E}[X_0|X_t = x] \in [-1, 1]$ for all $t \geq 0$ and $x \in \mathbb{R}^K$.

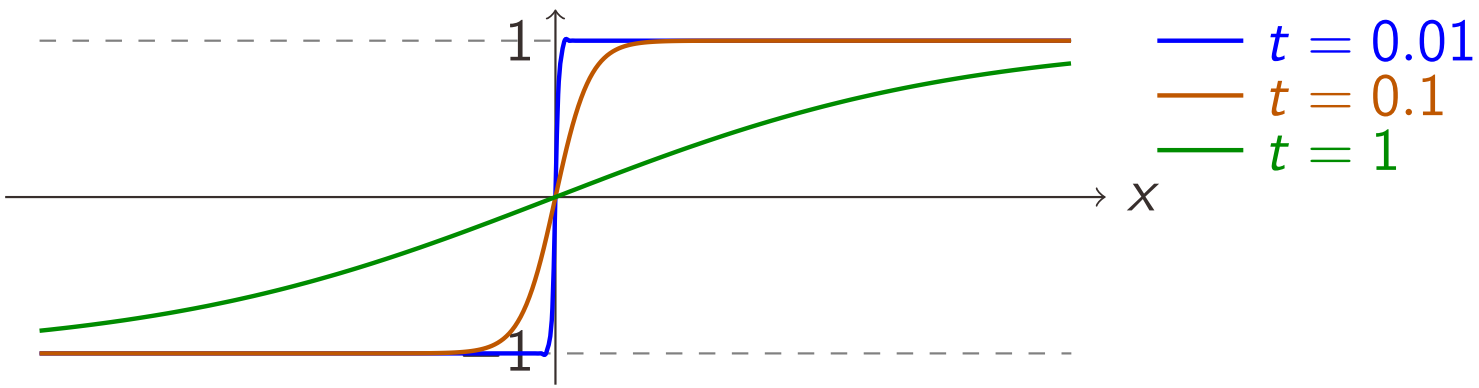
Binary prior: explicit posterior mean

Consider

$$dX_t = -X_t\,dt + \sqrt{2}\,dB_t, \qquad X_0 \sim \tfrac{1}{2}(\delta_{-1} + \delta_1).$$

For fixed $t > 0$:

$$\mathbb{E}[X_0 \mid X_t = x] = \tanh\left(\frac{e^{-t}}{1 - e^{-2t}}x\right) = \tanh\left(\frac{x}{e^t - e^{-t}}\right).$$



DPS Algorithm

- Pick a time horizon T large enough.

DPS Algorithm

- Pick a time horizon T large enough.
- Sample a seed $Y_0 \sim \mathcal{N}(0, I)$.

DPS Algorithm

- Pick a time horizon T large enough.
- Sample a seed $Y_0 \sim \mathcal{N}(0, I)$.
- Solve the SDE numerically:

$$dY_t = \left(Y_t + 2\nabla \log \mu_{T-t}^{DPS}(Y_t) \right) dt + \sqrt{2} dB_t$$

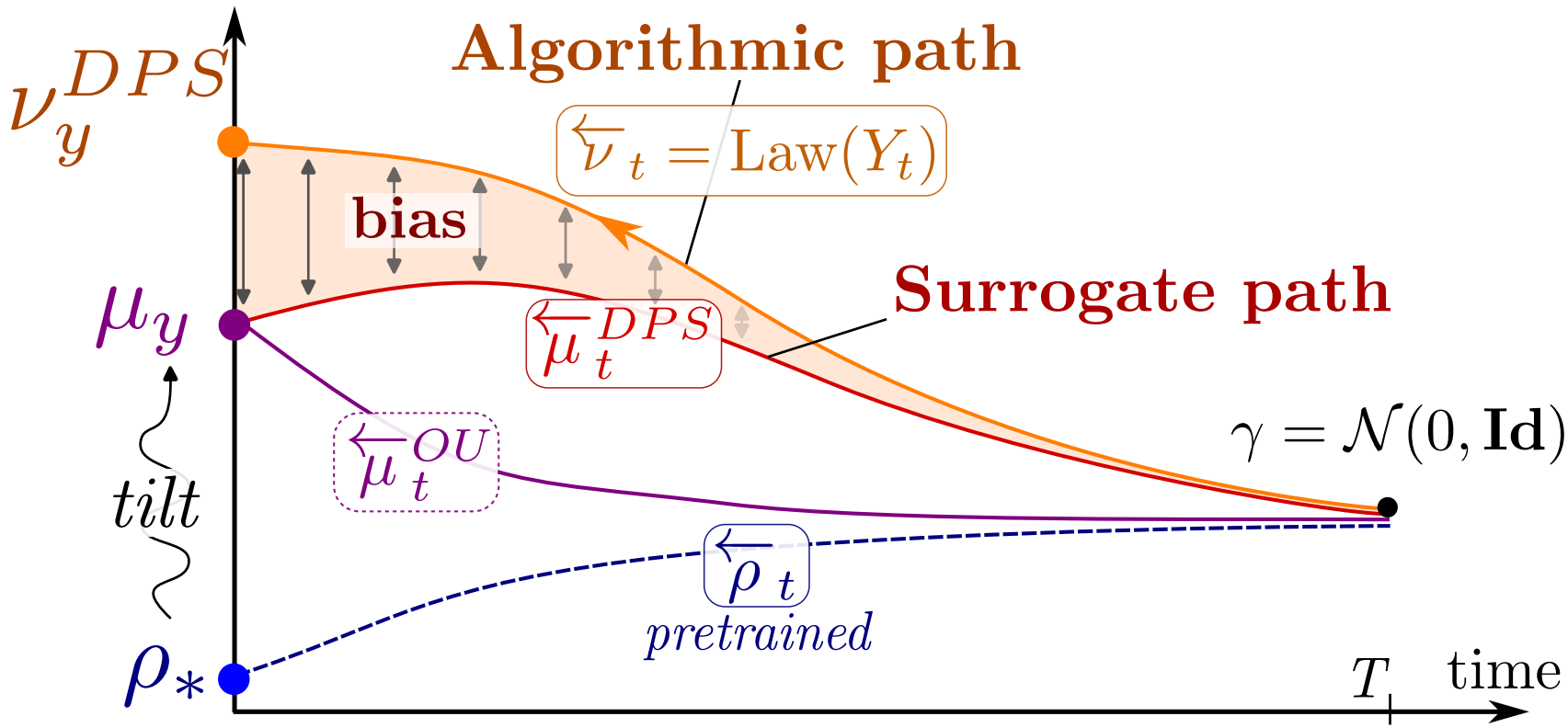
DPS Algorithm

- Pick a time horizon T large enough.
- Sample a seed $Y_0 \sim \mathcal{N}(0, I)$.
- Solve the SDE numerically:

$$dY_t = \left(Y_t + 2\nabla \log \mu_{T-t}^{DPS}(Y_t) \right) dt + \sqrt{2} dB_t$$

- Use Y_T as a sample of ν_y^{DPS}.

Overview of the Construction



Where is the catch?

Biased sampler

DPS is not exact even for Gaussian measures!

Where is the catch?

Biased sampler

DPS is not exact even for Gaussian measures!

In a nutshell, μ_t^{DPS} does not satisfy O-U:

Where is the catch?

Biased sampler

DPS is not exact even for Gaussian measures!

In a nutshell, μ_t^{DPS} does not satisfy O-U:

$$\begin{cases} \partial_t \mu_t^{DPS} = \Delta \mu_t^{DPS} + \nabla \cdot (x \mu_t^{DPS}) + (c_t - \mathbb{E}_{\mu_t^{DPS}} c_t) \mu_t^{DPS} \\ \mu_0^{DPS} = \mu_y \end{cases}$$

where

$$\begin{aligned} c_t(x) = \operatorname{trace}(\nabla \mathbb{E}[X_0 | X_t = x] \, D^2 R_y \nabla \mathbb{E}[X_0 | X_t = x]) \\ - |\nabla R_y \nabla \mathbb{E}[X_0 | X_t = x]|^2 \end{aligned}$$

The reverse process for DPS

Given a fixed time horizon T, we can consider the equation satisfied by

$$\overleftarrow{\mu}_t^{DPS} = \mu_{T-t}^{DPS}.$$

The reverse process for DPS

Given a fixed time horizon T, we can consider the equation satisfied by

$$\overleftarrow{\mu}_t^{DPS} = \mu_{T-t}^{DPS}.$$

$$\begin{cases} \partial_t \overleftarrow{\mu}_t^{DPS} = \Delta \overleftarrow{\mu}_t^{DPS} - 2\nabla \cdot (\nabla \log \overrightarrow{\mu}_{T-t}^{DPS} \overleftarrow{\mu}_t^{DPS}) - \nabla \cdot (x \overleftarrow{\mu}_t^{DPS}) \\ \qquad\qquad -(c_{T-t} - \mathbb{E}_{\mu_{T-t}^{DPS}} c_{T-t}) \overleftarrow{\mu}_t^{DPS} \\ \overleftarrow{\mu}_0^{DPS} = \overrightarrow{\mu}_T^{DPS}. \end{cases}$$

The reverse process for DPS

Given a fixed time horizon T, we can consider the equation satisfied by

$$\overleftarrow{\mu}_t^{DPS} = \mu_{T-t}^{DPS}.$$

$$\begin{cases} \partial_t \overleftarrow{\mu}_t^{DPS} = \Delta \overleftarrow{\mu}_t^{DPS} - 2\nabla \cdot (\nabla \log \overrightarrow{\mu}_{T-t}^{DPS} \overleftarrow{\mu}_t^{DPS}) - \nabla \cdot (x \overleftarrow{\mu}_t^{DPS}) \\ \qquad\qquad -(c_{T-t} - \mathbb{E}_{\mu_{T-t}^{DPS}} c_{T-t}) \overleftarrow{\mu}_t^{DPS} \\ \overleftarrow{\mu}_0^{DPS} = \overrightarrow{\mu}_T^{DPS}. \end{cases}$$

While $\overleftarrow{\nu}_t = \text{Law}(Y_t)$ satisfies

$$\begin{cases} \partial_t \overleftarrow{\nu}_t = \Delta \overleftarrow{\nu}_t - 2\nabla \cdot (\nabla \log \overrightarrow{\mu}_{T-t}^{DPS} \overleftarrow{\nu}_t) - \nabla \cdot (x \overleftarrow{\nu}_t) \\ \overleftarrow{\nu}_0 = \gamma \end{cases}$$

Main Result

Theorem (D., Motsch, Parulekar, Porteous, Shakkottai)

$$\mu_y(x) = \omega(x)\, \nu_y^{DPS}(x). \tag{1}$$

The weight ω admits a Feynman–Kac representation along the ***backward path****:*

$$\omega(x) = \mathbb{E}\left[\frac{\overrightarrow{\mu}_T(Y_0)}{\gamma(Y_0)} \exp\Big(- \int_0^T c_{DPS}(T-s, Y_s)\, ds \Big) \;\Big|\; Y_T = x \right]. \tag{2}$$

Main Result (continued)

Theorem (continued)

Equivalently, along the ***forward path****:*

$$\frac{1}{\omega(x)} = \mathbb{E}_{X\sim OU}\left[\frac{\gamma(X_T)}{\overrightarrow{\mu}_T(X_T)}\,\exp\Big(\int_0^T c_{DPS}(s, X_s)\,ds\Big)\;\Big|\; X_0 = x\right]. \tag{3}$$

Main Result (continued)

Theorem (continued)

Equivalently, along the ***forward path****:*

$$\frac{1}{\omega(x)} = \mathbb{E}_{X\sim OU}\left[\frac{\gamma(X_T)}{\overrightarrow{\mu}_T(X_T)}\exp\left(\int_0^T c_{DPS}(s, X_s)\,ds\right) \;\middle|\; X_0 = x\right]. \tag{3}$$

Consequence

Importance-weighting DPS samples by ω recovers μ_y exactly.

Discarding trajectories

Trajectories that have a small weight

$$\exp\Big(-\int_0^T c_{DPS}(T-s, Y_s)\, ds\Big)$$

are not meaningful.

Discarding trajectories

Trajectories that have a small weight

$$\exp\Big(-\int_0^T c_{DPS}(T-s, Y_s)\, ds\Big)$$

are not meaningful.

We could potentially just do rejection sampling on the trajectories.

Discarding trajectories

Trajectories that have a small weight

$$\exp\Big(-\int_0^T c_{DPS}(T-s, Y_s)\, ds\Big)$$

are not meaningful.

We could potentially just do rejection sampling on the trajectories.

The catch is that posterior sampling with access to an oracle is exponentially bad in dimension for the general case. So any result that has a polynomial dependence on dimension needs to be for a specific class of simple priors ρ_*.

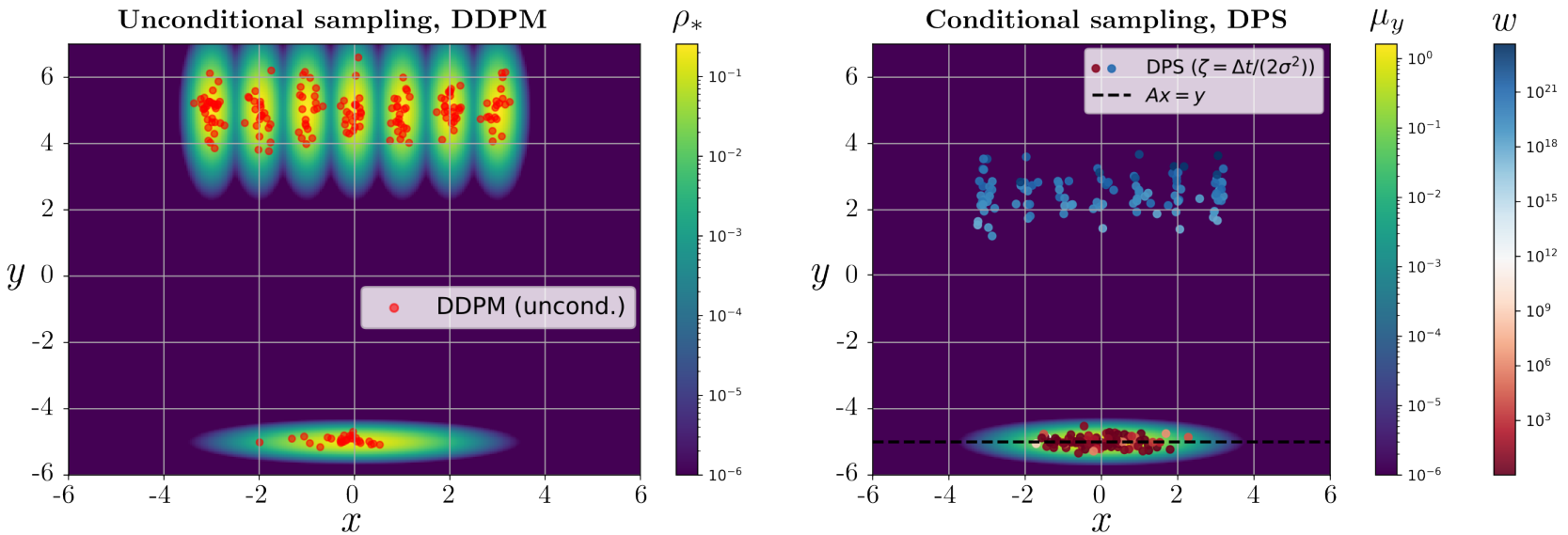


Figure: Left: prior and samples from the denoising SDE. Right: posterior for $R_y(x) = \|x_2 + 5\|^2$ and DPS samples; meaningless samples carry weights $w \sim 10^{20}$.

Numerical Stability

Algorithm 1 DPS

Require: N, y, $\{\zeta_i\}_{i=1}^N$, $\{\tilde{\sigma}_i\}_{i=1}^N$

$x_N \sim \mathcal{N}(0, I)$ ▷ Initialize with Gaussian noise
for $i = N-1$ **to** 0 **do**
  $\hat{\mathsf{s}} \leftarrow \mathsf{s}_\theta(x_i, i)$ ▷ Evaluate the score oracle
  $\hat{x}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(x_i + (1-\bar{\alpha}_i)\hat{\mathsf{s}})$ ▷ Compute $\mathbb{E}[X_0|X_t = x]$
  $z \sim \mathcal{N}(0, I)$
  $x'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i} x_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{x}_0 + \tilde{\sigma}_i z$ ▷ Integrate the score term
  $x_{i-1} \leftarrow x'_{i-1} - \zeta_i \nabla_{x_i} \|y - \mathcal{A}(\hat{x}_0)\|^2$ ▷ Integrate the bias term **explicitly**
end for
return x_0

Numerical Stability

Algorithm 2 DPS

Require: N, y, $\{\zeta_i\}_{i=1}^N$, $\{\tilde{\sigma}_i\}_{i=1}^N$

$x_N \sim \mathcal{N}(0, I)$ ▷ Initialize with Gaussian noise
for $i = N-1$ **to** 0 **do**
  $\hat{s} \leftarrow s_\theta(x_i, i)$ ▷ Evaluate the score oracle
  $\hat{x}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(x_i + (1-\bar{\alpha}_i)\hat{s})$ ▷ Compute $\mathbb{E}[X_0|X_t = x]$
  $z \sim \mathcal{N}(0, I)$
  $x'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i} x_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{x}_0 + \tilde{\sigma}_i z$ ▷ Integrate the score term
  $x_{i-1} \leftarrow x'_{i-1} - \zeta_i \nabla_{x_i} \|y - \mathcal{A}(\hat{x}_0)\|^2$ ▷ Integrate the bias term **explicitly**
end for
return x_0

$$\zeta_i = \frac{1}{\|y - \mathcal{A}(\hat{x}_0)\|} \neq \Delta t$$

What is Δt?

The time discretization is not fixed in time

$$\Delta t_i = -\frac{1}{2}\ln(1-\beta_i) \tag{4}$$

The linear noise schedule of DDPM is given by

$$\beta_i = \beta_{min} + i\frac{\beta_{max}-\beta_{min}}{N} \approx 10^{-4} + 2i\,10^{-5} \qquad \text{for } i = 1,\ldots,1000,$$

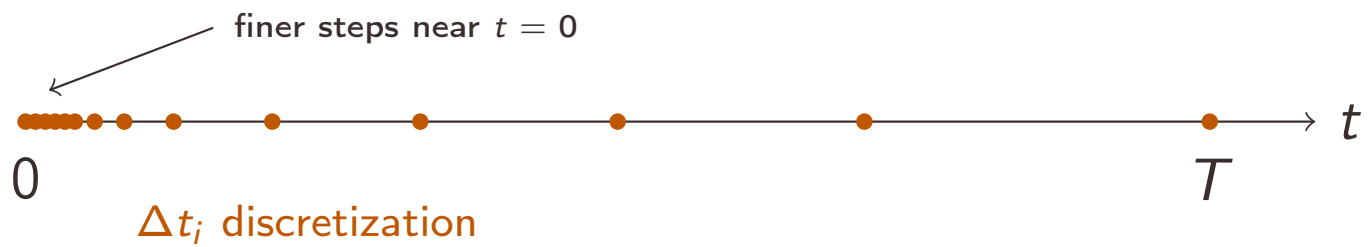
What is Δt?

The time discretization is not fixed in time

$$\Delta t_i = -\frac{1}{2}\ln(1-\beta_i) \tag{4}$$

The linear noise schedule of DDPM is given by

$$\beta_i = \beta_{min} + i\frac{\beta_{max}-\beta_{min}}{N} \approx 10^{-4} + 2i\,10^{-5} \qquad \text{for } i = 1,\ldots,1000,$$



Hidden annealing schedule

The actual surrogate path of DPS has the form

$$\mu_t^{DPS} = \frac{e^{-\eta_t R_y(\mathbb{E}[X_0|X_t=x])}}{Z}\rho_t$$

where the annealing schedule is given by

$$\eta_t \approx \frac{10^5}{1+300\sqrt{t}}.$$

Hidden annealing schedule

The actual surrogate path of DPS has the form

$$\mu_t^{DPS} = \frac{e^{-\eta_t R_y(\mathbb{E}[X_0|X_t=x])}}{Z}\rho_t$$

where the annealing schedule is given by

$$\eta_t \approx \frac{10^5}{1+300\sqrt{t}}.$$

The numerical integration of the biasing vector field does not satisfy the Forward Euler stability criterion!

Oscillations!

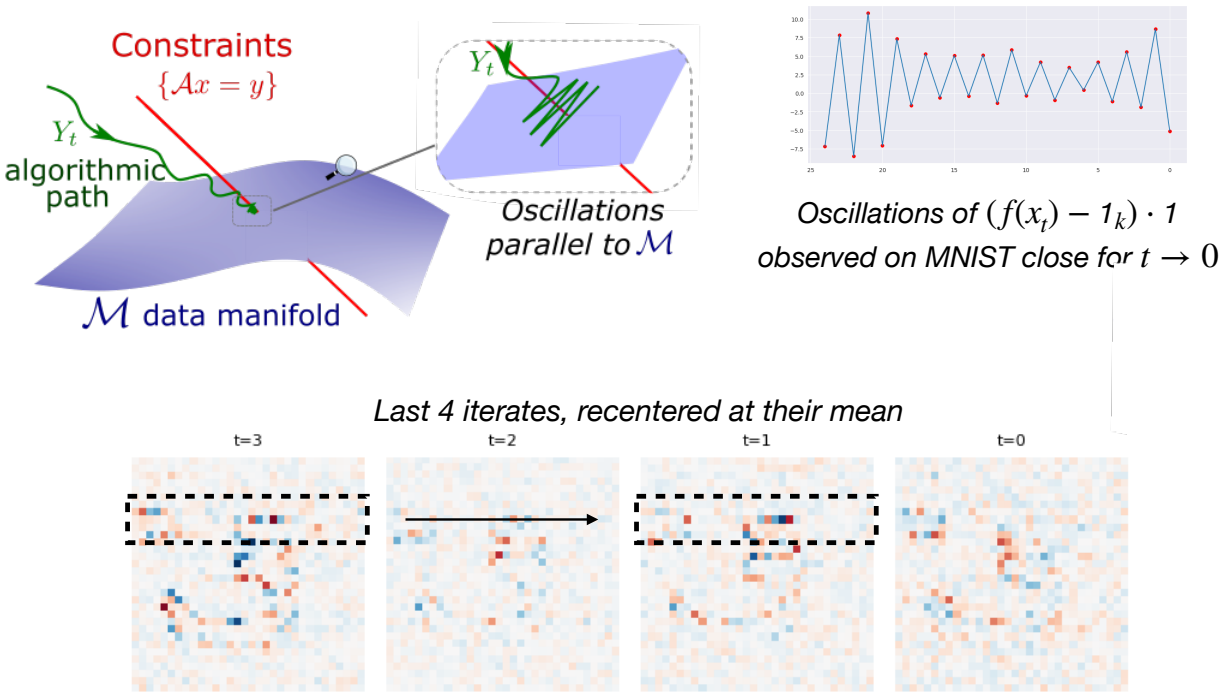


Figure: **(Top Left)** A pictorial depiction of instability. **(Top Right)** An exhibition of these oscillation on a posterior sampling task with an MNIST prior. **(Bottom)** A plot of the last four iterates of DPS, re-centered about their mean. The guidance tilted the distribution towards the digit 3. We observe periodic oscillations in pixel space.

Fixes

Fixes

- If the goal is not to enforce the hard constraint, practitioners (e.g. Nano Banana) turn off guidance in the last few steps.

Fixes

- If the goal is not to enforce the hard constraint, practitioners (e.g. Nano Banana) turn off guidance in the last few steps.
- Implicit integration is another successful fix, e.g. RB-Modulation, deployed in Google Pixel (2026).

Fixes

- If the goal is not to enforce the hard constraint, practitioners (e.g. Nano Banana) turn off guidance in the last few steps.
- Implicit integration is another successful fix, e.g. RB-Modulation, deployed in Google Pixel (2026).

The analysis explains post-hoc the advantage of these choices made for deployed versions of the algorithms.

Outlook

Outlook

- **Design problem:** How do you find a **surrogate path** that reduces bias?

Outlook

- **Design problem:** How do you find a **surrogate path** that reduces bias? Our analysis also explains from first principles some heuristic choices.

Outlook

- **Design problem:** How do you find a **surrogate path** that reduces bias? Our analysis also explains from first principles some heuristic choices.
- The duality between reaction terms and drift has not been understood/exploited in the ML community.

Outlook

- **Design problem:** How do you find a **surrogate path** that reduces bias? Our analysis also explains from first principles some heuristic choices.
- The duality between reaction terms and drift has not been understood/exploited in the ML community. Possible applications to RL.

Outlook

- **Design problem:** How do you find a **surrogate path** that reduces bias? Our analysis also explains from first principles some heuristic choices.
- The duality between reaction terms and drift has not been understood/exploited in the ML community. Possible applications to RL.
- The paradigm of sampling a prior might be changing from diffusion to gradient flows.

Outlook

- **Design problem:** How do you find a **surrogate path** that reduces bias? Our analysis also explains from first principles some heuristic choices.
- The duality between reaction terms and drift has not been understood/exploited in the ML community. Possible applications to RL.
- The paradigm of sampling a prior might be changing from diffusion to gradient flows.Their posterior sampling analogue is not yet clear.

Thank You!