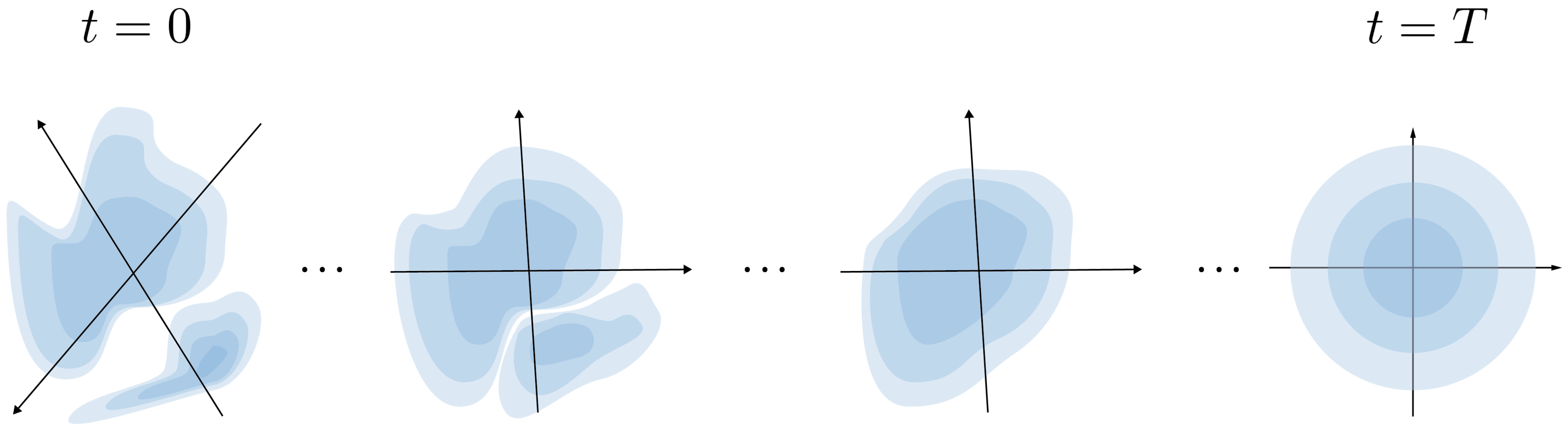
Images from @midjourney.gallery

GENERATIVE MODELS

Flexibility

Tractability

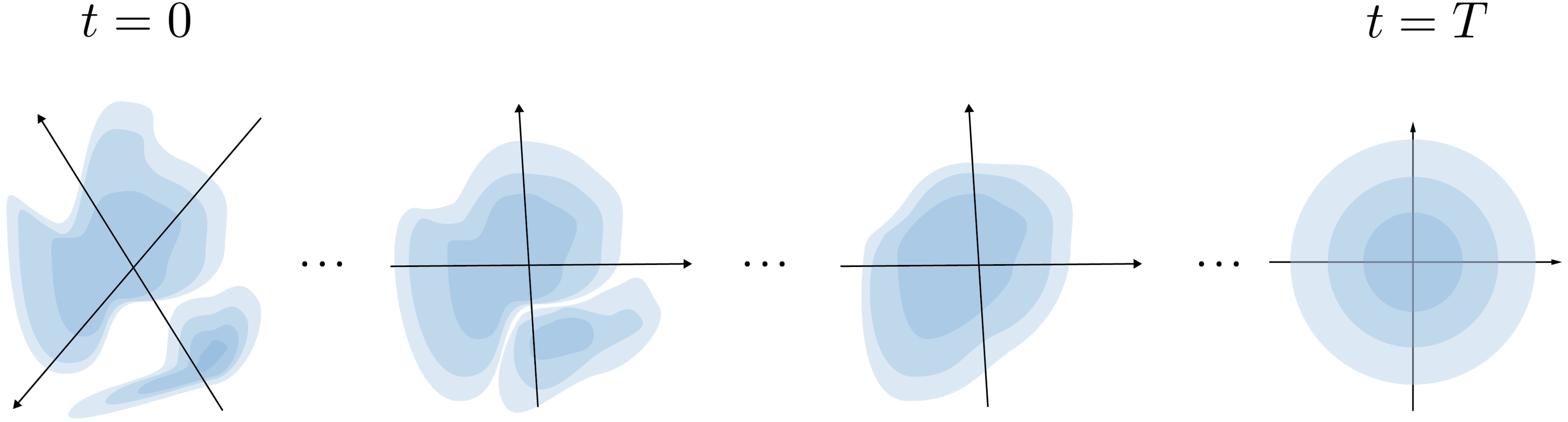
THE DIFFUSION PROCESS



[1] Feller, W. On the theory of stochastic processes, with particular reference to applications. In Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability. The Regents of the University of California, 1949.

THE DIFFUSION PROCESS

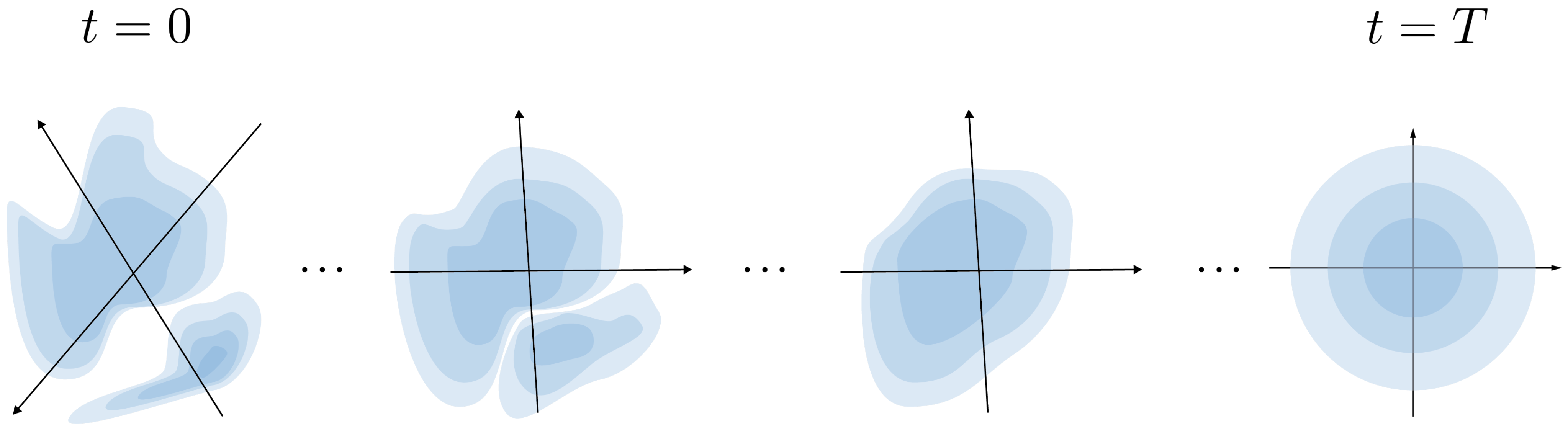
In the limit of small step size, the reversal of the diffusion process has the identical functional form as the forward process [1]

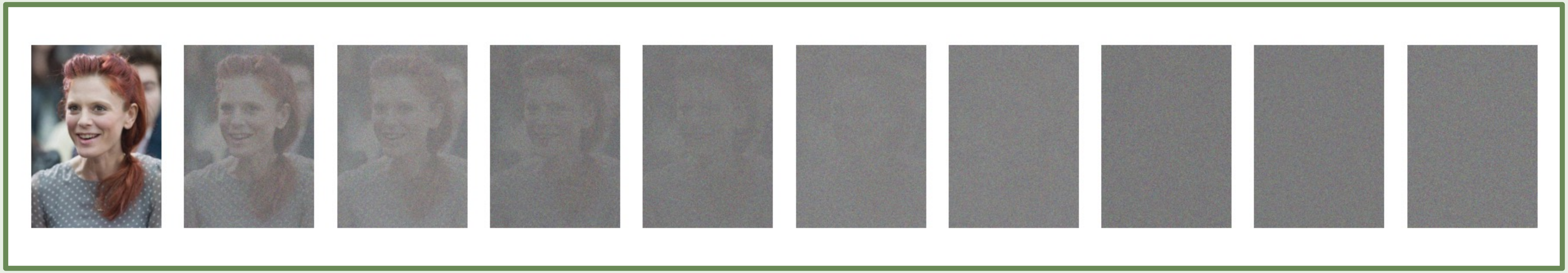


[1] Feller, W. On the theory of stochastic processes, with particular reference to applications. In Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability. The Regents of the University of California, 1949.

THE DIFFUSION PROCESS

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$$





$$X_t = \sqrt{1-\beta_t}X_{t-1} + \sqrt{\beta_t}\epsilon_t$$

Take one step at a time.

THE REPARAMETRIZATION TRICK

Taking one step at a time is slow.
We need a faster way to sample to allow quick forward diffusion.

$$X_t = \sqrt{1-\beta_t}\boxed{X_{t-1}} + \sqrt{\beta_t}\epsilon_t$$

Expand the recursive formulation.

$$X_t = \sqrt{1-\beta_t}\boxed{\left(\sqrt{1-\beta_{t-1}}X_{t-2} + \sqrt{\beta_{t-1}}\epsilon_{t-1}\right)} + \sqrt{\beta_t}\epsilon_t$$

Develop the calculations as an exercise (check here for the solution).

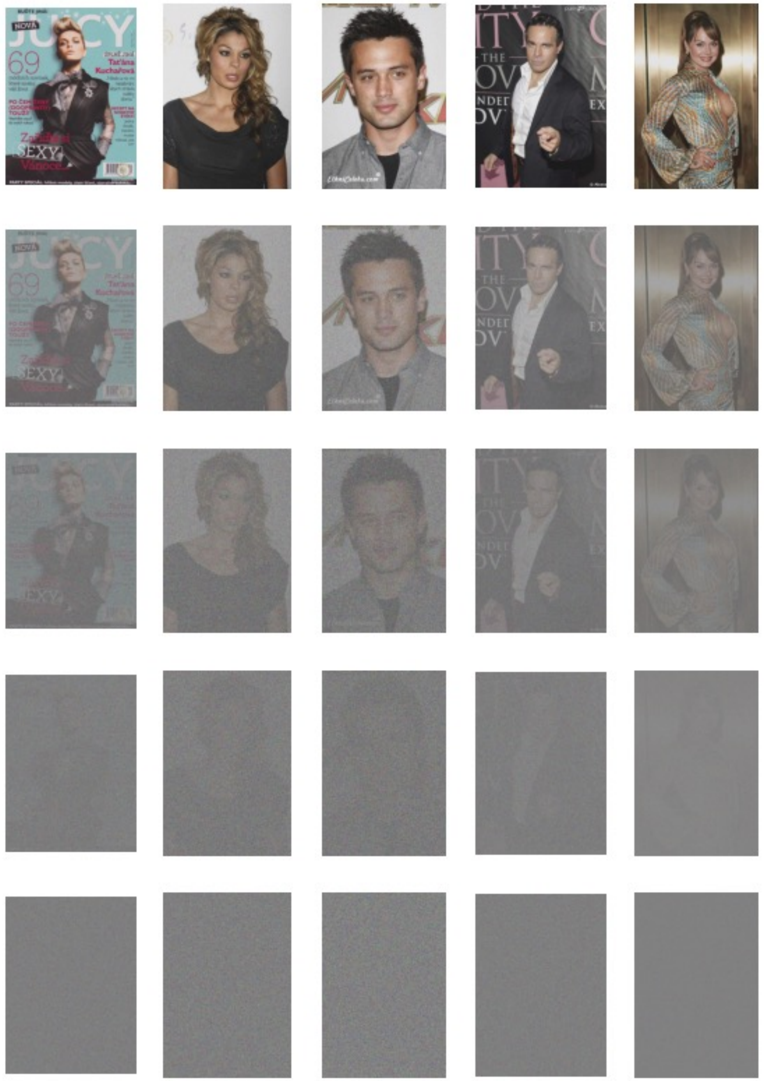
THE FORWARD PROCESS

$$q(x_t|x_0) = \mathcal{N}(x_t; \underbrace{\boxed{\sqrt{\bar{\alpha}_t} x_0}}_{\text{mean}}, \underbrace{\boxed{(1-\bar{\alpha}_t)I}}_{\text{variance}})$$

$$\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$$

$$= \prod_{i=1}^{t}\alpha_i$$

This is used only to simplify notation



$\bar{\alpha}_t$

0.99

0.93

0.56

0.06

0.001

THE FORWARD PROCESS

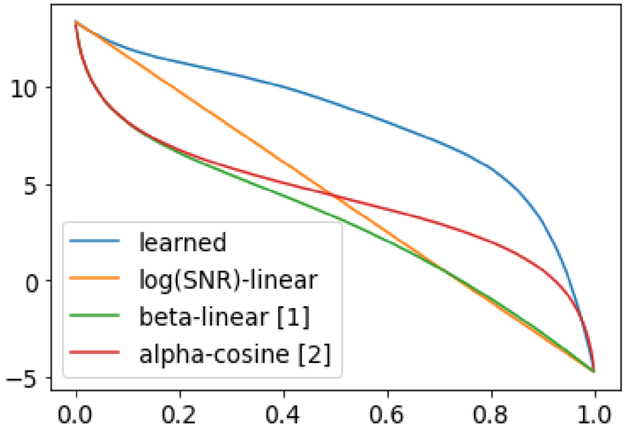
$$q(x_t|x_0) = \mathcal{N}(x_t; \underbrace{\sqrt{\bar{\alpha}_t} x_0}_{\text{mean}}, \underbrace{(1-\bar{\alpha}_t)I}_{\text{variance}})$$

Signal-to-Noise Ratio

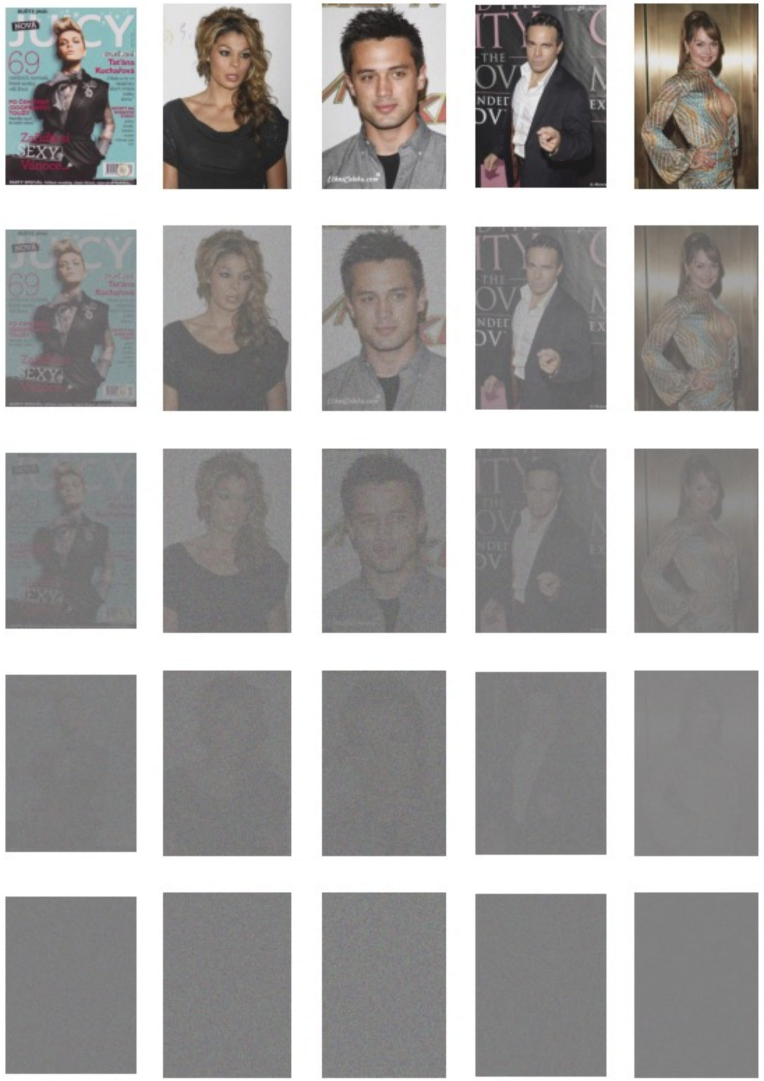
Direct measure of the effect of the noise on the input image

$$\mathrm{SNR}(t) = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}$$

Noise schedule



(a) log SNR vs time t



$\bar{\alpha}_t$

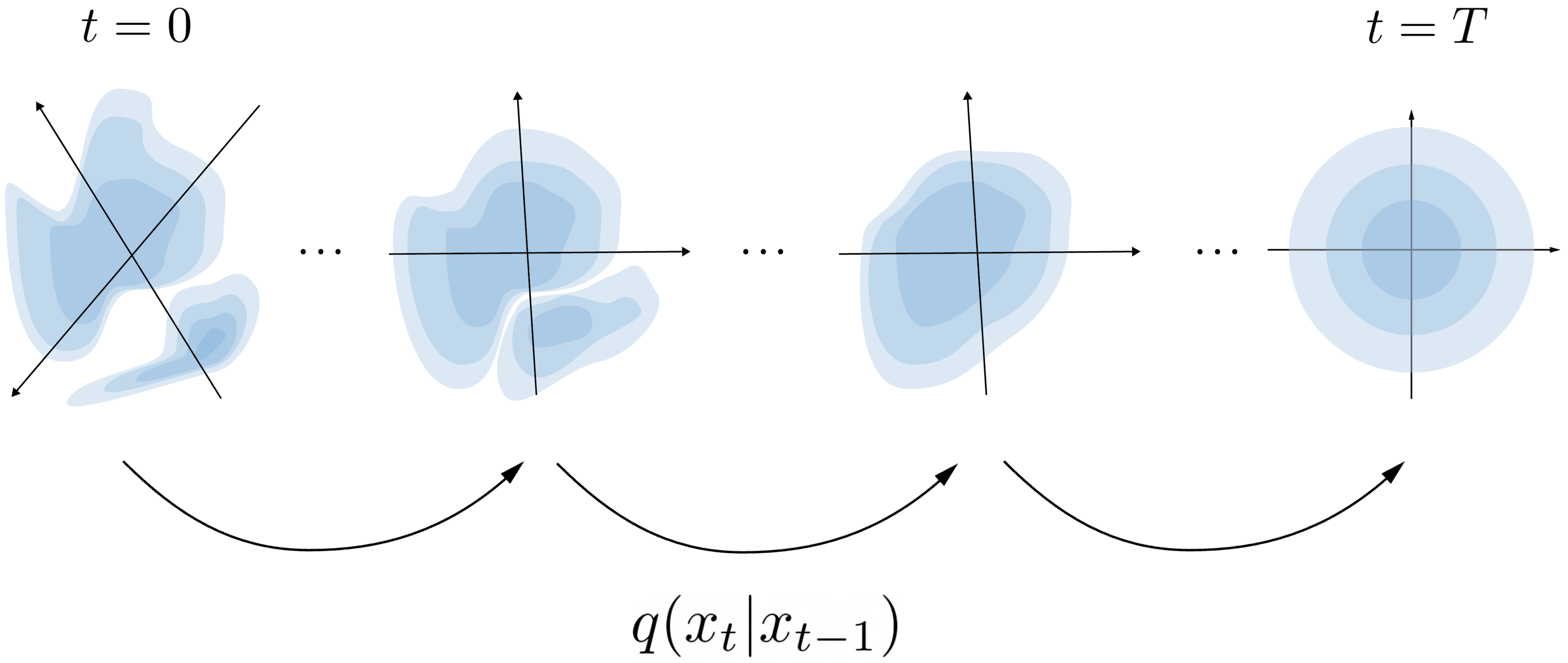
0.99

0.93

0.56

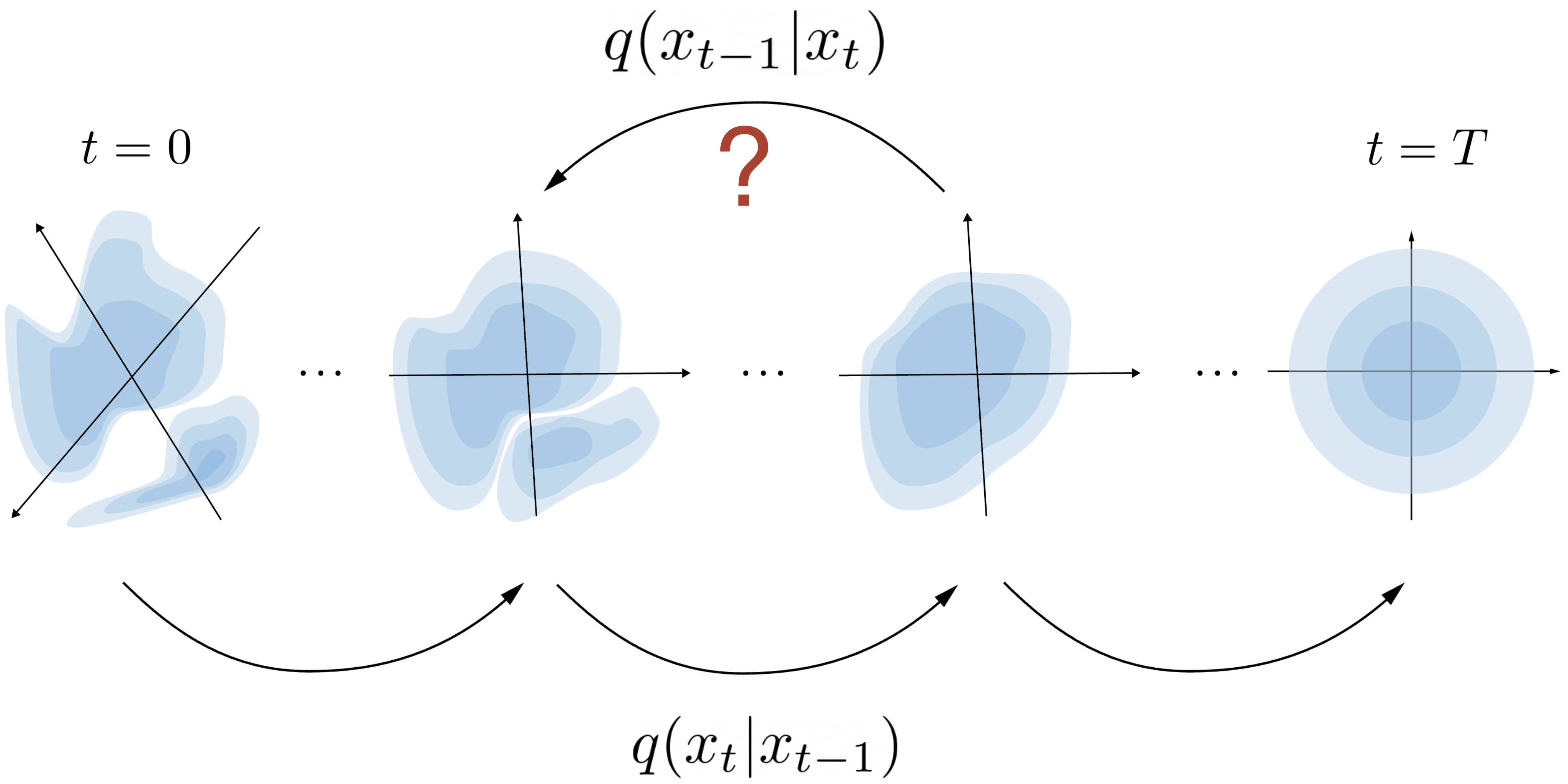
0.06

0.001



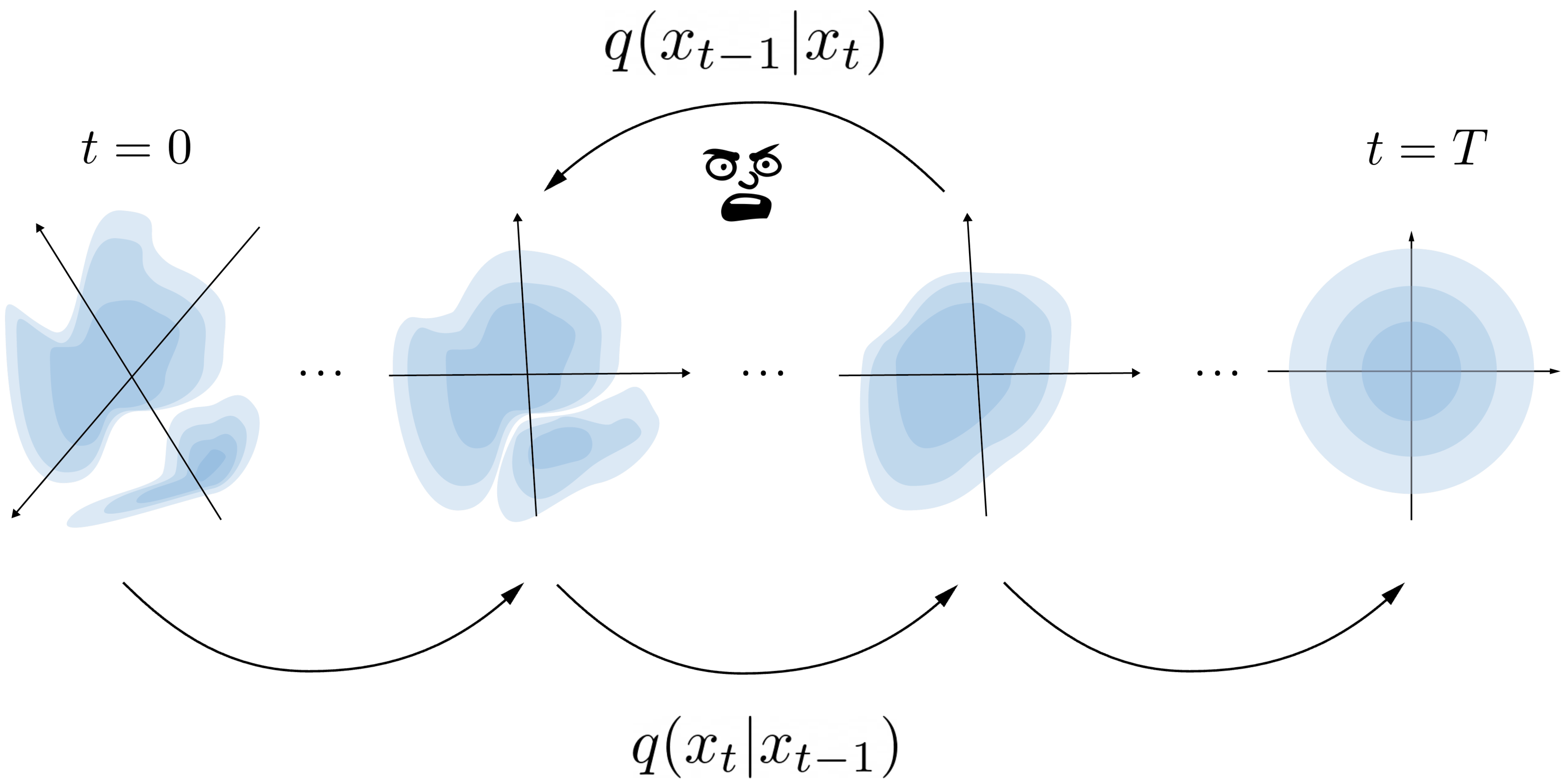
Now we know how to map from complex distribution to a simple one. How do we go back?

REVERSING THE PROCESS



Can we calculate this analytically? We know everything about the forward process.

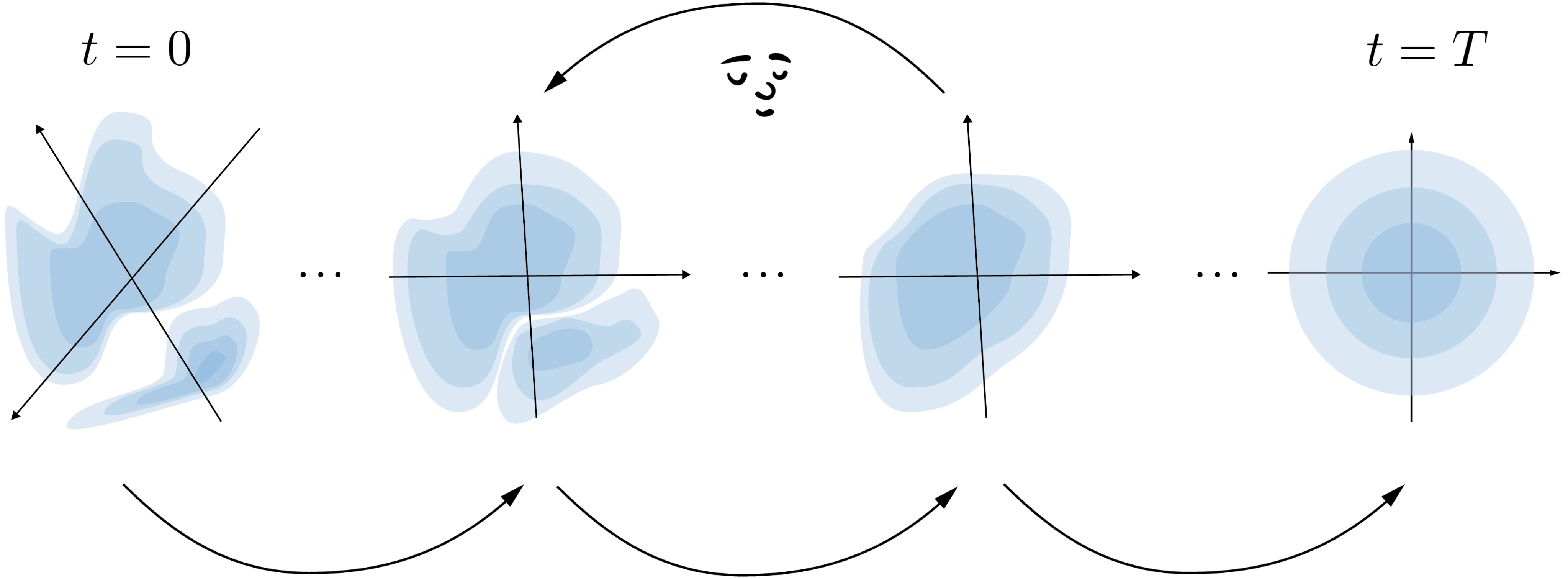
REVERSING THE PROCESS



We need marginalization over the whole dataset.

REVERSING THE PROCESS

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_t; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$



$$q(x_t|x_{t-1})$$

Just learn it!

LEARNING THE REVERSE

Minimize the `expected negative log-likelihood`

$$L = \mathbb{E}_{x_0 \sim q(x_0)} \left[-\log p_\theta(x_0) \right]$$

LEARNING THE REVERSE

A little taste of the algebra, use the total probability theorem

$$L = -\mathbb{E}_{x_0}\left[\log \int_{x_{1:T}} p_\theta(x_{0:T})dx_{1:T}\right]$$

LEARNING THE REVERSE

Use the **Markov Chain** definition and upperbound with **Jensen's inequality**

$$\mathcal{L} = \mathbb{E}_{x_{0:T}}\left[\underbrace{\log \frac{q(x_T|x_{T-1})}{q(x_T)}}_{\text{prior}} + \underbrace{\sum_{t=1}^{T-1} \log \frac{q(x_t|x_{t-1})}{p_\theta(x_t|x_{t+1})}}_{\text{forward vs. reverse}} \underbrace{- \log p_\theta(x_0|x_1)}_{\text{recon.}}\right]$$

LEARNING THE REVERSE

Use the **Markov Chain** definition and upperbound with **Jensen's inequality**

$$\mathcal{L} = \mathbb{E}_{x_{0:T}} \left[\log \frac{q(x_T|x_{T-1})}{q(x_T)} + \underbrace{\sum_{t=1}^{T-1} \log \frac{q(x_t|x_{t-1})}{p_\theta(x_t|x_{t+1})}}_{\text{forward vs. reverse}} - \log p_\theta(x_0|x_1) \right]$$

LEARNING THE REVERSE

Use the **Markov Chain** definition and upperbound with **Jensen's inequality**

?

$$\mathcal{L} = \mathbb{E}_{x_{0:T}}\left[\log \frac{q(x_T|x_{T-1})}{q(x_T)} + \underbrace{\sum_{t=1}^{T-1} \log \frac{q(x_t|\boxed{x_{t-1}})}{p_\theta(x_t|\boxed{x_{t+1}})}}_{\text{forward vs. reverse}} - \log p_\theta(x_0|x_1)\right]$$

LEARNING THE REVERSE

Use the **Markov Chain** definition and upperbound with **Jensen's inequality**

HIGH VARIANCE

$$\mathcal{L} = \mathbb{E}_{x_{0:T}} \left[\log \frac{q(x_T|x_{T-1})}{q(x_T)} + \sum_{t=1}^{T-1} \underbrace{\log \frac{q(x_t|\boxed{x_{t-1}})}{p_\theta(x_t|\boxed{x_{t+1}})}}_{\text{forward vs. reverse}} - \log p_\theta(x_0|x_1) \right]$$

REDUCING THE VARIANCE

Use Markov property and Bayes' rule

$$q(x_t|x_{t-1}) = q(x_t|x_{t-1}, x_0) = \frac{q(x_{t-1}|x_t, x_0)q(x_t|x_0)}{q(x_{t-1}|x_0)}$$

REDUCING THE VARIANCE

Plug the previous equation back in

$$\mathcal{L} = \underbrace{\underbrace{\mathbb{E}_{x_{0:T}}\left[-\log\frac{q(x_T|x_0)}{q(x_T)}\right]}_{L_T}}_{\text{prior}} + \sum_{t=2}^{T}\underbrace{\underbrace{\mathbb{E}_{x_{0:T}}\left[\log\frac{q(x_{t-1}|x_t,x_0)}{p_\theta(x_{t-1}|x_t)}\right]}_{L_{t-1}}}_{\text{reverse vs. posterior of reverse}} - \underbrace{\underbrace{\mathbb{E}_{x_{0:T}}\left[\log p_\theta(x_0|x_1)\right]}_{L_0}}_{\text{recon.}}$$

THE LOSS FUNCTION

Plug the previous equation back in

$$\mathcal{L} = \underbrace{\mathbb{E}_{x_{0:T}}\left[-\log\frac{q(x_T|x_0)}{q(x_T)}\right]}_{L_T} + \sum_{t=2}^{T}\underbrace{\mathbb{E}_{x_{0:T}}\left[\log\boxed{\frac{q(x_{t-1}|x_t,x_0)}{p_\theta(x_{t-1}|x_t)}}\right]}_{L_{t-1}} - \underbrace{\mathbb{E}_{x_{0:T}}\left[\log p_\theta(x_0|x_1)\right]}_{L_0}$$

prior

reverse vs. posterior of reverse

recon.

MINIMIZING THE LOSS FUNCTION

Notice the KL divergence in the loss term

$$\mathcal{L} = \underbrace{\mathbb{E}_{x_{0:T}}\left[-\log \frac{q(x_T|x_0)}{q(x_T)}\right]}_{L_T} + \sum_{t=2}^{T} \underbrace{\mathbb{E}_{x_{0:T}}\left[\log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)}\right]}_{L_{t-1}} - \underbrace{\mathbb{E}_{x_{0:T}}\left[\log p_\theta(x_0|x_1)\right]}_{L_0}$$

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\, \mathrm{KL}\left(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)\right)\right]$$

HOW TO CALCULATE THE LOSS?

They are both Gaussian distributions

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t I\right)$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_t; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

WE CAN NOW SIMPLIFY

Write down the formula for the KL divergence

$$L_{t-1} \quad = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2}\left[\log\frac{|\Sigma_\theta|}{\tilde{\beta}_t} - d + \operatorname{tr}\left(\Sigma_\theta^{-1}\tilde{\beta}_t\right) + (\tilde{\mu} - \mu_\theta)^T \Sigma_\theta^{-1} (\tilde{\mu} - \mu_\theta)\right]\right]$$

WE CAN NOW SIMPLIFY

Write down the formula for the KL divergence

$$L_{t-1} = \mathbb{E}_{x_0,x_t}\left[\frac{1}{2}\left[\boxed{\log\frac{|\Sigma_\theta|}{\tilde{\beta}_t}} - d + \mathrm{tr}\left(\Sigma_\theta^{-1}\tilde{\beta}_t\right) + (\tilde{\mu} - \mu_\theta)^T\Sigma_\theta^{-1}(\tilde{\mu} - \mu_\theta)\right]\right]$$

WE CAN NOW SIMPLIFY

Write down the formula for the KL divergence

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2}\left[\boxed{\log\frac{|\Sigma_\theta|}{\tilde{\beta}_t}} - d + \boxed{\mathrm{tr}\left(\Sigma_\theta^{-1}\tilde{\beta}_t\right)} + (\tilde{\mu} - \mu_\theta)^T \Sigma_\theta^{-1} (\tilde{\mu} - \mu_\theta)\right]\right]$$

WE CAN NOW SIMPLIFY

Write down the formula for the KL divergence

$$L_{t-1} = \mathbb{E}_{x_0,x_t}\left[\frac{1}{2}\left[\boxed{\log\frac{|\Sigma_\theta|}{\tilde{\beta}_t}} - d + \boxed{\mathrm{tr}\left(\Sigma_\theta^{-1}\tilde{\beta}_t\right)} + \boxed{(\tilde{\mu}-\mu_\theta)^T\Sigma_\theta^{-1}(\tilde{\mu}-\mu_\theta)}\right]\right]$$

WE CAN NOW SIMPLIFY

Write down the formula for the KL divergence

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2}\left[\boxed{\log\frac{|\Sigma_\theta|}{\tilde{\beta}_t}} - d + \boxed{\mathrm{tr}\left(\Sigma_\theta^{-1}\tilde{\beta}_t\right)} + \boxed{(\tilde{\mu} - \mu_\theta)^T\Sigma_\theta^{-1}(\tilde{\mu} - \mu_\theta)}\right]\right]$$

WE CAN NOW SIMPLIFY

What assumption can we make to simplify this formula

?

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2}\left[\boxed{\log \frac{|\Sigma_\theta|}{\tilde{\beta}_t}} - d + \boxed{\operatorname{tr}\left(\Sigma_\theta^{-1}\tilde{\beta}_t\right)} + \boxed{(\tilde{\mu} - \mu_\theta)^T \Sigma_\theta^{-1} (\tilde{\mu} - \mu_\theta)}\right]\right]$$

WE CAN NOW SIMPLIFY

Just don't learn it!

$$L_{t-1} = \mathbb{E}_{x_0,x_t}\left[\frac{1}{2}\left[\boxed{\log\frac{|\Sigma_\theta|}{\tilde{\beta}_t}} - d + \boxed{\operatorname{tr}\left(\Sigma_\theta^{-1}\tilde{\beta}_t\right)} + \boxed{(\tilde{\mu}-\mu_\theta)^T\Sigma_\theta^{-1}(\tilde{\mu}-\mu_\theta)}\right]\right]$$

NEW LOSS FUNCTION

Assuming we don't learn the forward variance

$$\Sigma_\theta = \sigma_t^2$$

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}(x_t, x_0) - \mu_\theta(x_t, t)\|_2^2\right] + C$$

THE FORMULATIONS

Change the formulation of the model

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}(x_t, x_0) - \mu_\theta(x_t, t)\|_2^2\right] + C$$

image denoiser

noise predictor

score matching

IMAGE DENOISER

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2\sigma_t^2}\|\boxed{\tilde{\mu}(x_t, x_0)} - \mu_\theta(x_t, t)\|_2^2\right] + C$$

$$\boxed{\tilde{\mu}(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0}$$

IMAGE DENOISER

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}(x_t, x_0) - \mu_\theta(\boxed{x_t}, t)\|_2^2\right] + C$$

$$\tilde{\mu}(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boxed{x_t} + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0$$

input

IMAGE DENOISER

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}(x_t, x_0) - \boxed{\mu_\theta(x_t, t)}\|_2^2\right] + C$$

$$\tilde{\mu}(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0$$

$$\boxed{\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t\hat{x}_0(x_t, t)}{1-\bar{\alpha}_t}}$$

IMAGE DENOISER

$$L_{t-1} = \mathbb{E}_{x_0, x_t}\left[\frac{1}{2\sigma_t^2}\frac{\bar{\alpha}_{t-1}\beta_t^2}{(1-\bar{\alpha}_t)^2}\|x_0 - \hat{x}_0(x_t, t)\|_2^2\right]$$

NOISE PREDICTOR

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

NOISE PREDICTOR

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

$$x_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon}{\sqrt{\bar{\alpha}_t}}$$

NOISE PREDICTOR

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

$$x_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon}{\sqrt{\bar{\alpha}_t}}$$

$$\hat{x}_0(x_t, t) = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \hat{\epsilon}_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}$$

NOISE PREDICTOR

$$L_{t-1} = \mathbb{E}_{x_0, x_t} \left[\frac{1}{2\sigma_t^2} \frac{\beta_t^2}{(1-\bar{\alpha}_t)\alpha_t} \|\epsilon - \hat{\epsilon}_\theta(x_t, t)\|_2^2 \right]$$

SAMPLING

Just sample from the distribution!

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2)$$

Then follow the Markov Chain in reverse

SAMPLING

This is the simplest form of sampling. **Very slow**!

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

SCORE-MATCHING

$$X_t = \sqrt{1-\beta_t}X_{t-1} + \sqrt{\beta_t}\epsilon_t$$

Langevin dynamics for sampling from a known distribution.

$$X_t = X_t + \tau \nabla_X \log p\left(X_{t-1}\right) + \sqrt{2\tau}\epsilon_t$$

We can learn a **score** function

$$s_\theta\left(X_{t-1}\right) = \nabla_X \log p_\theta\left(X_{t-1}\right)$$

It can be written in terms of noise

$$\nabla_X \log p_\theta\left(X_t\right) = \frac{\epsilon_\theta}{\sqrt{1-\bar{\alpha}_t}}$$

SCORE-MATCHING

One can show:

$$L_{t-1} = \mathbb{E}_{x_0,x_t}\left[\frac{1}{2\sigma_t^2}\frac{\beta_t^2}{\alpha_t}\left\|\frac{\epsilon}{\sqrt{1-\bar{\alpha}_t}} + s_\theta(x_t, t)\right\|_2^2\right]$$

TRAINING ARCHITECTURE

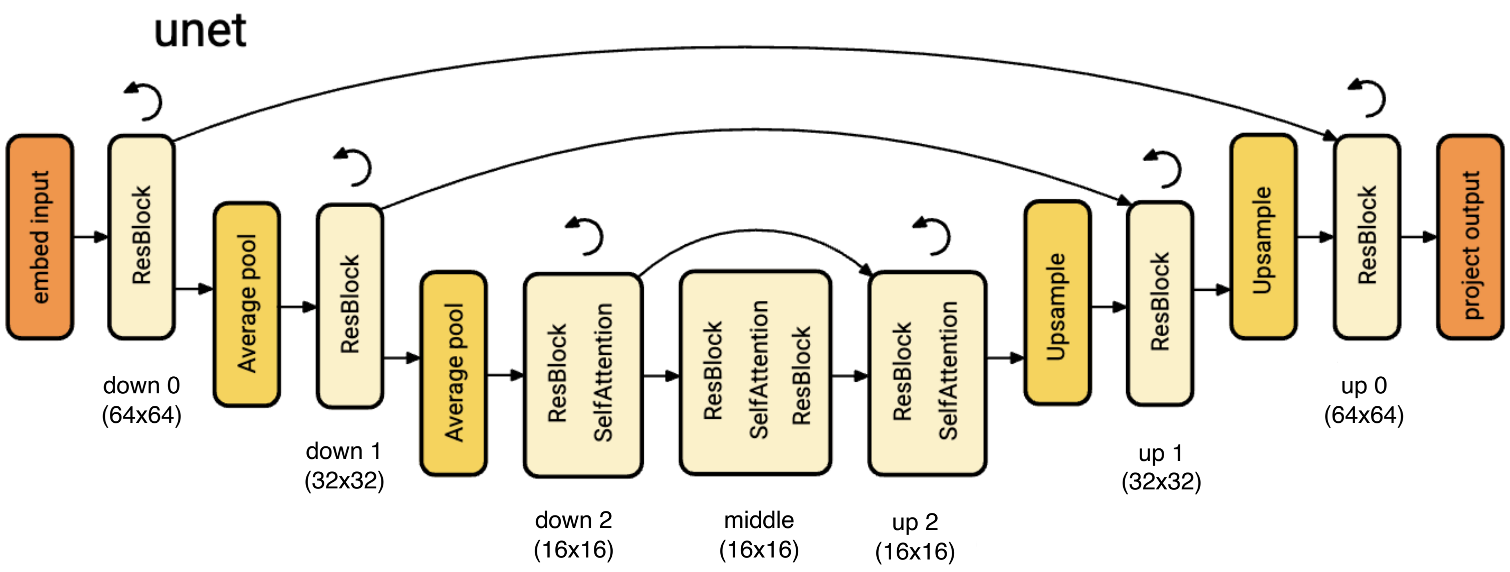


Image Credit: Hoogeboom, Emiel, Jonathan Heek, and Tim Salimans. "simple diffusion: End-to-end diffusion for high resolution images." International Conference on Machine Learning. PMLR, 2023.

TRAINING ARCHITECTURE

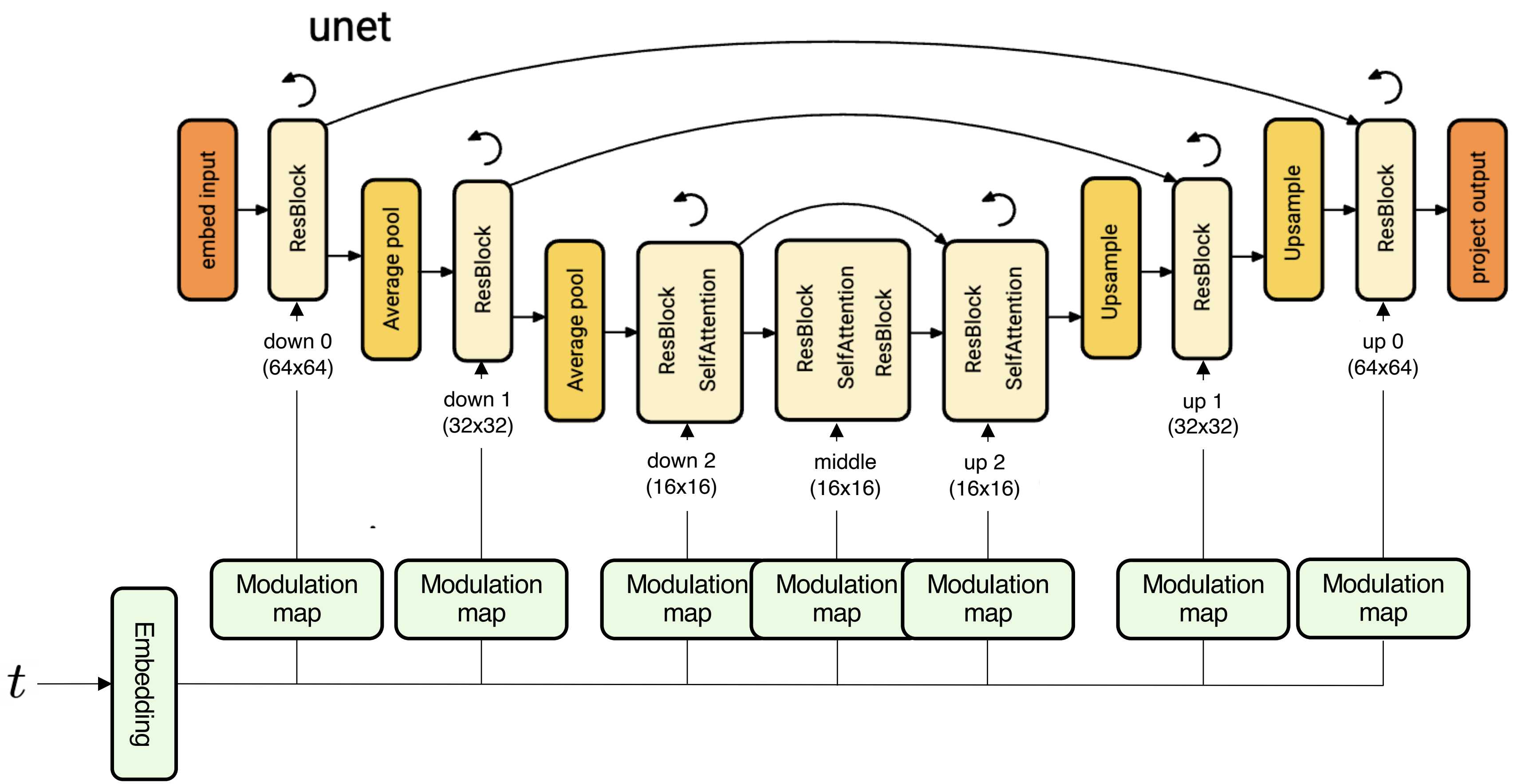


Image Credit: Hoogeboom, Emiel, Jonathan Heek, and Tim Salimans. "simple diffusion: End-to-end diffusion for high resolution images." International Conference on Machine Learning. PMLR, 2023.

SAMPLING IN PRACTICE

One can choose different samplers even when given the same trained model

Denoising Diffusion Implicit Models (DDIM) makes sampling deterministic

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}} \underbrace{\left(X_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(X_t, t)\right)}_{\text{predicted "image"}} + \underbrace{\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\epsilon_\theta(X_t, t)}_{\text{direction pointing towards single-step denoising}} + \sigma_t z$$

[1] Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." arXiv preprint arXiv:2010.02502 (2020).

SAMPLING IN PRACTICE

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}} \underbrace{\left(X_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(X_t, t)\right)}_{\text{predicted "image"}} + \underbrace{\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\,\epsilon_\theta(X_t, t)}_{\text{direction pointing towards single-step denoising}} + \sigma_t z$$

σ_t is the amount of randomness in the sampling

DDPM sampling

$$\sigma_t = \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_{t-1}}}$$

Deterministic

$$\sigma_t = 0$$

[1] Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." arXiv preprint arXiv:2010.02502 (2020).

SUMMARY

- Diffusion models balance **flexibility** and **tractability**.
- They minimize a version of the **ELBO** from VAEs (they are hierarchical VAEs with infinite layers).
- **Different formulations** can be obtained with only practical consequence, no theoretical difference in the loss optimized.
- **Sampling** can be seen separately from training and made deterministic.

BIBLIOGRAPHY

The fundamental papers

1. Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International conference on machine learning*. PMLR, 2015.

2. Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

3. Kingma, Diederik, et al. "Variational diffusion models." *Advances in neural information processing systems 34* (2021): 21696-21707.

4. Song, Yang, et al. "Score-Based Generative Modeling through Stochastic Differential Equations." *International Conference on Learning Representations*. 2020.

Tutorials

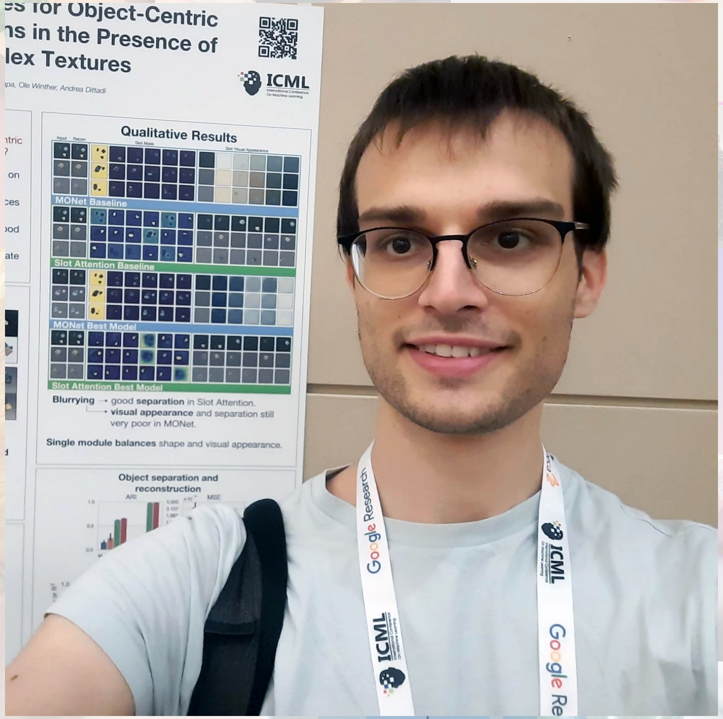
1. Karsten Kreis, Ruiqi Gao, Arash Vadat. "CVPR 2022 Tutorial: Denoising Diffusion-based Generative Modeling: Foundations and Applications." https://cvpr2022-tutorial-diffusion-models.github.io/

2. Lilian Weng." What are diffusion Models?" https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

3. Chan, Stanley H. "Tutorial on Diffusion Models for Imaging and Vision." *arXiv preprint arXiv:2403.18103* (2024).

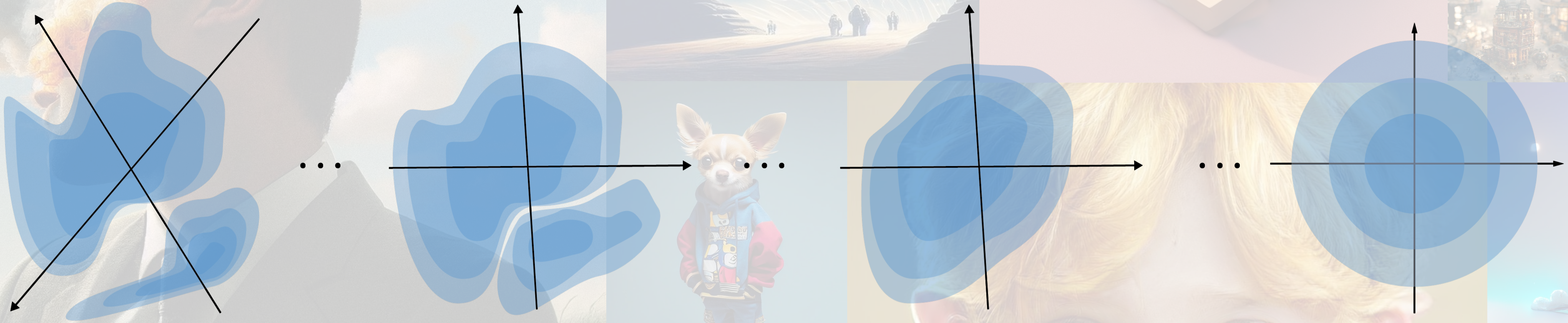
ABOUT ME

Website: https://samuelepapa.github.io/
LinkedIn: https://www.linkedin.com/in/samuelepapa/
Email: s.papa@uva.nl



Images from @midjourney.gallery

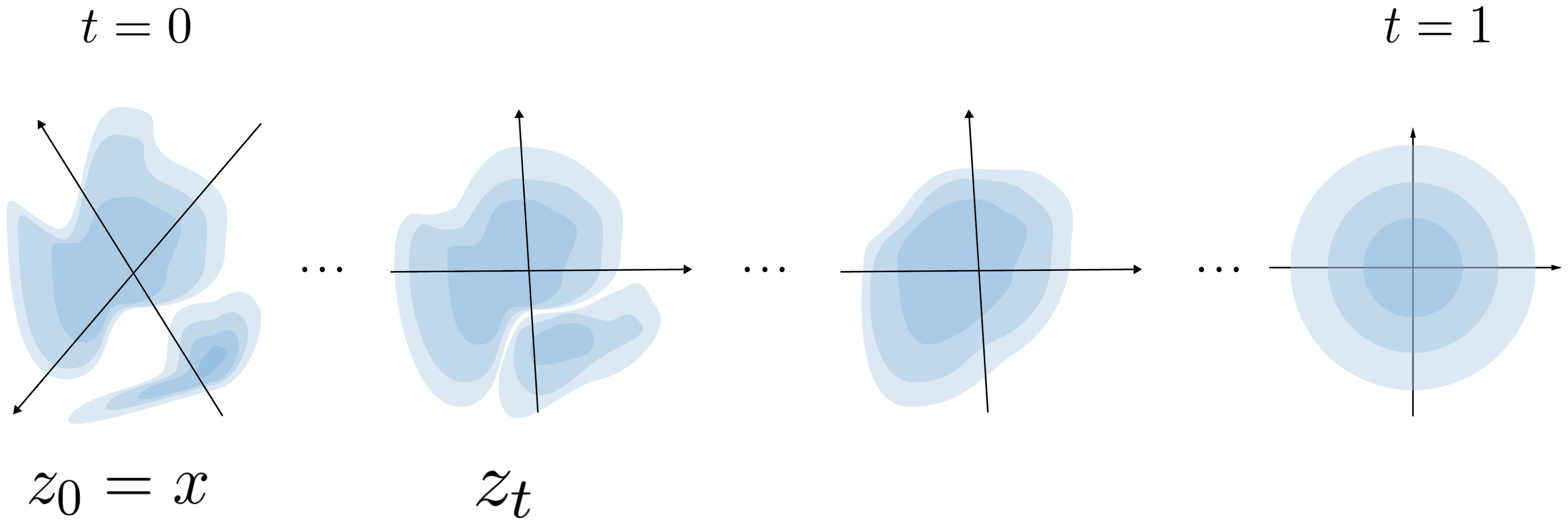
BONUS TOPICS



Images from @midjourney.gallery

SIMPLIFIED NOTATION

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}\left(\alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}\right)$$



[1] Kingma, Diederik, et al. "Variational diffusion models." Advances in neural information processing systems 34 (2021): 21696-21707.

LEARNED NOISE SCHEDULE

Define notation based on accumulated steps

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}\left(\alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}\right) \quad \mathrm{SNR}(t) = \alpha_t^2/\sigma_t^2$$

Learned noise schedule

$$\sigma_t^2 = \mathrm{sigmoid}(\gamma_{\boldsymbol{\eta}}(t))$$

$$\alpha_t^2 = \mathrm{sigmoid}(-\gamma_{\boldsymbol{\eta}}(t))$$

Simple expression for SNR

$$\mathrm{SNR}(t) = \exp(-\gamma_{\boldsymbol{\eta}}(t))$$

[1] Kingma, Diederik, et al. "Variational diffusion models." Advances in neural information processing systems 34 (2021): 21696-21707.

GENERAL FORMULATIONS

Loss function with new notation, VLB is the Variational Lower Bound

$$-\log p(\mathbf{x}) \leq -\text{VLB}(\mathbf{x}) = \underbrace{D_{KL}(q(\mathbf{z}_1|\mathbf{x})||p(\mathbf{z}_1))}_{\text{Prior loss}} + \underbrace{\mathbb{E}_{q(\mathbf{z}_0|\mathbf{x})}\left[-\log p(\mathbf{x}|\mathbf{z}_0)\right]}_{\text{Reconstruction loss}} + \underbrace{\mathcal{L}_T(\mathbf{x})}_{\text{Diffusion loss}}.$$

Discrete-time, i.e. $s(i) = (i-1)/T, t(i) = i/T$

Generic form

$$\mathcal{L}_T(\mathbf{x}) = \sum_{i=1}^{T} \mathbb{E}_{q(\mathbf{z}_{t(i)}|\mathbf{x})} D_{KL}[q(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}, \mathbf{x})||p(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)})].$$

Simplifies to

$$\mathcal{L}_T(\mathbf{x}) = \frac{T}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}), i\sim U\{1,T\}}\left[(\text{SNR}(s) - \text{SNR}(t))\,||\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)||_2^2\right]$$

[1] Kingma, Diederik, et al. "Variational diffusion models." Advances in neural information processing systems 34 (2021): 21696-21707.

DISCRETE-TIME

Discrete-time, i.e. $s(i) = (i-1)/T, t(i) = i/T$

Generic form

$$\mathcal{L}_T(\mathbf{x}) = \sum_{i=1}^{T} \mathbb{E}_{q(\mathbf{z}_{t(i)}|\mathbf{x})} D_{KL}[q(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}, \mathbf{x})||p(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)})].$$

Simplifies to

$$\mathcal{L}_T(\mathbf{x}) = \frac{T}{2} \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}), i\sim U\{1,T\}} \left[(\mathrm{SNR}(s) - \mathrm{SNR}(t)) \, ||\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)||_2^2 \right]$$

$$\mathcal{L}_T(\mathbf{x}) = \frac{T}{2} \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}), i\sim U\{1,T\}} \left[(\exp(\gamma_{\boldsymbol{\eta}}(t) - \gamma_{\boldsymbol{\eta}}(s)) - 1) \, \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\|_2^2 \right]$$

[1] Kingma, Diederik, et al. "Variational diffusion models." Advances in neural information processing systems 34 (2021): 21696-21707.

CONTINUOUS-TIME

Keep the timesteps continuous and take derivative of SNR w.r.t. time.

$$\begin{aligned}\mathcal{L}_\infty(\mathbf{x}) &= -\frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\int_0^1 \mathrm{SNR}'(t)\left\|\mathbf{x}-\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\right\|_2^2 dt,\\ &= -\frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}),t\sim\mathcal{U}(0,1)}\left[\mathrm{SNR}'(t)\left\|\mathbf{x}-\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\right\|_2^2\right]\end{aligned}$$

Simplifies to (note the analogy to discrete-time):

$$\mathcal{L}_\infty(\mathbf{x}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}),t\sim\mathcal{U}(0,1)}\left[\gamma'_{\boldsymbol{\eta}}(t)\left\|\boldsymbol{\epsilon}-\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\right\|_2^2\right]$$

[1] Kingma, Diederik, et al. "Variational diffusion models." Advances in neural information processing systems 34 (2021): 21696-21707.

EQUIVALENCE OF DIFFUSION MODELS

Let $v \equiv \mathrm{SNR}(t)$ and use this to change the variables in the continuous-time loss:

$$\tilde{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}, v) \equiv \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}, \mathrm{SNR}^{-1}(v))$$

$$\mathcal{L}_{\infty}(\mathbf{x}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})} \int_{\mathrm{SNR}_{\mathrm{min}}}^{\mathrm{SNR}_{\mathrm{max}}} \|\mathbf{x} - \tilde{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_v, v)\|_2^2 \, dv$$

The functions for the noise (aka the noise schedule) has no effect on the loss function itself, which is only dependent on the SNR at the start and end of the schedule.

Noise schedule still has an effect during training. This is because this perfect situation does not happen and the timesteps effectively sampled will affect how the model is trained.

[1] Kingma, Diederik, et al. "Variational diffusion models." Advances in neural information processing systems 34 (2021): 21696-21707.

PROGRESSIVE DISTILLATION

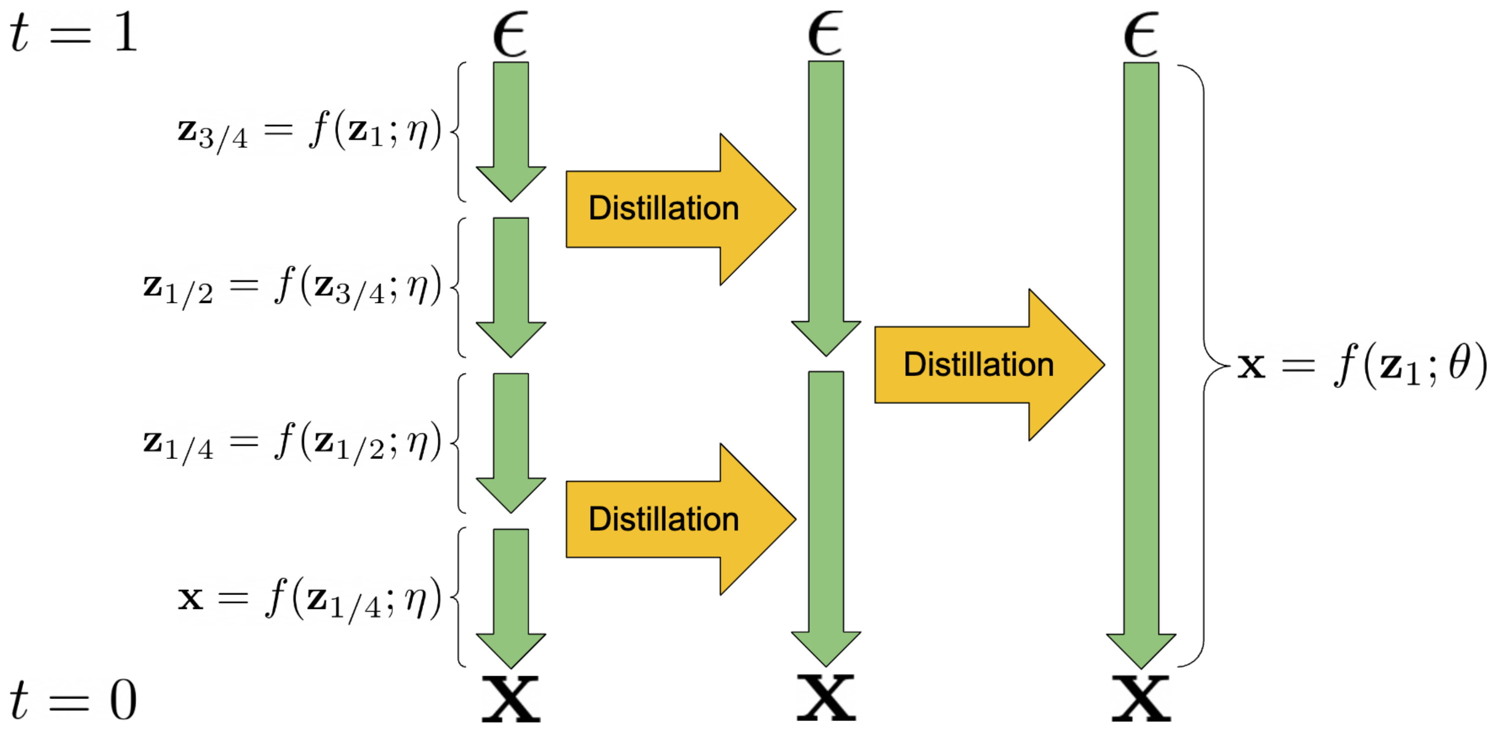


Figure 1: A visualization of two iterations of our proposed *progressive distillation* algorithm. A sampler $f(\mathbf{z}; \eta)$, mapping random noise ϵ to samples $\mathbf{x}$ in 4 deterministic steps, is distilled into a new sampler $f(\mathbf{z}; \theta)$ taking only a single step. The original sampler is derived by approximately integrating the *probability flow ODE* for a learned diffusion model, and distillation can thus be understood as learning to integrate in fewer steps, or *amortizing* this integration into the new sampler.

[1] Salimans, Tim, and Jonathan Ho. "Progressive distillation for fast sampling of diffusion models." arXiv preprint arXiv:2202.00512 (2022).

PROGRESSIVE DISTILLATION

Algorithm 1 Standard diffusion training

Require: Model $\hat{\mathbf{x}}_\theta(\mathbf{z}_t)$ to be trained
Require: Data set $\mathcal{D}$
Require: Loss weight function $w()$

while not converged **do**
- $\mathbf{x} \sim \mathcal{D}$ ▷ Sample data
- $t \sim U[0, 1]$ ▷ Sample time
- $\epsilon \sim N(0, I)$ ▷ Sample noise
- $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ ▷ Add noise to data
- $\tilde{\mathbf{x}} = \mathbf{x}$ ▷ Clean data is target for $\hat{\mathbf{x}}$
- $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$ ▷ log-SNR
- $L_\theta = w(\lambda_t)\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$ ▷ Loss
- $\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$ ▷ Optimization

end while

Algorithm 2 Progressive distillation

Require: Trained teacher model $\hat{\mathbf{x}}_\eta(\mathbf{z}_t)$
Require: Data set $\mathcal{D}$
Require: Loss weight function $w()$
Require: Student sampling steps N

for K iterations **do**
- $\theta \leftarrow \eta$ ▷ Init student from teacher
- **while** not converged **do**
  - $\mathbf{x} \sim \mathcal{D}$
  - $t = i/N, \quad i \sim Cat[1, 2, \ldots, N]$
  - $\epsilon \sim N(0, I)$
  - $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
  - `# 2 steps of DDIM with teacher`
  - $t' = t - 0.5/N, \quad t'' = t - 1/N$
  - $\mathbf{z}_{t'} = \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_t) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\eta(\mathbf{z}_t))$
  - $\mathbf{z}_{t''} = \alpha_{t''} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}_{t'} - \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}))$
  - $\tilde{\mathbf{x}} = \frac{\mathbf{z}_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
  - $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$
  - $L_\theta = w(\lambda_t)\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$
  - $\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$
- **end while**
- $\eta \leftarrow \theta$ ▷ Student becomes next teacher
- $N \leftarrow N/2$ ▷ Halve number of sampling steps

end for

[1] Salimans, Tim, and Jonathan Ho. "Progressive distillation for fast sampling of diffusion models." arXiv preprint arXiv:2202.00512 (2022).

CONSISTENCY MODELS

Multi-step sampling directly in the design of the model to trade-off speed and quality.

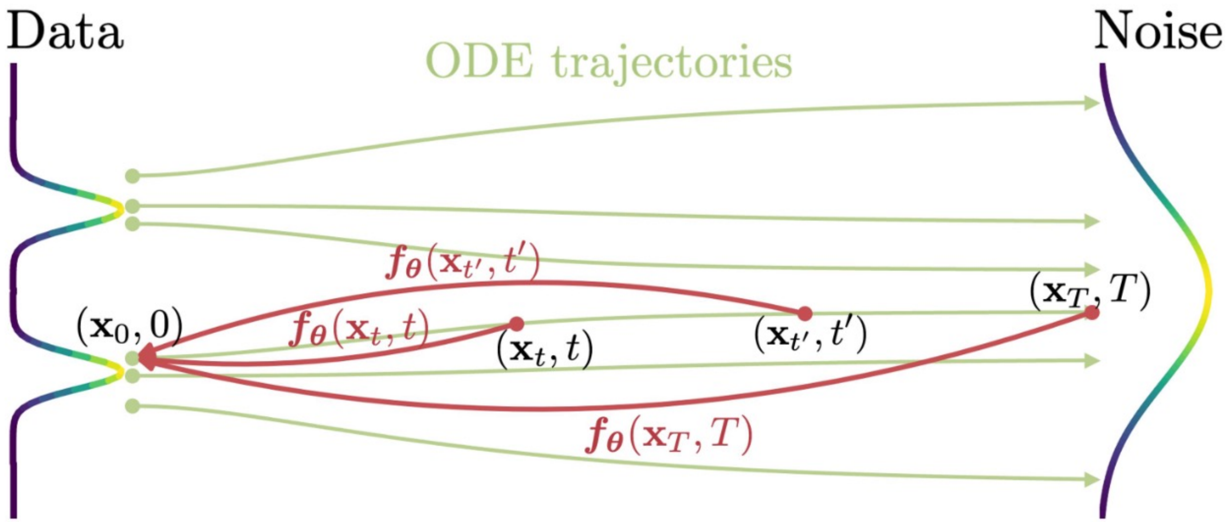


Figure 2: Consistency models are trained to map points on any trajectory of the PF ODE to the trajectory's origin.

[1] Salimans, Tim, and Jonathan Ho. "Progressive distillation for fast sampling of diffusion models." arXiv preprint arXiv:2202.00512 (2022).

CONSISTENCY MODELS

Algorithm 2 Consistency Distillation (CD)

Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, ODE solver $\Phi(\cdot, \cdot; \boldsymbol{\phi})$, $d(\cdot, \cdot)$, $\lambda(\cdot)$, and μ

$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$

repeat

  Sample $\mathbf{x} \sim \mathcal{D}$ and $n \sim \mathcal{U}[\![1, N-1]\!]$

  Sample $\mathbf{x}_{t_{n+1}} \sim \mathcal{N}(\mathbf{x}; t_{n+1}^2 \boldsymbol{I})$

  $\hat{\mathbf{x}}_{t_n}^{\boldsymbol{\phi}} \leftarrow \mathbf{x}_{t_{n+1}} + (t_n - t_{n+1})\Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}; \boldsymbol{\phi})$

  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi}) \leftarrow$

    $\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}_{t_n}^{\boldsymbol{\phi}}, t_n))$

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi})$

  $\boldsymbol{\theta}^- \leftarrow \operatorname{stopgrad}(\mu \boldsymbol{\theta}^- + (1-\mu)\boldsymbol{\theta})$

until convergence

Algorithm 3 Consistency Training (CT)

Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, step schedule $N(\cdot)$, EMA decay rate schedule $\mu(\cdot)$, $d(\cdot, \cdot)$, and $\lambda(\cdot)$

$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$ and $k \leftarrow 0$

repeat

  Sample $\mathbf{x} \sim \mathcal{D}$, and $n \sim \mathcal{U}[\![1, N(k)-1]\!]$

  Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-) \leftarrow$

    $\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x} + t_{n+1}\mathbf{z}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\mathbf{x} + t_n \mathbf{z}, t_n))$

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-)$

  $\boldsymbol{\theta}^- \leftarrow \operatorname{stopgrad}(\mu(k)\boldsymbol{\theta}^- + (1-\mu(k))\boldsymbol{\theta})$

  $k \leftarrow k + 1$

until convergence

[1] Salimans, Tim, and Jonathan Ho. "Progressive distillation for fast sampling of diffusion models." arXiv preprint arXiv:2202.00512 (2022).

FREQUENCY INTERPRETATION

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

FREQUENCY INTERPRETATION

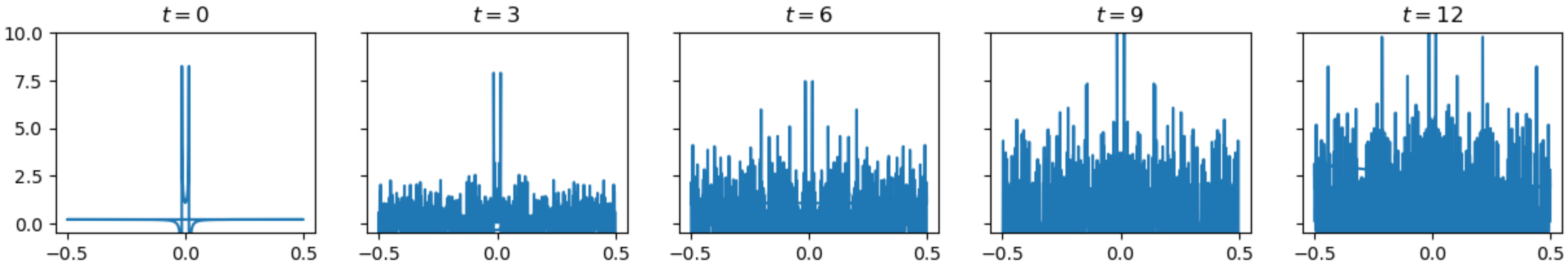
$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

$$\mathcal{F}[x_t] = \sqrt{\bar{\alpha}_t} \mathcal{F}[x_0] + \sqrt{1 - \bar{\alpha}_t} \mathcal{F}[\epsilon]$$

FREQUENCY INTERPRETATION

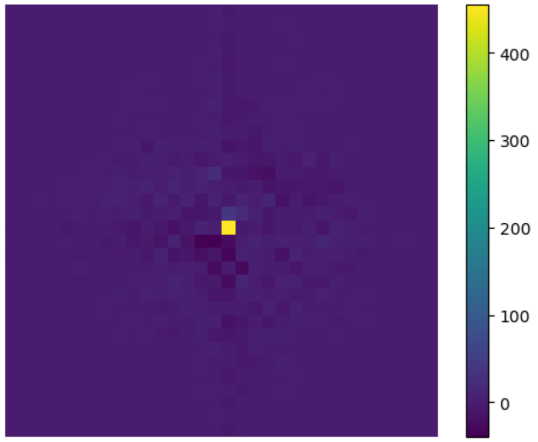
$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

$$\mathcal{F}[x_t] = \sqrt{\bar{\alpha}_t} \mathcal{F}[x_0] + \sqrt{1 - \bar{\alpha}_t} \mathcal{F}[\epsilon]$$



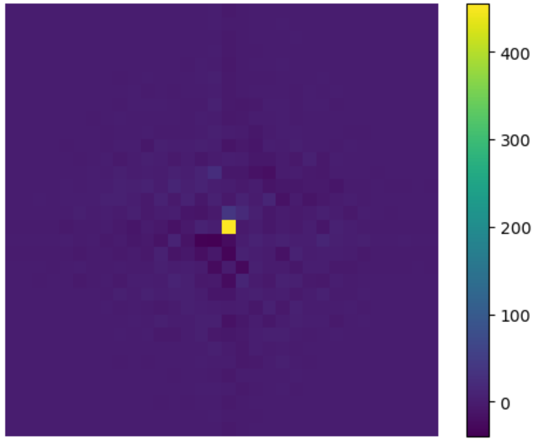
FREQUENCY INTERPRETATION

FREQUENCY INTERPRETATION



1. Find the 2D Fourier Transform

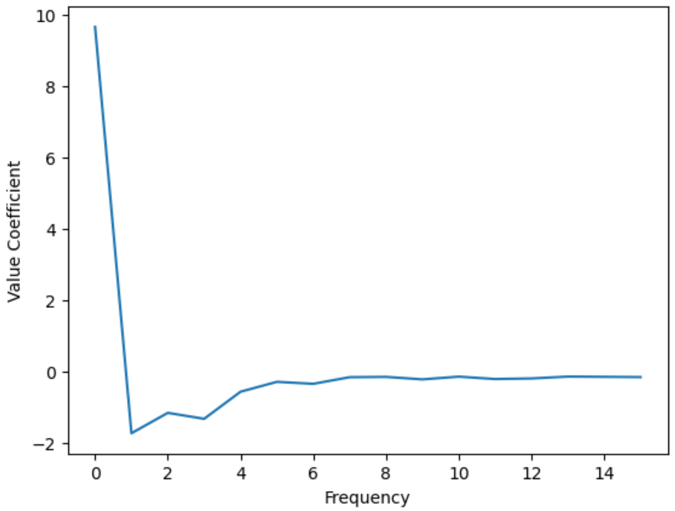
FREQUENCY INTERPRETATION



1. Find the 2D Fourier Transform

2. Average power across noise samples

FREQUENCY INTERPRETATION



1. Find the 2D Fourier Transform

2. Average power across noise samples

3. Average across height to get 1D plot

FREQUENCY INTERPRETATION

1. Find the 2D Fourier Transform

2. Average power across noise samples

3. Average across height to get 1D plot