EDM (Elucidating the Design Space of Diffusion-Based Generative Models)

Suk-Hwan Lee

Artificial Intelligence

Creating the Future

Dong-A University

Division of Computer Engineering & Artificial Intelligence

References

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, (nvidia)

"**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

https://arxiv.org/abs/2206.00364

https://github.com/NVlabs/edm

https://www.youtube.com/watch?v=T0Qxzf0eaio

https://research.nvidia.com/publication/2022-11_elucidating-design-space-diffusion-based-generative-models

Blog

https://sang-yun-lee.notion.site/Elucidating-the-Design-Space-of-Diffusion-Based-Generative-Models-a81b14bf297743ec90a72c11f0fbce57

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Youtube Presentaiton

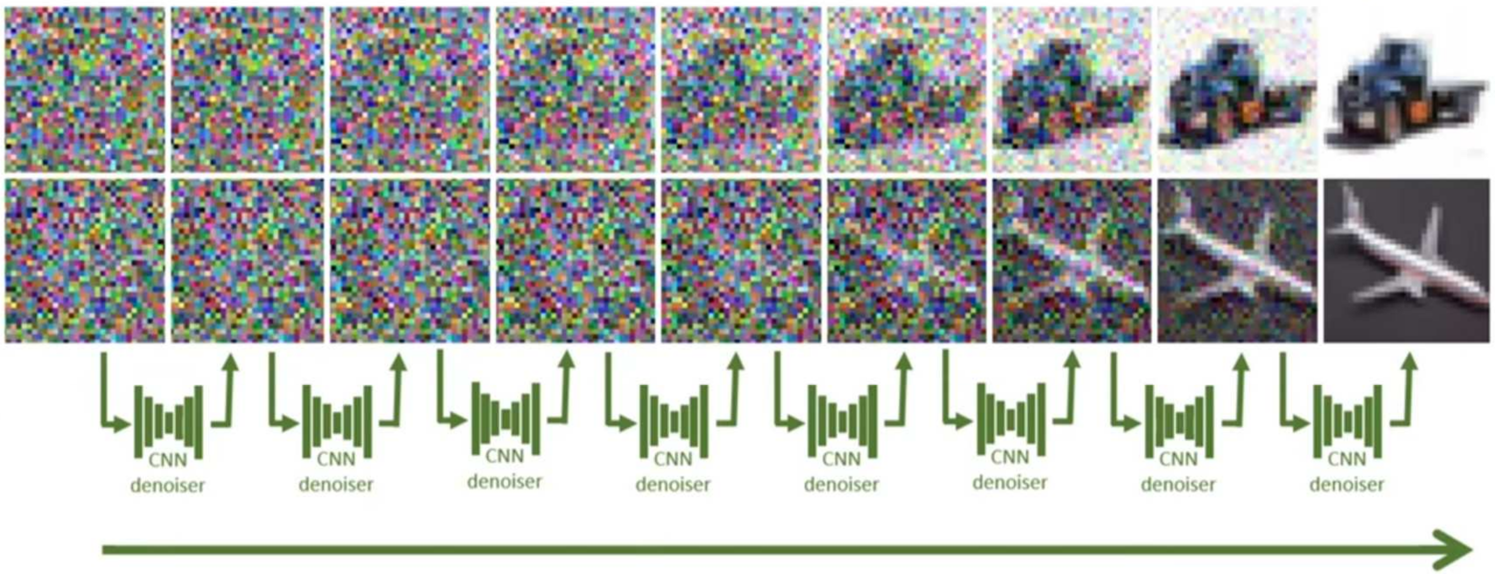
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Intuition on denoising diffusion

2) Quality of individual images, and variety?

Tradeoff with the quality of the individual generated images and with the distribution as whole

start from random noise



1) Efficiency: typically 50 to 2000 network evaluations – slow!

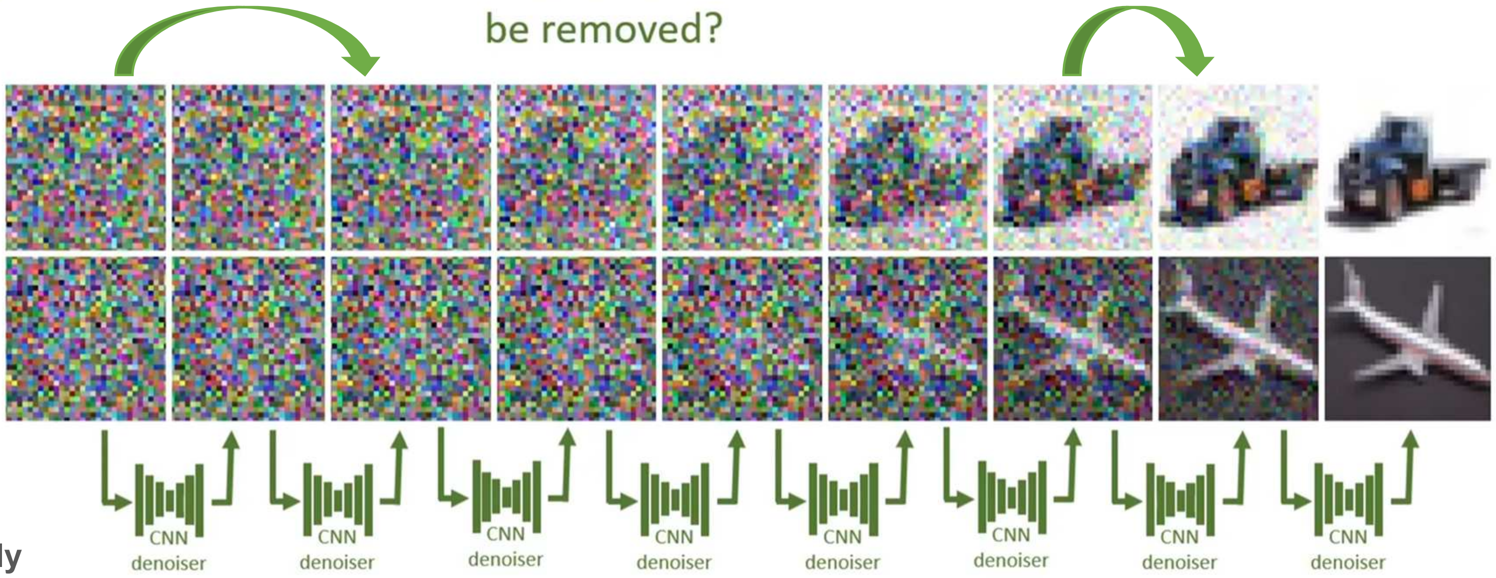
gradually removing noise reveals a random image – *a* ***generative model***!

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Deterministically "fade out" noise, vs. replace it?

On what schedule should noise be removed?



Differ vastly in Practical design choices like at what rate do you reduce the noise level at different stages of the generation

Deterministically or stastically

How do you deal with the vastly different signal magnitudes at different stages of this process and how do you predict the signal or the noise

Large numerical scale of the noise?
Normalization?
Where to implement it?

Predict noise or clean image?
Training effort per noise level?

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Differential equation formalism

by Y. Song et al. (2021)

- Image evolves according to a **stochastic differential equation (SDE)**
- Also deterministic **ordinary differential equation (ODE)** variant

- Generalizes existing methods, in principle

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.

Outline

- Part I: **Common framework**
  - Identifying the moving parts in existing work
- Part II: **Deterministic sampling**
  - Solving the ODE efficiently
- Part III: **Stochastic sampling**
  - Why SDE's? How to do stochastic stepping?
- Part IV: **Preconditioning and training**
  - How to train the CNN used in evaluating a step?

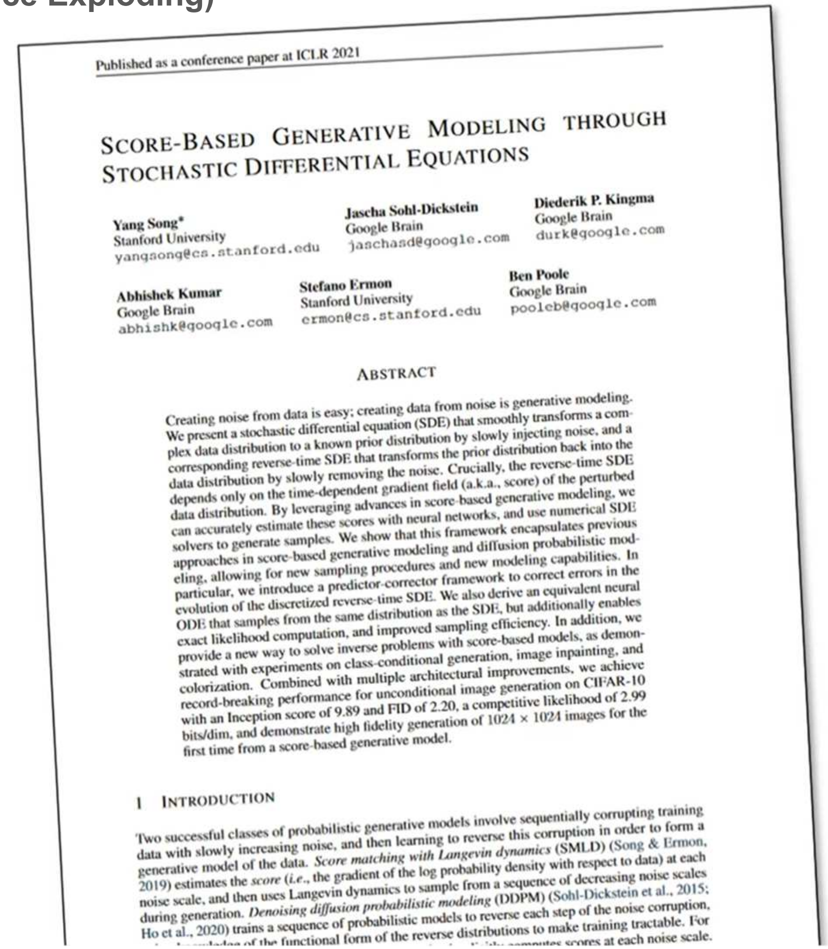
- We will **not** study network architectures (what layers to use, etc)!

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

VP (Variance Preserving)
VE (Variance Exploding)

VP VE


Published as a conference paper at ICLR 2021

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song* Stanford University yangsong@cs.stanford.edu
Jascha Sohl-Dickstein Google Brain jaschasd@google.com
Diederik P. Kingma Google Brain durk@google.com
Abhishek Kumar Google Brain abhishk@google.com
Stefano Ermon Stanford University ermon@cs.stanford.edu
Ben Poole Google Brain pooleb@google.com

ABSTRACT

Creating noise from data is easy; creating data from noise is generative modeling. We present a stochastic differential equation (SDE) that smoothly transforms a complex data distribution to a known prior distribution by slowly injecting noise, and a corresponding reverse-time SDE that transforms the prior distribution back into the data distribution by slowly removing the noise. Crucially, the reverse-time SDE depends only on the time-dependent gradient field (a.k.a., score) of the perturbed data distribution. By leveraging advances in score-based generative modeling, we can accurately estimate these scores with neural networks, and use numerical SDE solvers to generate samples. We show that this framework encapsulates previous approaches in score-based generative modeling and diffusion probabilistic modeling, allowing for new sampling procedures and new modeling capabilities. In particular, we introduce a predictor-corrector framework to correct errors in the evolution of the discretized reverse-time SDE. We also derive an equivalent neural ODE that samples from the same distribution as the SDE, but additionally enables exact likelihood computation, and improved sampling efficiency. In addition, we provide a new way to solve inverse problems with score-based models, as demonstrated with experiments on class-conditional generation, image inpainting, and colorization. Combined with multiple architectural improvements, we achieve record-breaking performance for unconditional image generation on CIFAR-10 with an Inception score of 9.89 and FID of 2.20, a competitive likelihood of 2.99 bits/dim, and demonstrate high fidelity generation of 1024 × 1024 images for the first time from a score-based generative model.

1 INTRODUCTION

Two successful classes of probabilistic generative models involve sequentially corrupting training data with slowly increasing noise, and then learning to reverse this corruption in order to form a generative model of the data. *Score matching with Langevin dynamics* (SMLD) (Song & Ermon, 2019) estimates the *score* (*i.e.*, the gradient of the log probability density with respect to data) at each noise scale, and then uses Langevin dynamics to sample from a sequence of decreasing noise scales during generation. *Denoising diffusion probabilistic modeling* (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) trains a sequence of probabilistic models to reverse each step of the noise corruption. For

iDDPM + DDIM


Published as a conference paper at ICLR 2021

DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

ABSTRACT

Denoising diffusion probabilistic models (DDPMs) ...
ity image generation without ...
Markov chain for ...

Improved Denoising Diffusion Probabilistic Models

Alex Nichol * [1] Prafulla Dhariwal * [1]

Abstract

Denoising diffusion probabilistic models (DDPM) are a class of generative models which have recently been shown to produce excellent samples. We show that with a few simple modifications, DDPMs can also achieve competitive log-likelihoods while maintaining high sample quality. Additionally, we find that learning variances of the reverse diffusion process allows sampling with an order of magnitude fewer forward passes with a negligible difference in sample quality, which is important for the practical deployment of these models. We additionally use precision and recall to compare how well DDPMs and GANs cover the target distribution. Finally, we show that the sample quality and likelihood of these models scale smoothly with model capacity and training compute, making them easily scalable. We release our code and pre-trained https://github.com/openai/

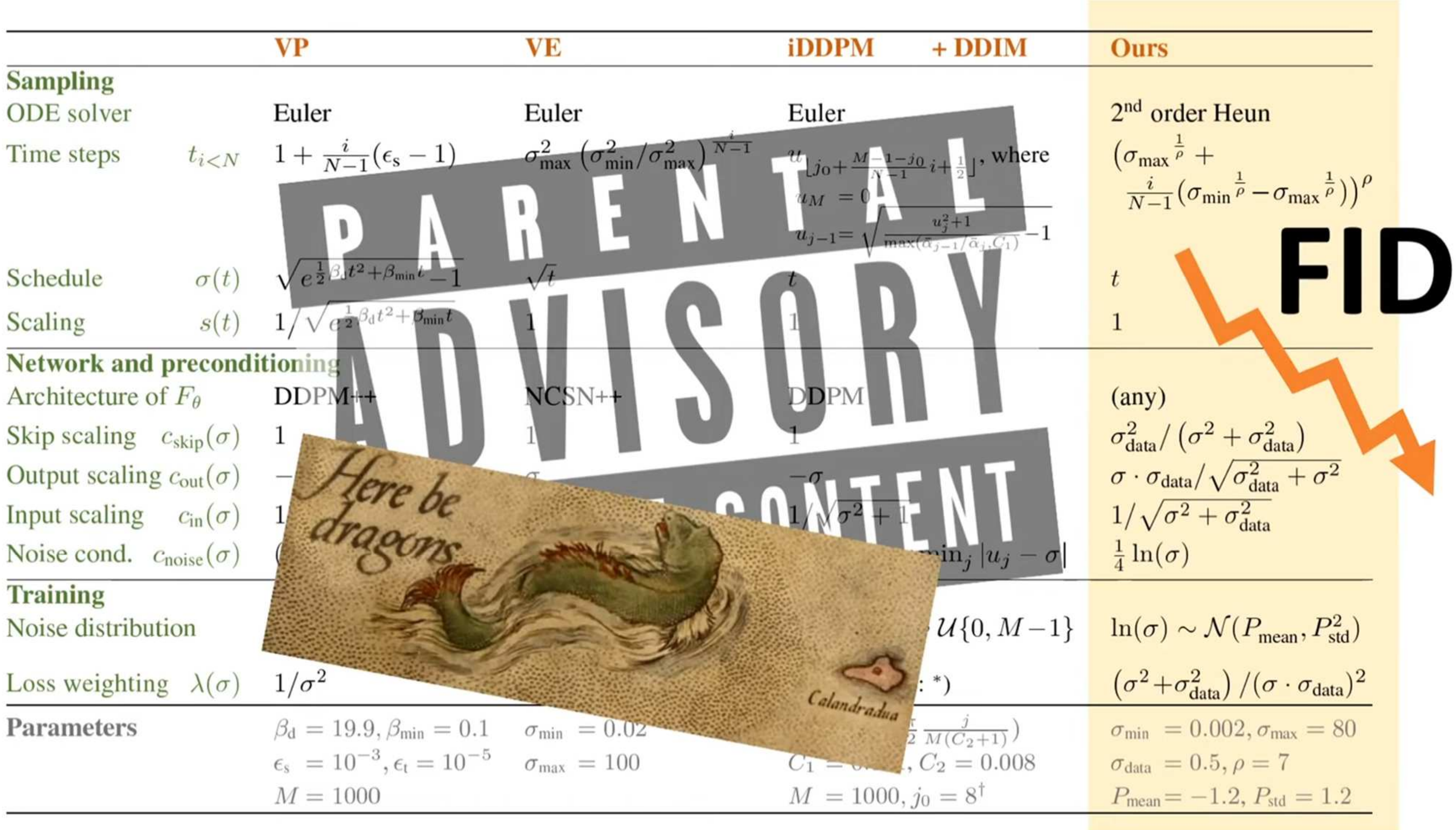
and VAEs (Kingma & Welling, 2013). This raises various questions, such as whether DDPMs are capable of capturing all the modes of a distribution. Furthermore, while Ho et al. (2020) showed extremely good results on the CIFAR-10 (Krizhevsky, 2009) and LSUN (Yu et al., 2015) datasets, it is unclear how well DDPMs scale to datasets with higher diversity such as ImageNet. Finally, while Chen et al. (2020b) found that DDPMs can efficiently generate audio using a small number of sampling steps, it has yet to be shown that the same is true for images.

In this paper, we show that DDPMs can achieve log-likelihoods competitive with other likelihood-based models, even on high-diversity datasets like ImageNet. To more tightly optimise the variational lower-bound (VLB), we learn the reverse process variances using a simple reparameterization and a hybrid learning objective that combines the VLB with the simplified objective from Ho et al. (2020).

We find surprisingly that, with our hybrid objective, our models obtain better log-likelihoods than those obtained by optimizing the log-likelihood directly, and discover that the latter objective has much more gradient noise during ...

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

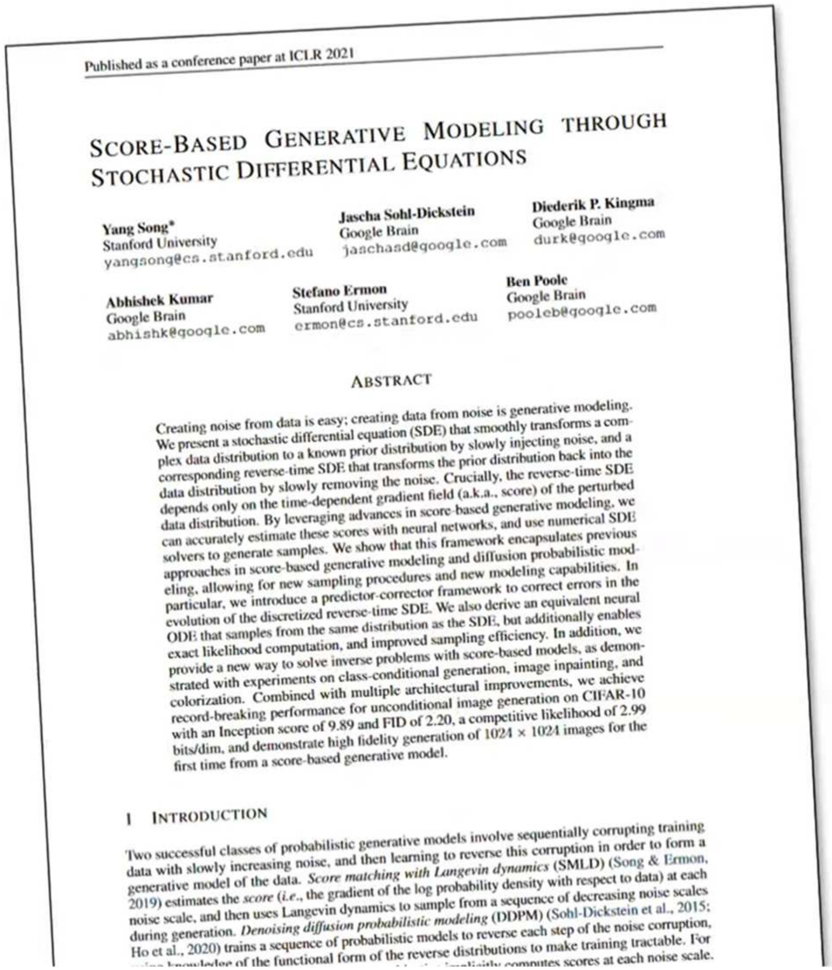


		VP	VE	iDDPM + DDIM	Ours
Sampling					
ODE solver		Euler	Euler	Euler	2nd order Heun
Time steps	$t_{i<N}$	$1+\frac{i}{N-1}(\epsilon_s-1)$	$\sigma_{\max}^2\left(\sigma_{\min}^2/\sigma_{\max}^2\right)^{\frac{i}{N-1}}$	$u_{\lfloor j_0+\frac{M-1-j_0}{N-1}i+\frac{1}{2}\rfloor}$, where $u_M=0$, $u_{j-1}=\sqrt{\frac{u_j^2+1}{\max(\bar\alpha_{j-1}/\bar\alpha_j,C_1)}-1}$	$\left(\sigma_{\max}{}^{\frac{1}{\rho}}+\frac{i}{N-1}\left(\sigma_{\min}{}^{\frac{1}{\rho}}-\sigma_{\max}{}^{\frac{1}{\rho}}\right)\right)^{\rho}$
Schedule	$\sigma(t)$	$\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}-1}$	$\sqrt{t}$	t	t
Scaling	$s(t)$	$1/\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}}$	1	1	1
Network and preconditioning					
Architecture of F_θ		DDPM++	NCSN++	DDPM	(any)
Skip scaling	$c_{\text{skip}}(\sigma)$	1	1	1	$\sigma_{\text{data}}^2/\left(\sigma^2+\sigma_{\text{data}}^2\right)$
Output scaling	$c_{\text{out}}(\sigma)$	[illegible]	[illegible]	$-\sigma$	$\sigma\cdot\sigma_{\text{data}}/\sqrt{\sigma_{\text{data}}^2+\sigma^2}$
Input scaling	$c_{\text{in}}(\sigma)$	[illegible]	[illegible]	$1/\sqrt{\sigma^2+1}$	$1/\sqrt{\sigma^2+\sigma_{\text{data}}^2}$
Noise cond.	$c_{\text{noise}}(\sigma)$	[illegible]	[illegible]	[illegible] $\min_j \lvert u_j-\sigma\rvert$	$\frac{1}{4}\ln(\sigma)$
Training					
Noise distribution		[illegible]	[illegible]	$\mathcal{U}\{0,M-1\}$	$\ln(\sigma)\sim\mathcal{N}(P_{\text{mean}},P_{\text{std}}^2)$
Loss weighting	$\lambda(\sigma)$	$1/\sigma^2$	[illegible]	[illegible] *)	$\left(\sigma^2+\sigma_{\text{data}}^2\right)/(\sigma\cdot\sigma_{\text{data}})^2$
Parameters		$\beta_d=19.9, \beta_{\min}=0.1$ $\epsilon_s=10^{-3}, \epsilon_t=10^{-5}$ $M=1000$	$\sigma_{\min}=0.02$ $\sigma_{\max}=100$	[illegible] $\frac{j}{M(C_2+1)}$) $C_1=$ [illegible], $C_2=0.008$ $M=1000, j_0=8^{\dagger}$	$\sigma_{\min}=0.002, \sigma_{\max}=80$ $\sigma_{\text{data}}=0.5, \rho=7$ $P_{\text{mean}}=-1.2, P_{\text{std}}=1.2$

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

VP VE



Published as a conference paper at ICLR 2021

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song*
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

Abhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

ABSTRACT

Creating noise from data is easy; creating data from noise is generative modeling. We present a stochastic differential equation (SDE) that smoothly transforms a complex data distribution to a known prior distribution by slowly injecting noise, and a corresponding reverse-time SDE that transforms the prior distribution back into the data distribution by slowly removing the noise. Crucially, the reverse-time SDE depends only on the time-dependent gradient field (a.k.a., score) of the perturbed data distribution. By leveraging advances in score-based generative modeling, we can accurately estimate these scores with neural networks, and use numerical SDE solvers to generate samples. We show that this framework encapsulates previous approaches in score-based generative modeling and diffusion probabilistic modeling, allowing for new sampling procedures and new modeling capabilities. In particular, we introduce a predictor-corrector framework to correct errors in the evolution of the discretized reverse-time SDE. We also derive an equivalent neural ODE that samples from the same distribution as the SDE, but additionally enables exact likelihood computation, and improved sampling efficiency. In addition, we provide a new way to solve inverse problems with score-based models, as demonstrated with experiments on class-conditional generation, image inpainting, and colorization. Combined with multiple architectural improvements, we achieve record-breaking performance for unconditional image generation on CIFAR-10 with an Inception score of 9.89 and FID of 2.20, a competitive likelihood of 2.99 bits/dim, and demonstrate high fidelity generation of 1024 × 1024 images for the first time from a score-based generative model.

1 INTRODUCTION

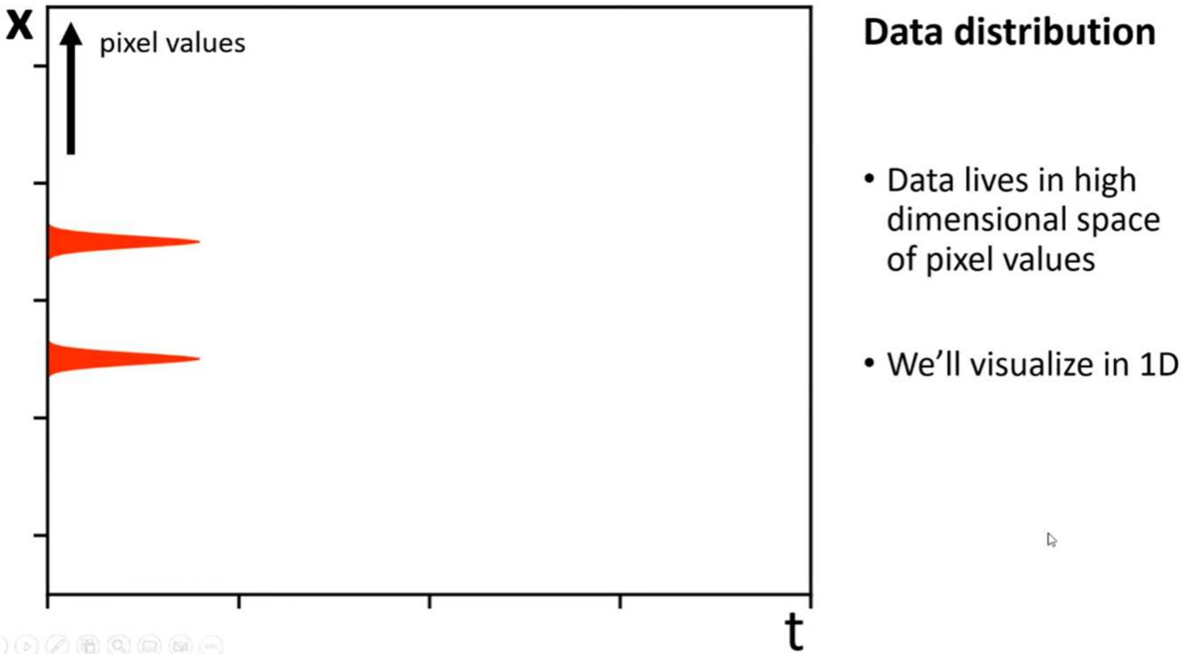
Two successful classes of probabilistic generative models involve sequentially corrupting training data with slowly increasing noise, and then learning to reverse this corruption in order to form a generative model of the data. *Score matching with Langevin dynamics* (SMLD) (Song & Ermon, 2019) estimates the *score* (*i.e.*, the gradient of the log probability density with respect to data) at each noise scale, and then uses Langevin dynamics to sample from a sequence of decreasing noise scales during generation. *Denoising diffusion probabilistic modeling* (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) trains a sequence of probabilistic models to reverse each step of the noise corruption. For

For background, let's review how Song et al. (2021) formulate denoising diffusion SDE/ODE

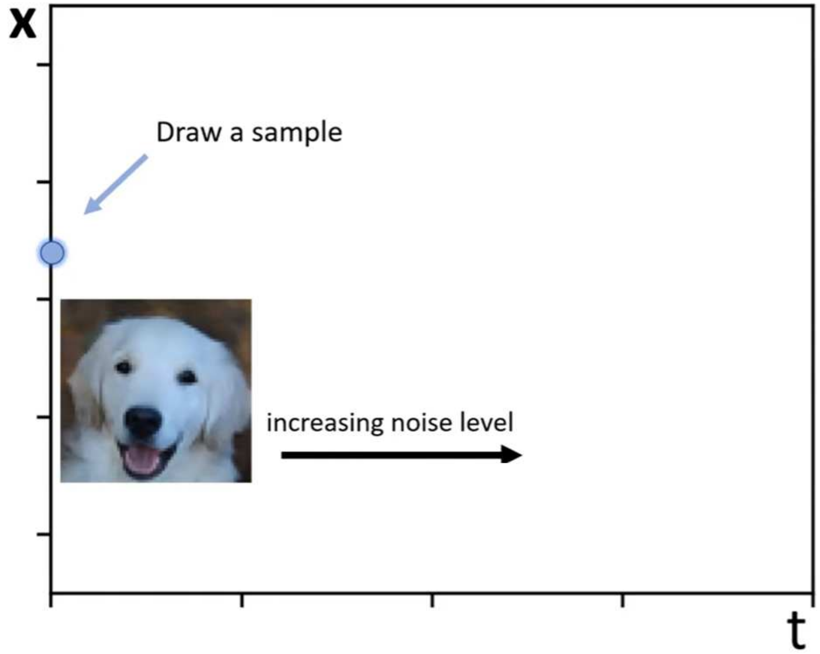
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Milliondimensional data space
Data distribution from Dataset



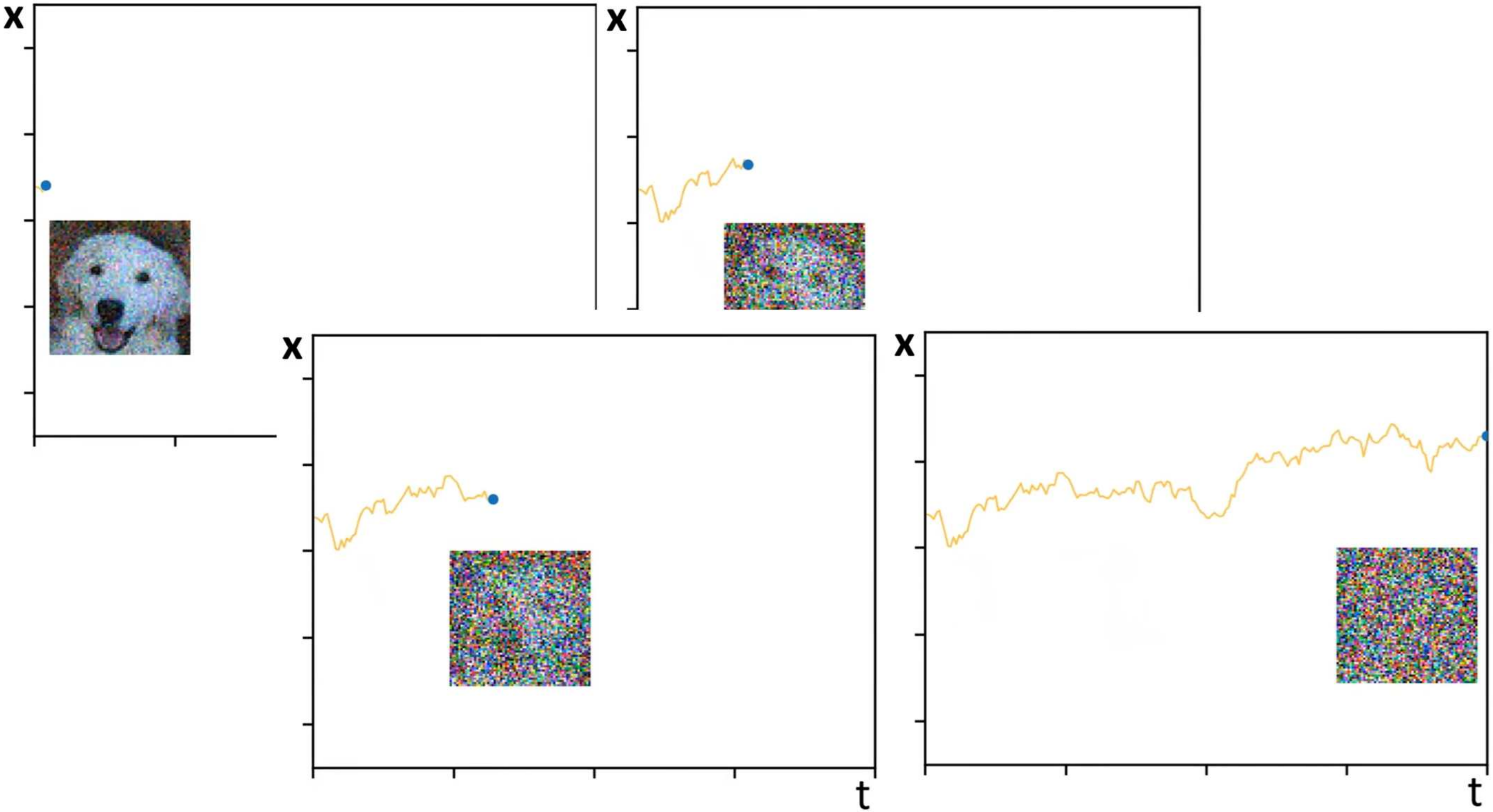
Learn to produce novel samples from this distribution



Increasing time which is an essential increasing noise level

Youtube Presentaiton

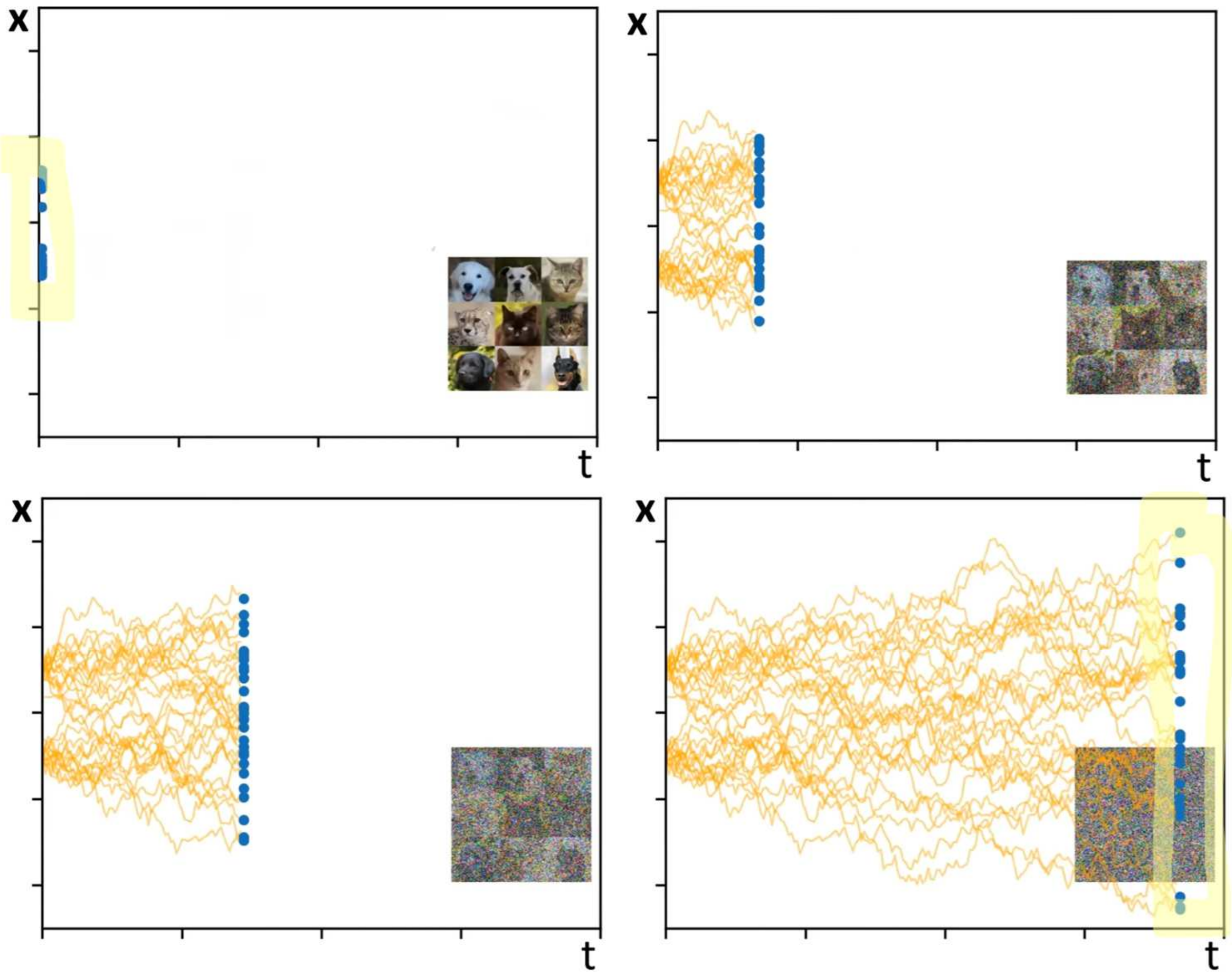
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models]," NeurIPS 2022.



Gradually add noise to destroy the image

Youtube Presentaiton

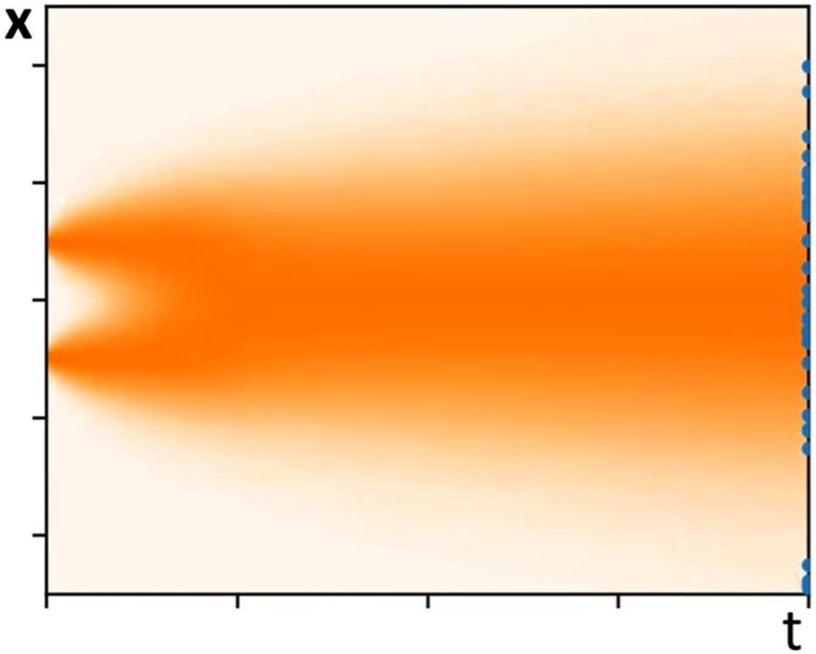
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

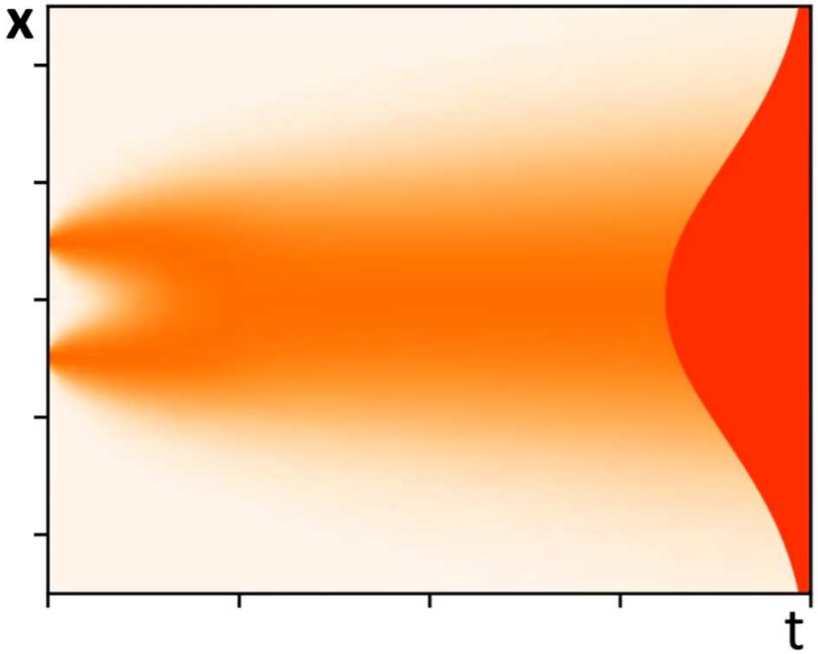


Gradually add noise to destroy the image

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.





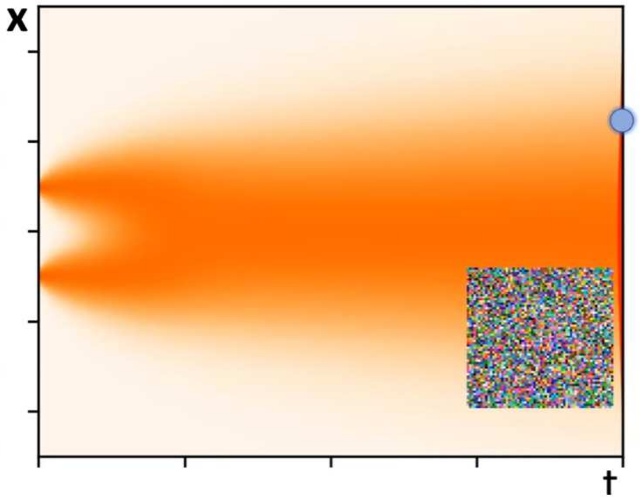
Data density is *diffused* over time

- All data ends up normally distributed

- **The density of the data on the left edge becomes diffused over time until it's completely normally distributed at the end.**
- **We can sample from this normal distribution at the right edge we just call random in pytorch.**

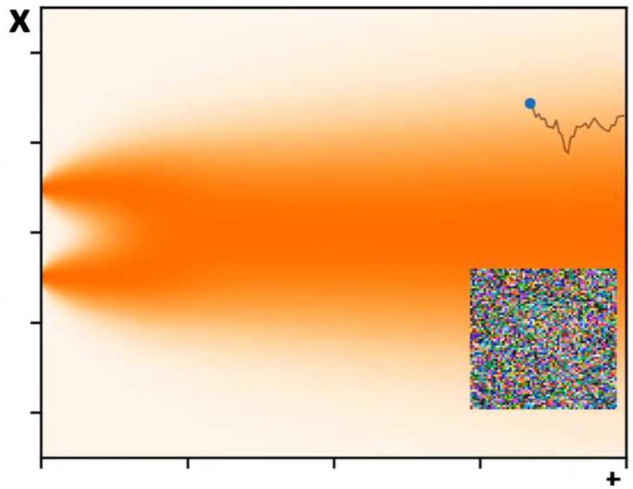
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models]," NeurIPS 2022.

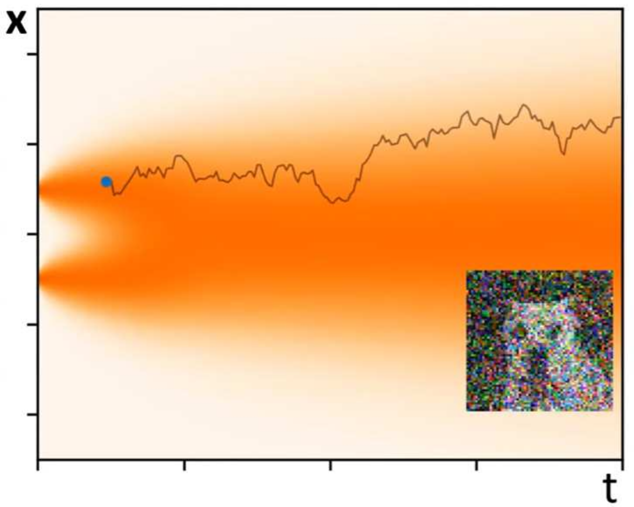


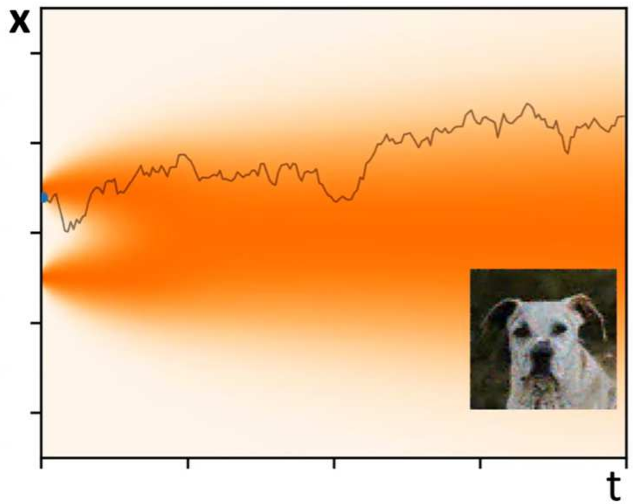
Data density is *diffused* over time

- All data ends up normally distributed
- Let's draw a sample...



We can reverse the trajectory by gradual denoising

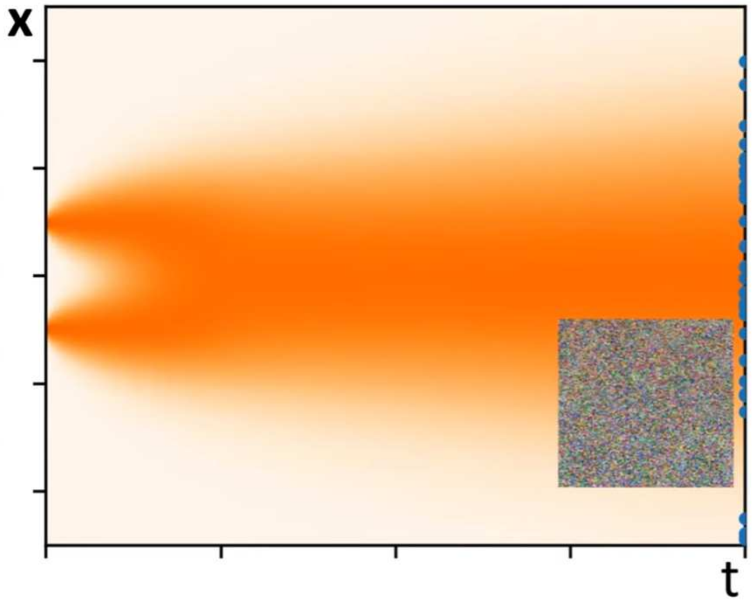




- **They'll give us a sample from that edge.**
- **There exist to sort of reverse this path so go backward in time.**
- **Land us on the left edge that have the density of the actual data - Generate an image**

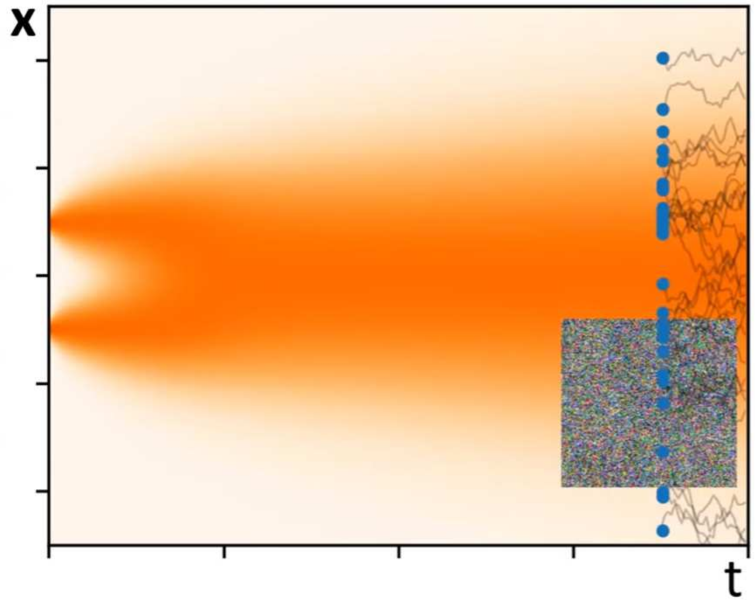
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



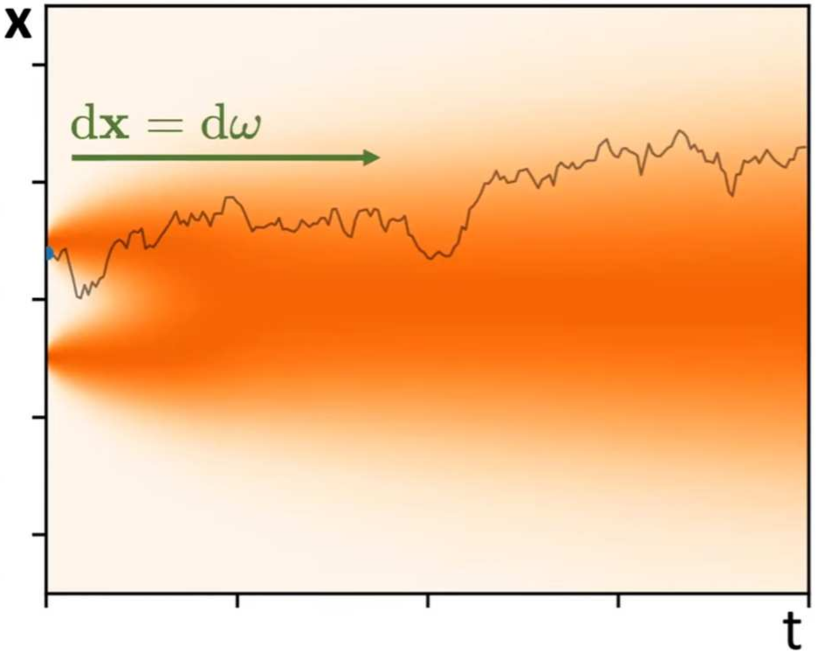
We can *reverse* the trajectory by gradual denoising

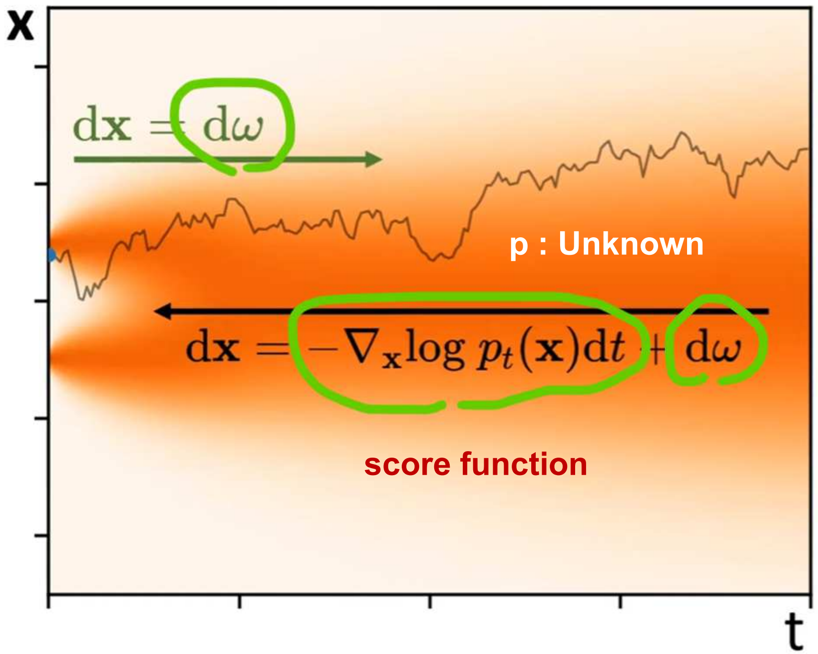
- Endpoint is a random sample from data distribution!



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.





Forward and reverse SDE

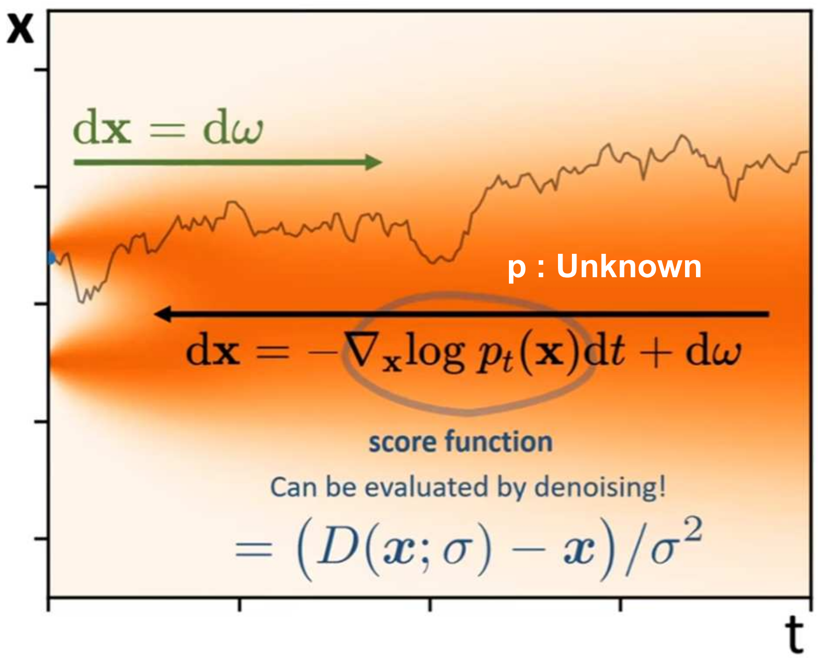
- Evolution governed by a *stochastic differential equation* (SDE)
- We'll generalize these later!

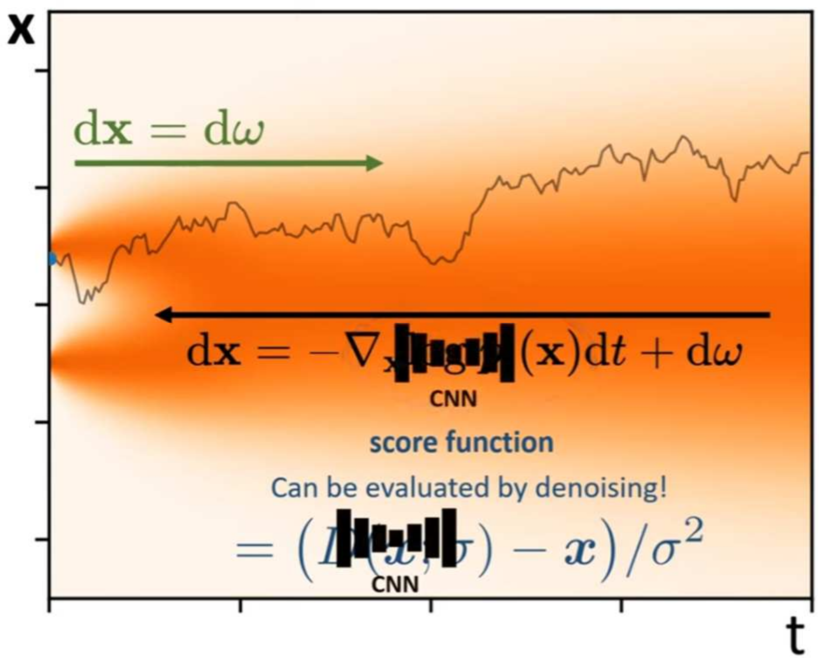
- **The change in image dx, equals dw which is a white noise so that's just the mathematical expression of doing cumulative sum of random noise.**

- **This forward equation corresponds a backward version that has this same stochastic component random walk component.**
- **Score function: Term that kind of attract the samples towards the data density you see some kind of a gradient of log of the data density. p is unknown**

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.





Forward and reverse SDE

- Evolution governed by a *stochastic differential equation* (SDE)
- We'll generalize these later!

- **You do not need to know the P if you have an optimal Denoiser for this data set so you can directly evaluate that formula**
- **This is an opportunity we train a neural network to be such a denoiser. This means that we can run this kind of backboard equation Evolution using that learn D.**

Youtube Presentaiton

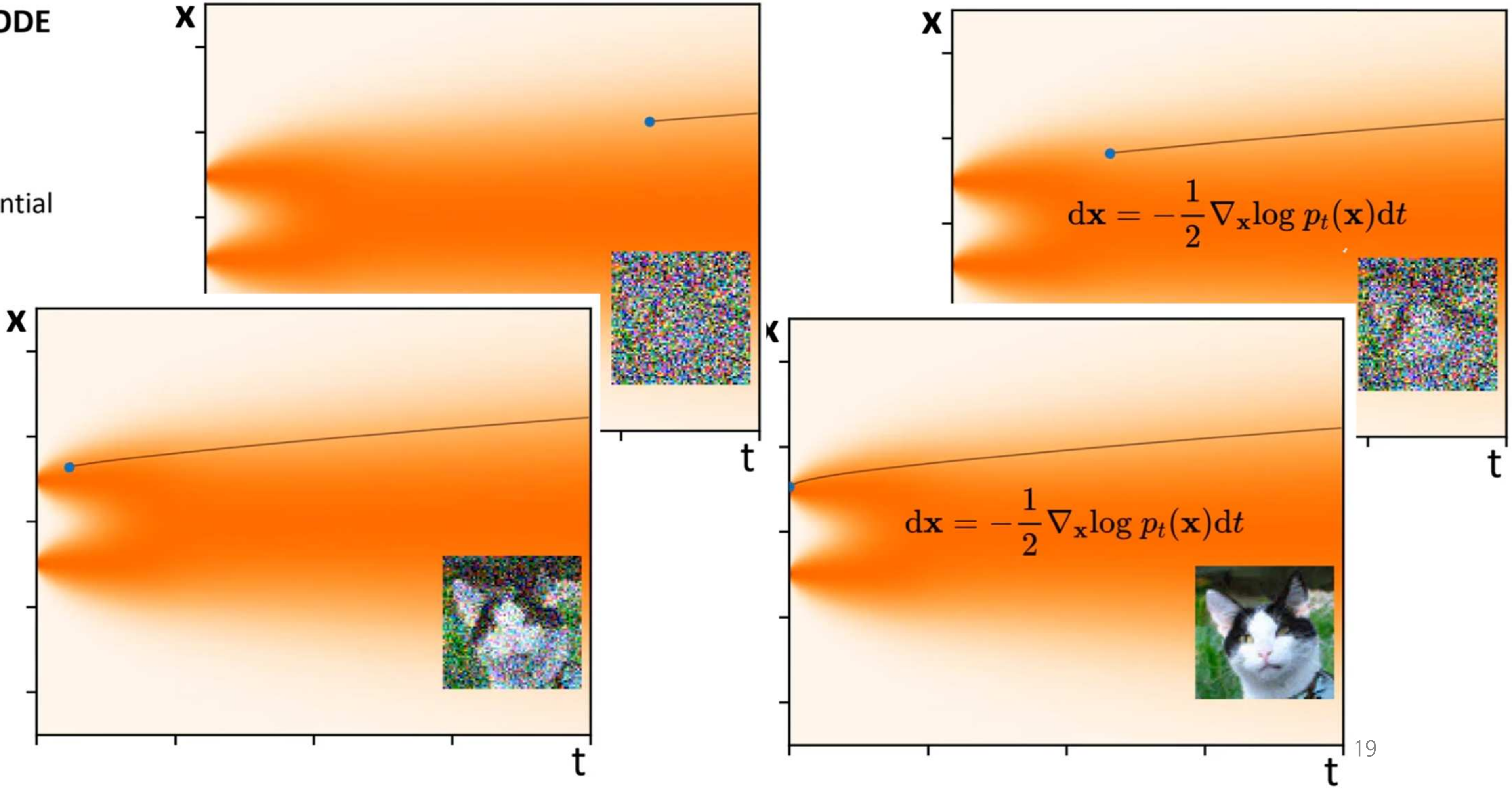
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Deterministic ODE

- Song et al. also present a *deterministic* ordinary differential equation

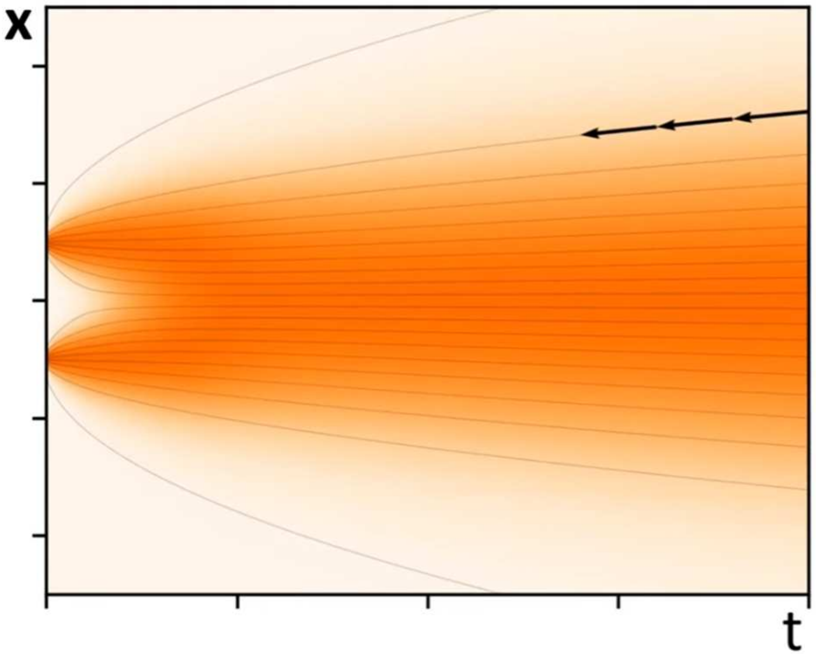
Don't have the stochastic term.

Have the core term scaled in a way.



Youtube Presentaiton

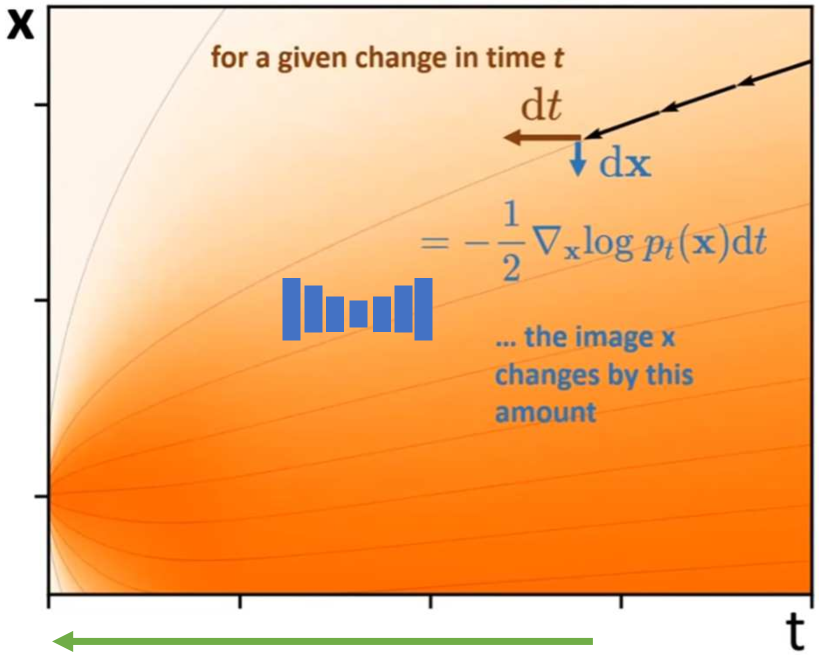
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Solution by discretization

Try to somehow follow flow lines to do the generation and in the way that happens is by discretization.

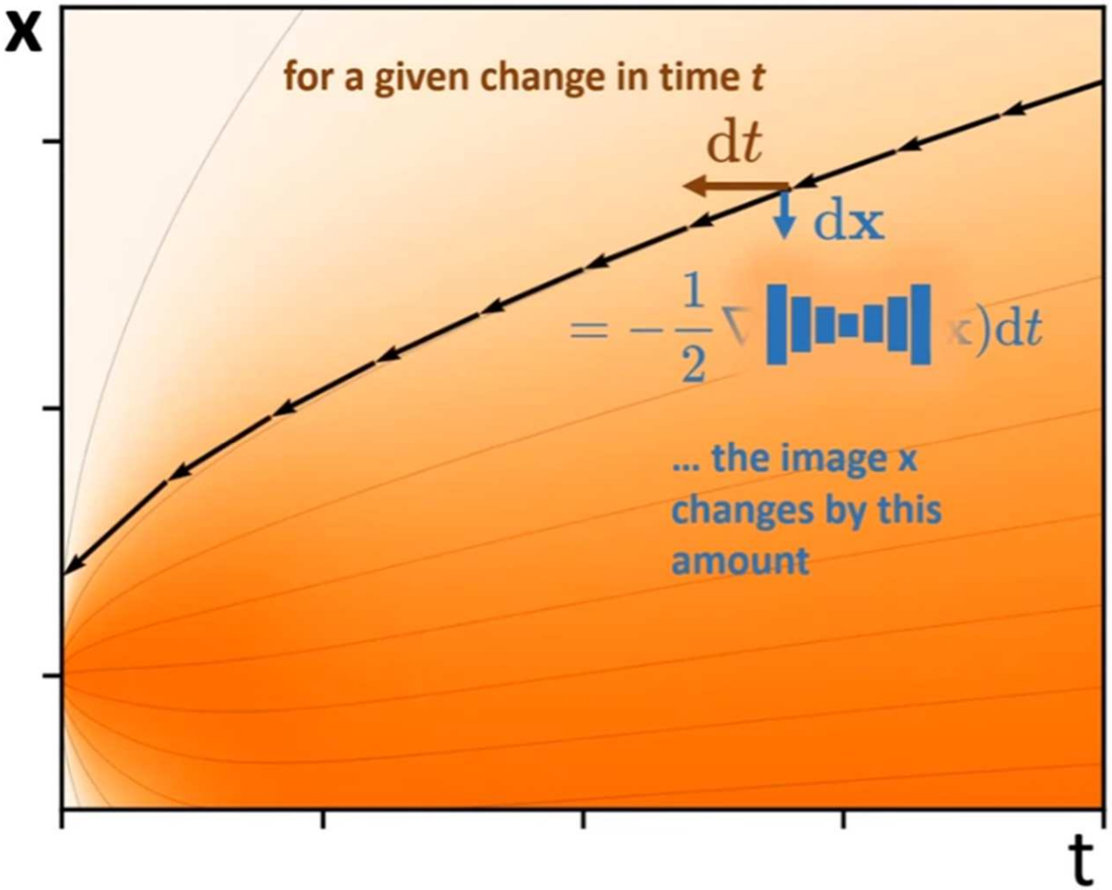
Time 축 스케일 확대



- **For any change in time I want to jump, the ODE formula tells me how much the image changes and again the ODE formula is evaluated this neural network, so the Network tells us where to go on the next step that's the general idea.**

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

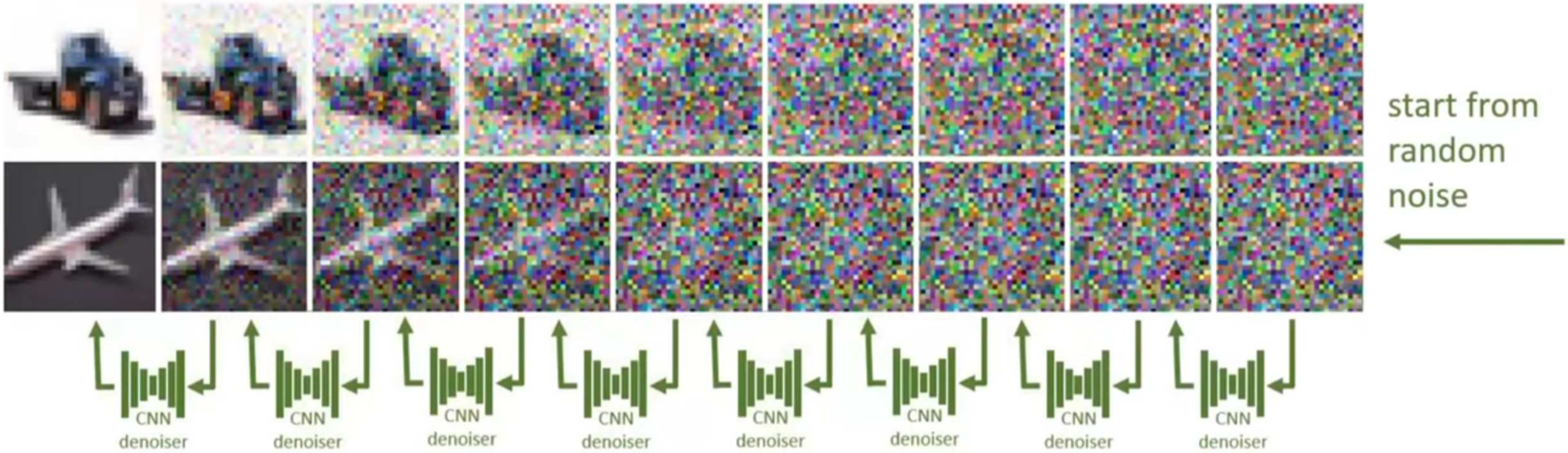


Solution by discretization

- In stochastic sampling (SDE), we would also inject noise at every step.
- We'll leave that for later and focus on the ODE first.

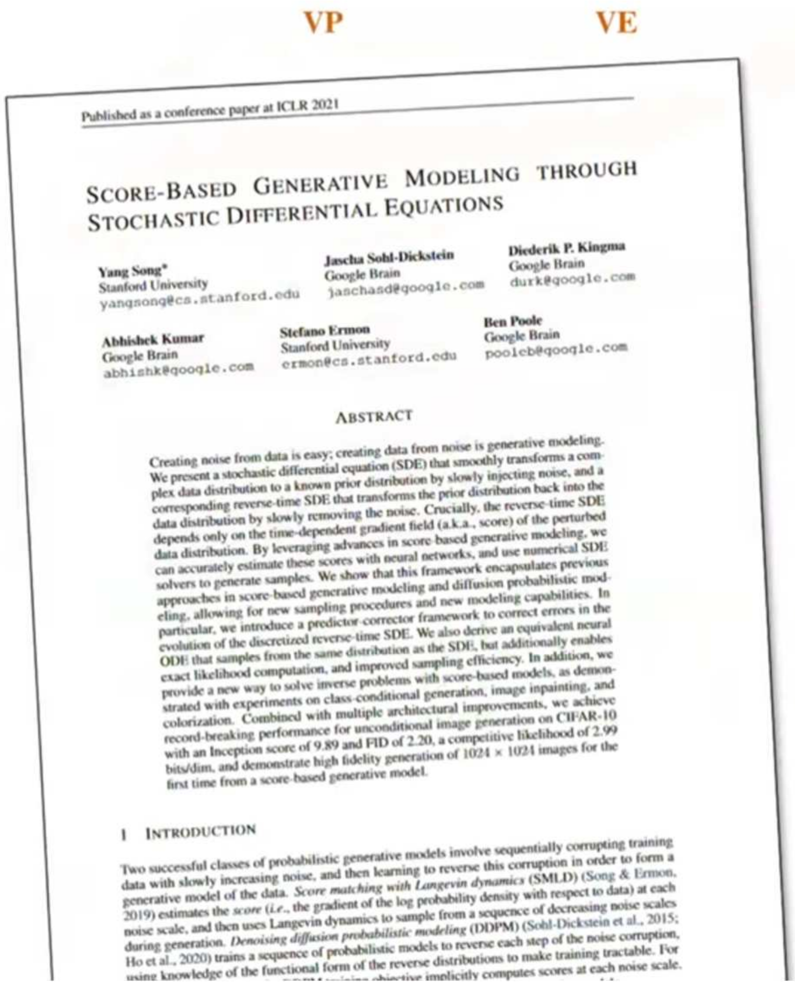
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Published as a conference paper at ICLR 2021

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song* Stanford University yangsong@cs.stanford.edu
Jascha Sohl-Dickstein Google Brain jaschasd@google.com
Diederik P. Kingma Google Brain durk@google.com
Abhishek Kumar Google Brain abhishk@google.com
Stefano Ermon Stanford University ermon@cs.stanford.edu
Ben Poole Google Brain pooleb@google.com

ABSTRACT

Creating noise from data is easy; creating data from noise is generative modeling. We present a stochastic differential equation (SDE) that smoothly transforms a complex data distribution to a known prior distribution by slowly injecting noise, and a corresponding reverse-time SDE that transforms the prior distribution back into the data distribution by slowly removing the noise. Crucially, the reverse-time SDE depends only on the time-dependent gradient field (a.k.a., score) of the perturbed data distribution. By leveraging advances in score-based generative modeling, we can accurately estimate these scores with neural networks, and use numerical SDE solvers to generate samples. We show that this framework encapsulates previous approaches in score-based generative modeling and diffusion probabilistic modeling, allowing for new sampling procedures and new modeling capabilities. In particular, we introduce a predictor-corrector framework to correct errors in the evolution of the discretized reverse-time SDE. We also derive an equivalent neural ODE that samples from the same distribution as the SDE, but additionally enables exact likelihood computation, and improved sampling efficiency. In addition, we provide a new way to solve inverse problems with score-based models, as demonstrated with experiments on class-conditional generation, image inpainting, and colorization. Combined with multiple architectural improvements, we achieve record-breaking performance for unconditional image generation on CIFAR-10 with an Inception score of 9.89 and FID of 2.20, a competitive likelihood of 2.99 bits/dim, and demonstrate high fidelity generation of 1024 × 1024 images for the first time from a score-based generative model.

1 INTRODUCTION

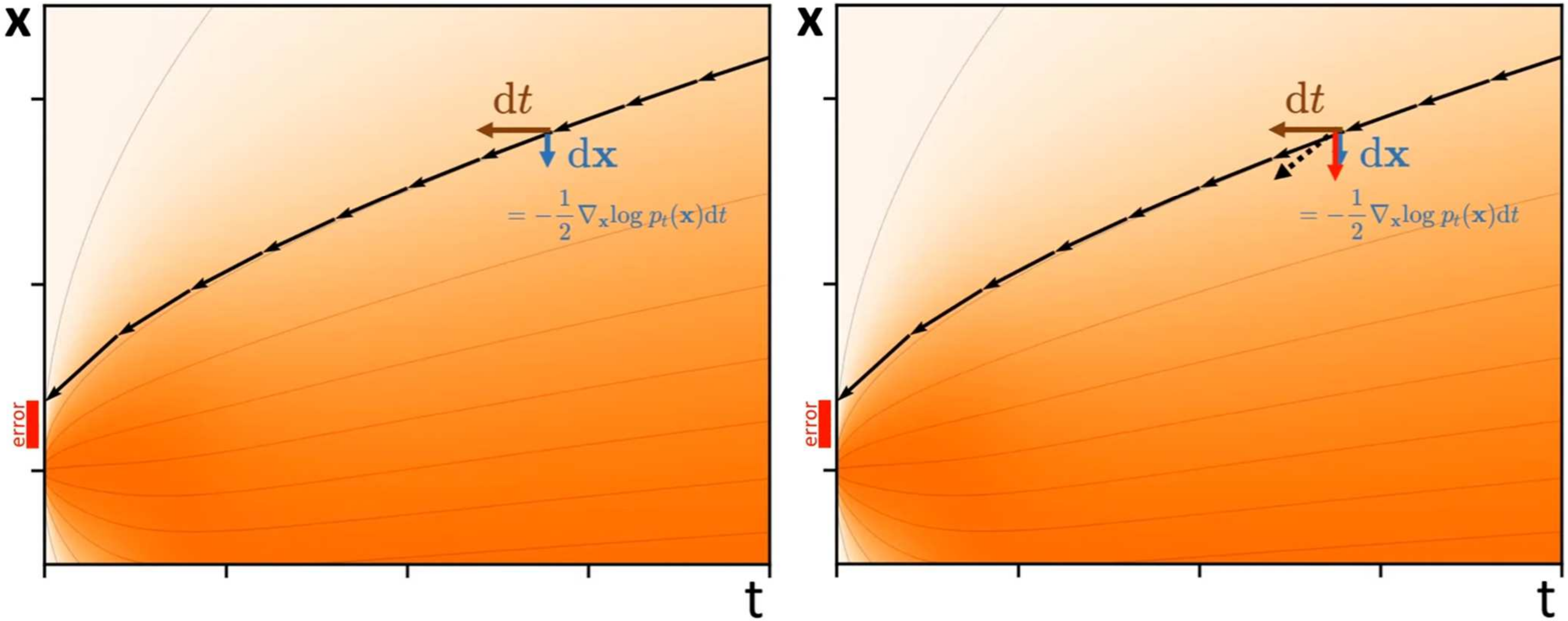
Two successful classes of probabilistic generative models involve sequentially corrupting training data with slowly increasing noise, and then learning to reverse this corruption in order to form a generative model of the data. *Score matching with Langevin dynamics* (SMLD) (Song & Ermon, 2019) estimates the *score* (*i.e.*, the gradient of the log probability density with respect to data) at each noise scale, and then uses Langevin dynamics to sample from a sequence of decreasing noise scales during generation. *Denoising diffusion probabilistic modeling* (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) trains a sequence of probabilistic models to reverse each step of the noise corruption, using knowledge of the functional form of the reverse distributions to make training tractable. For

That was
Song et al. (2021)
in a nutshell (for our purposes)

Next, let's identify some **design choices** from different methods, and generalize.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



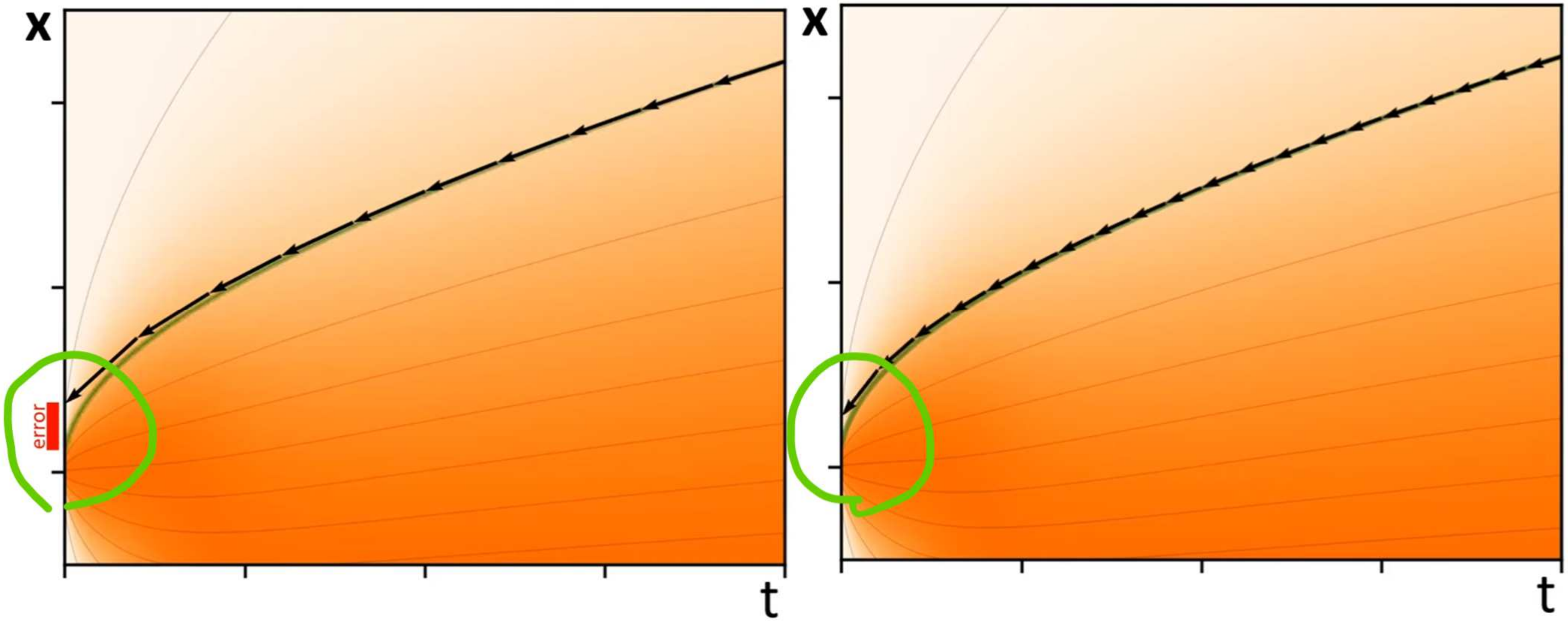
Error sources

1. Network fails to approximate the true score of data

- **When I do this sampling chain, the obvious one is that if the network gives me an incorrect direction and I end up moving in the incorrect direction and in the end I end up somewhat in the wrong place.**

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.



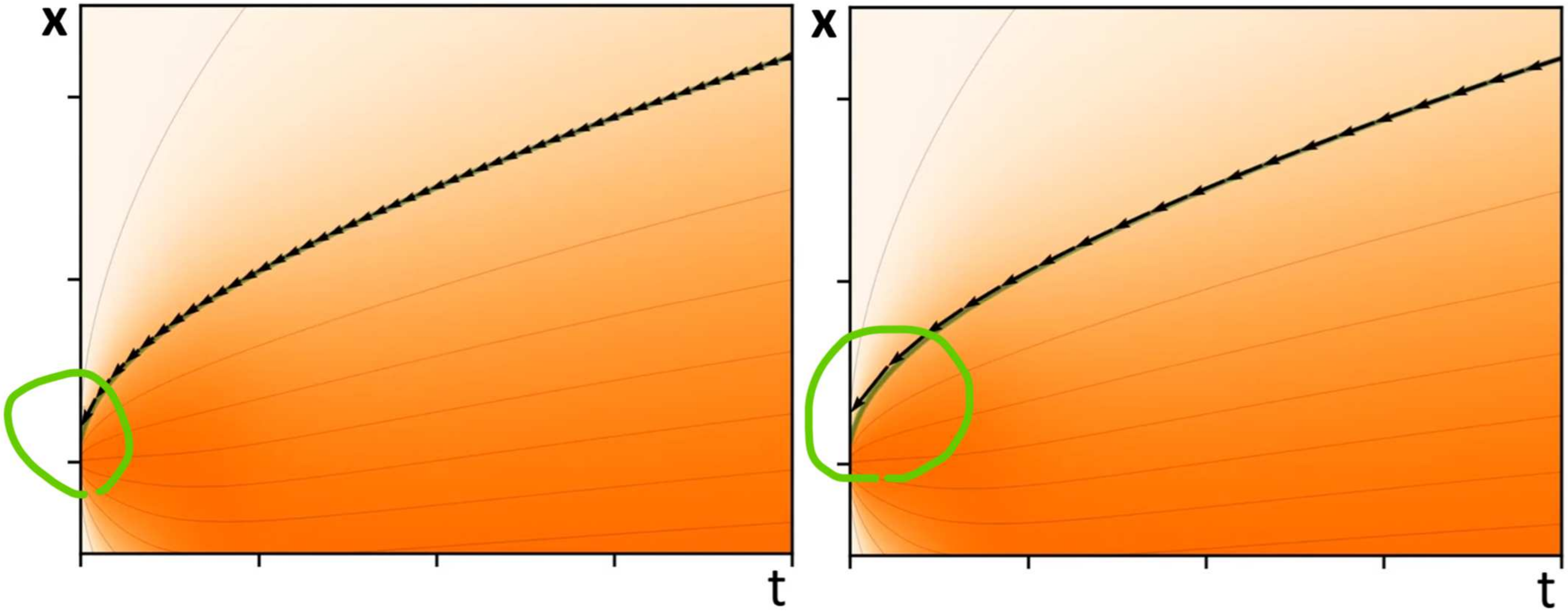
Error sources

1. Network fails to approximate the true score of data
2. Approximating ideal trajectory by finite steps

- **Try to approximate this continuous trajectory in green here using these linear segments.**
- **If try to jump too far at once, the curve will kind of move away from our feet.**

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models]," NeurIPS 2022.



Error sources

1. Network fails to approximate the true score of data
2. Approximating ideal trajectory by finite steps

- **Brutal solution is to take more steps but more compute to generate an image.**

Youtube Presentaiton

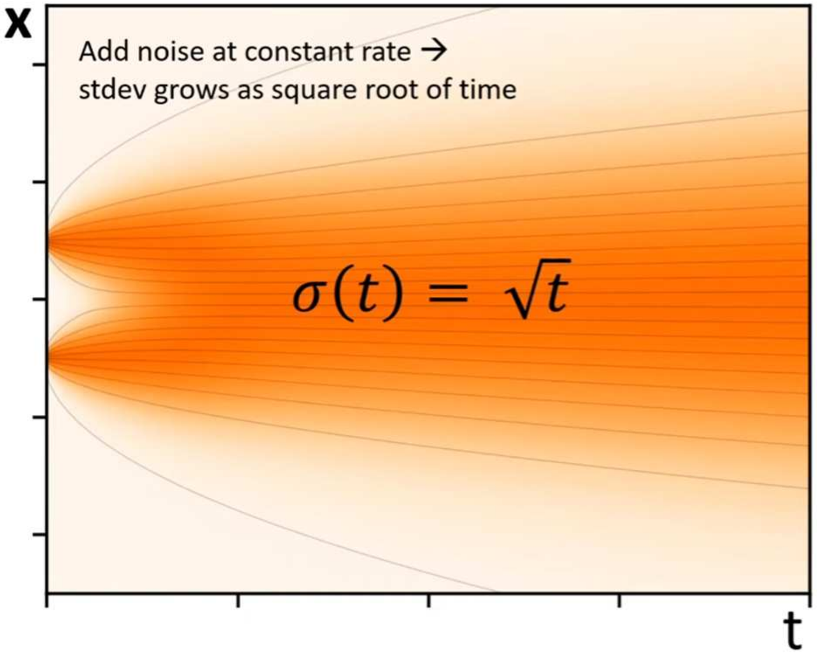
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

- You don't have to sample in a certain way just because you train your neural network in a certain way and so on you can decouple this.

➢ **We'll be looking at sampling first and then coming back to the training later.**

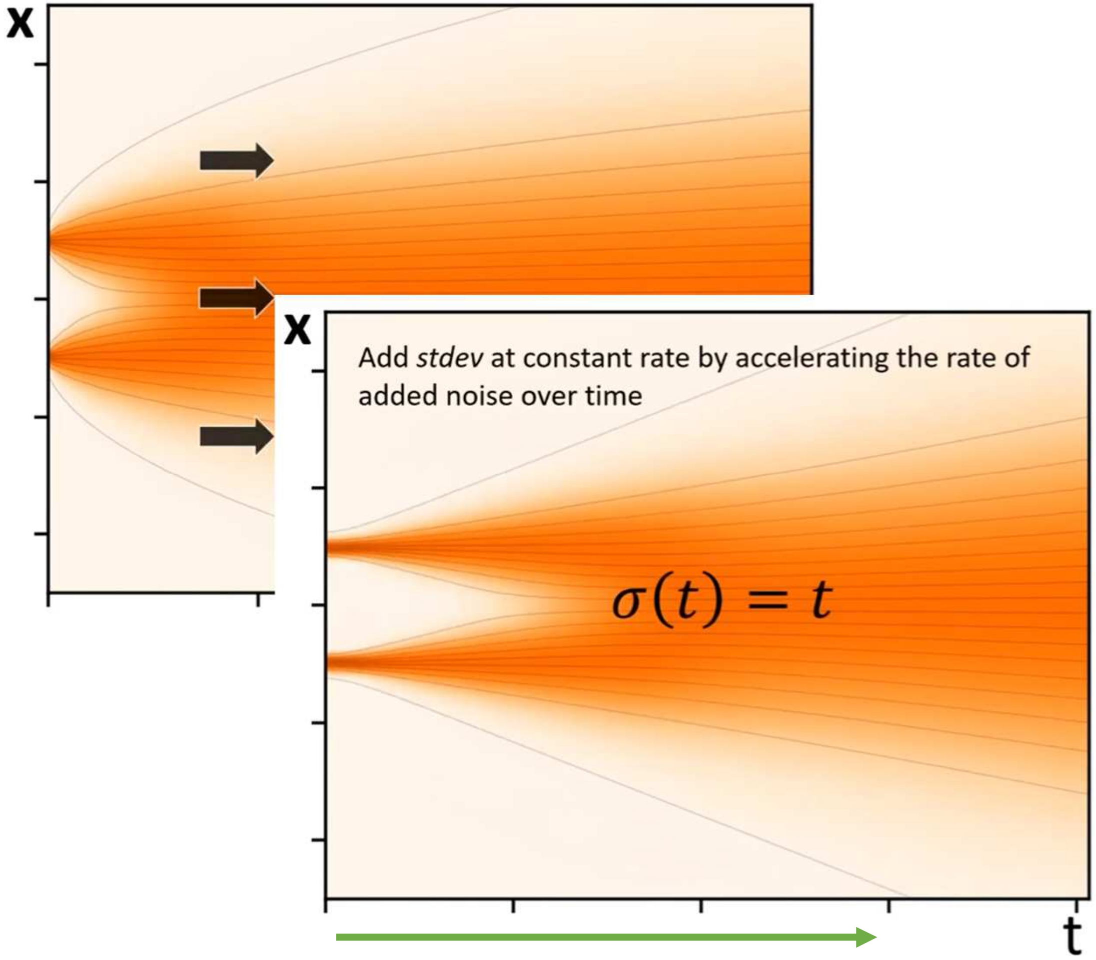
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Noise schedule

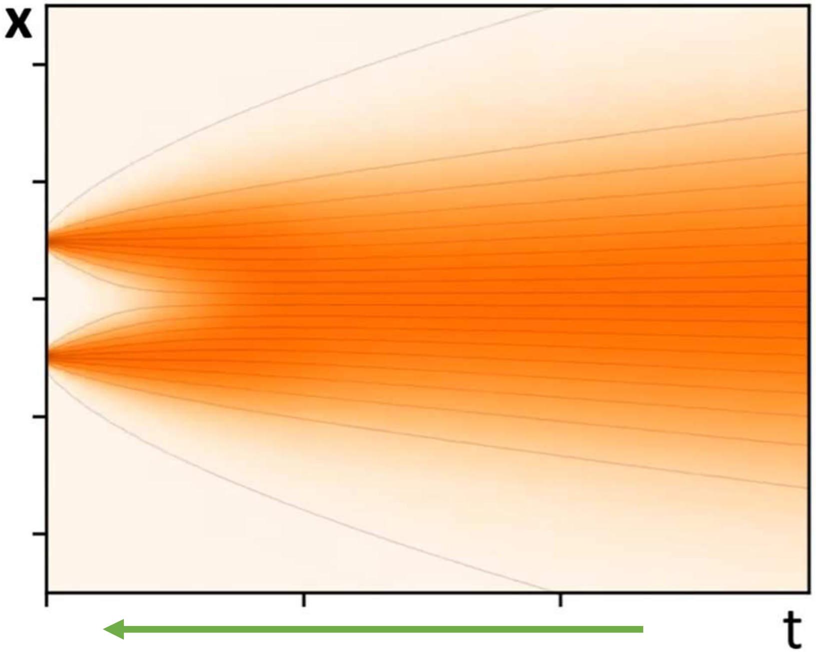
- Add noise at different rate → different ODE
- Many schedules in literature



- **Noise schedule where the noise level increase as a square root of time because that's how the variance grows linearly, so the standard deviation grows square root.**

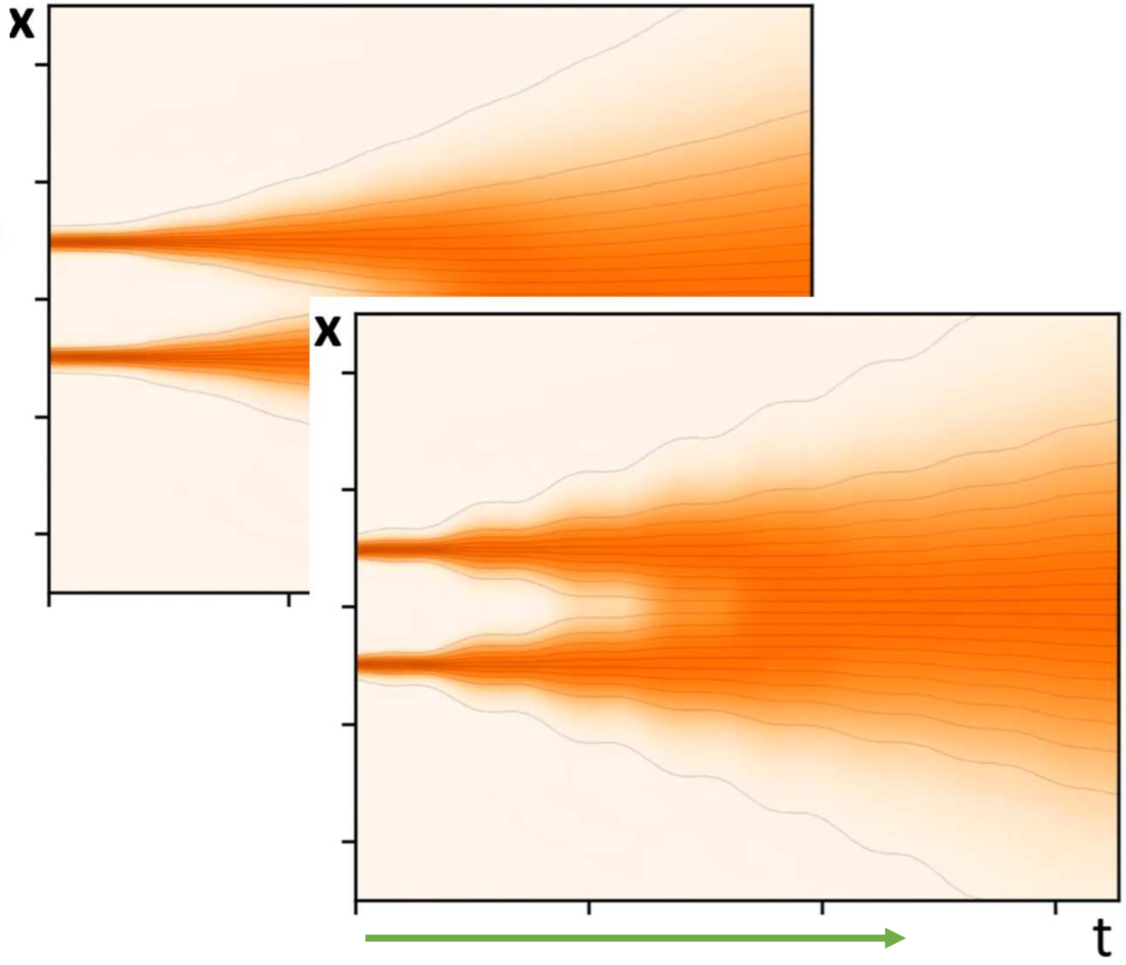
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



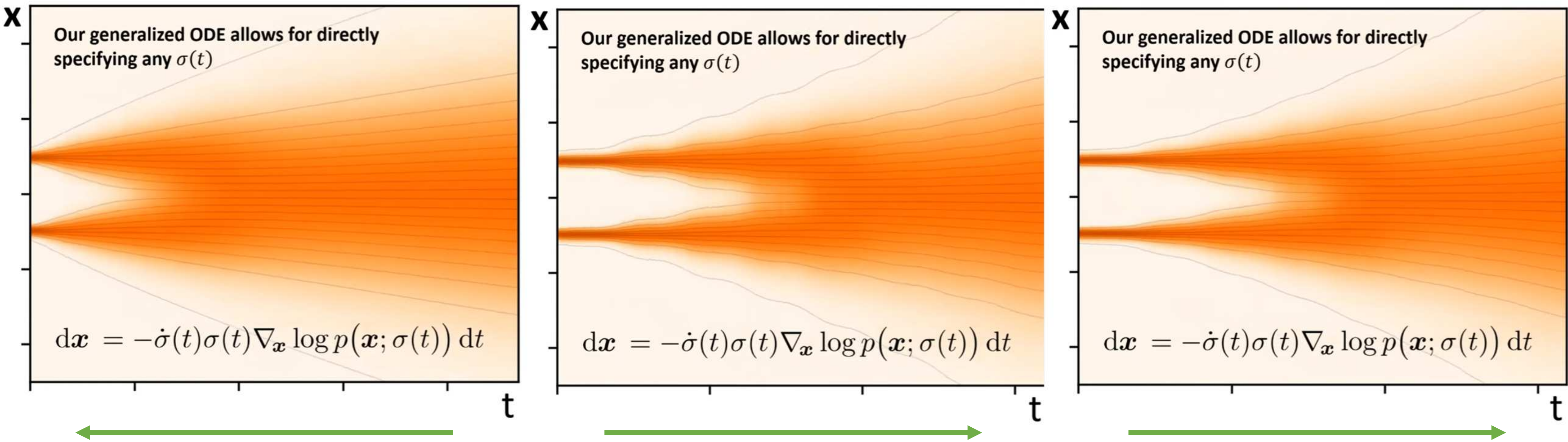
Noise schedule

- Add noise at different rate → different ODE
- Many schedules in literature
- All of them simply warp the *t*-axis



Youtube Presentaiton

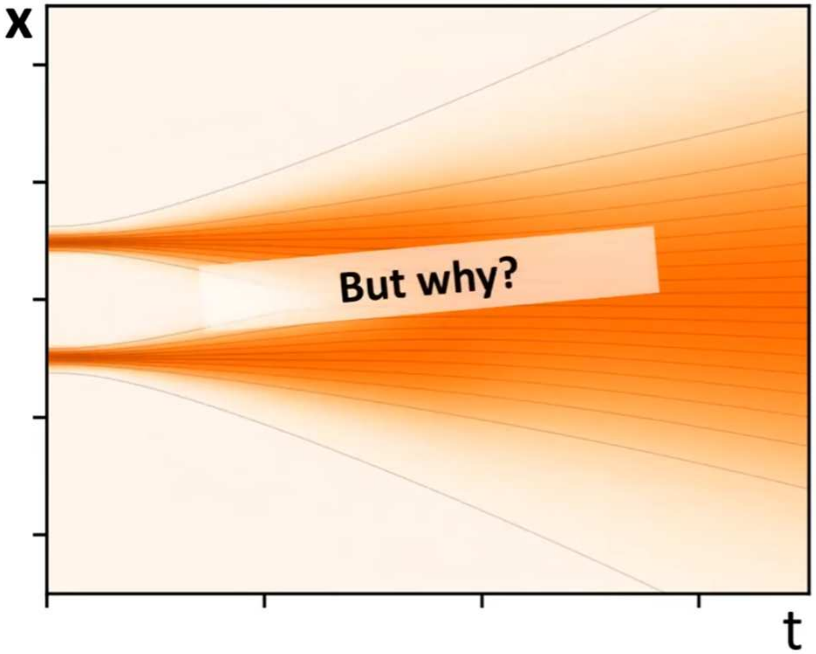
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



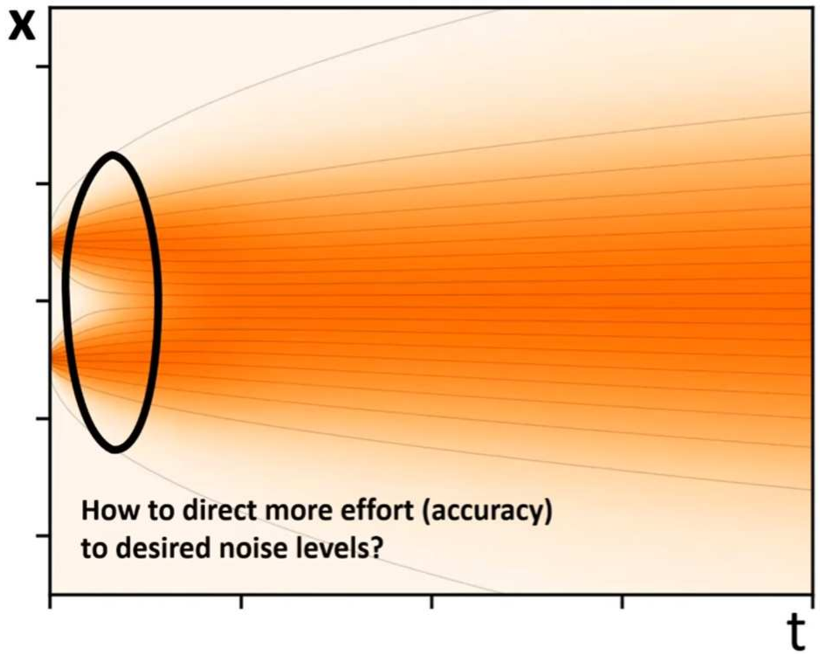
- **Generalize ODE**
- **We can parameterize the noise level we want to have reached by explicitly by this Sigma function.**

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.



- Almost nothing happens until at almost zero time, noise level suddenly curves rapidly to one of these two basins and there's high curvature.
- Careful in sampling that region and less careful here in the bulk

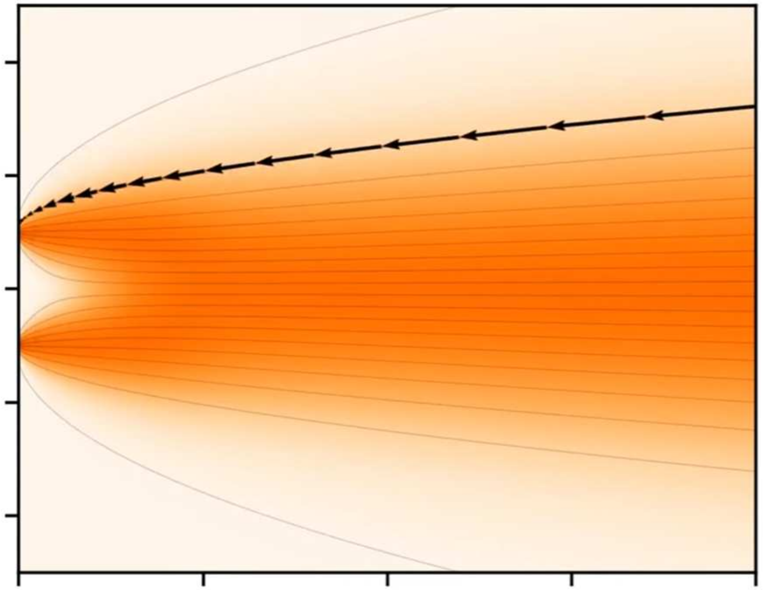


- There's two ideas of how you might do that.

Noise schedule
vs **step lengths**

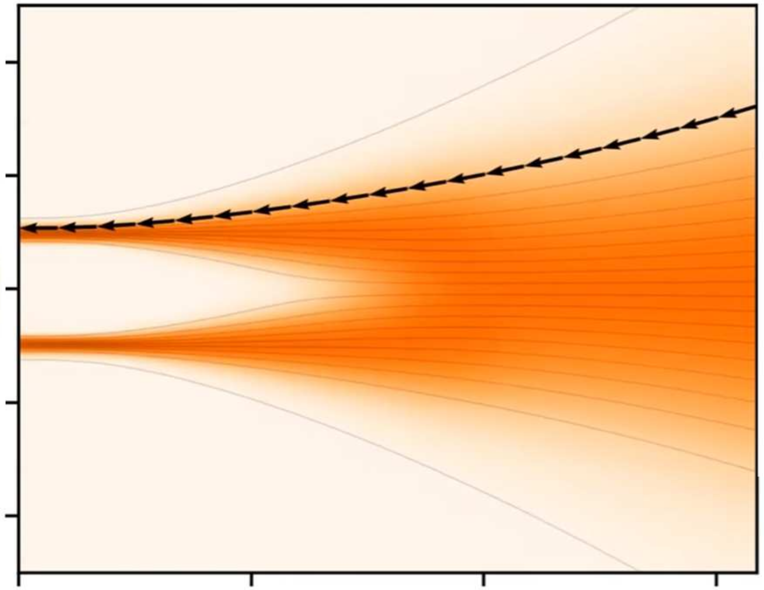
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.



Noise schedule vs step lengths

- Two options:
  - Take shorter steps at desired levels
  - Warp noise schedule to spend more time at those levels



Noise schedule vs step lengths

- Two options:
  - Take shorter steps at desired levels
  - Warp noise schedule to spend more time at those levels

- **Take shorter steps at the more difficult parts usually it's the low noise levels**

These two approaches are not equivalent.

The error characteristics can be vastly different between these choices like the eror that comes from tracking this continuous curve.

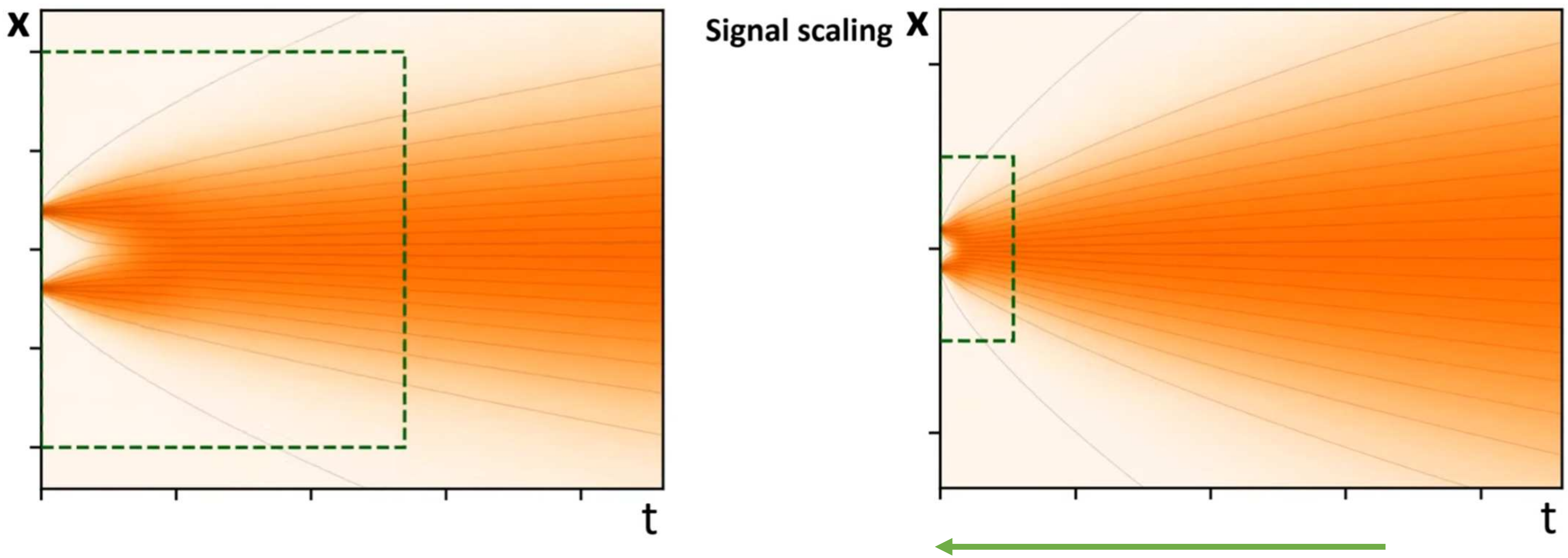
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

		VP	VE	iDDPM	+ DDIM	Ours
Sampling						
Time steps	$t_{i<N}$					
Sampling time						
Schedule	$\sigma(t)$					
Noise schedule						

Youtube Presentaiton

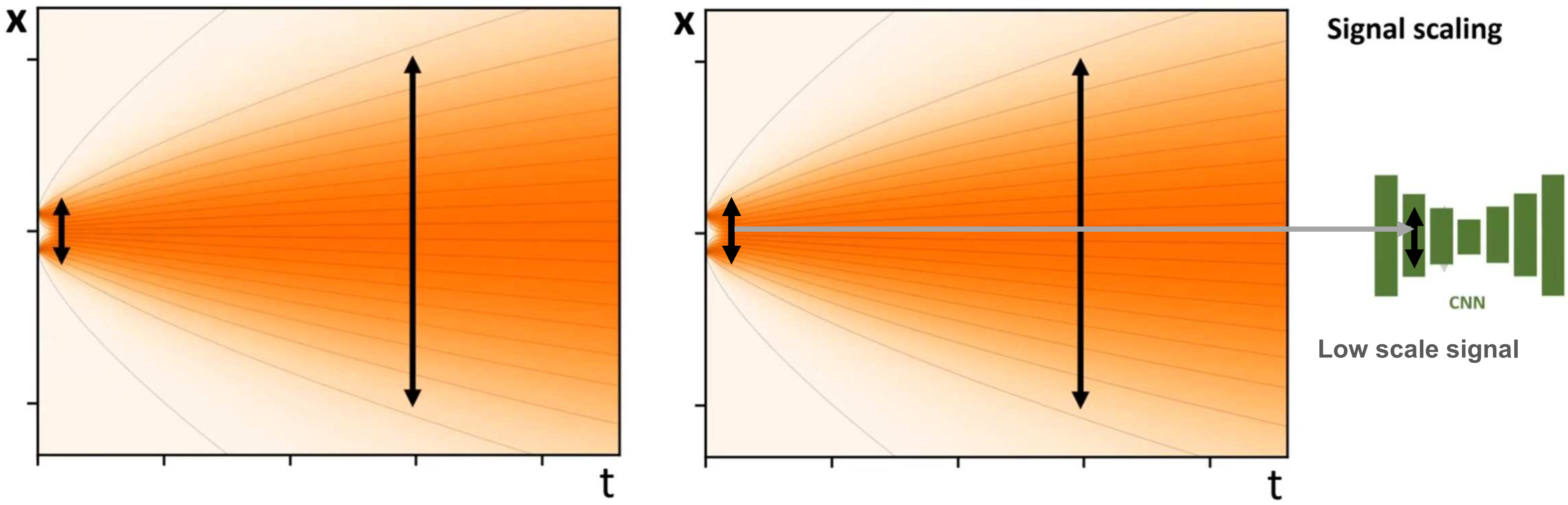
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



- **Zoom out a little because in reality we add a ton of noise, the noise level is very large at the other extreme.**

Youtube Presentaiton

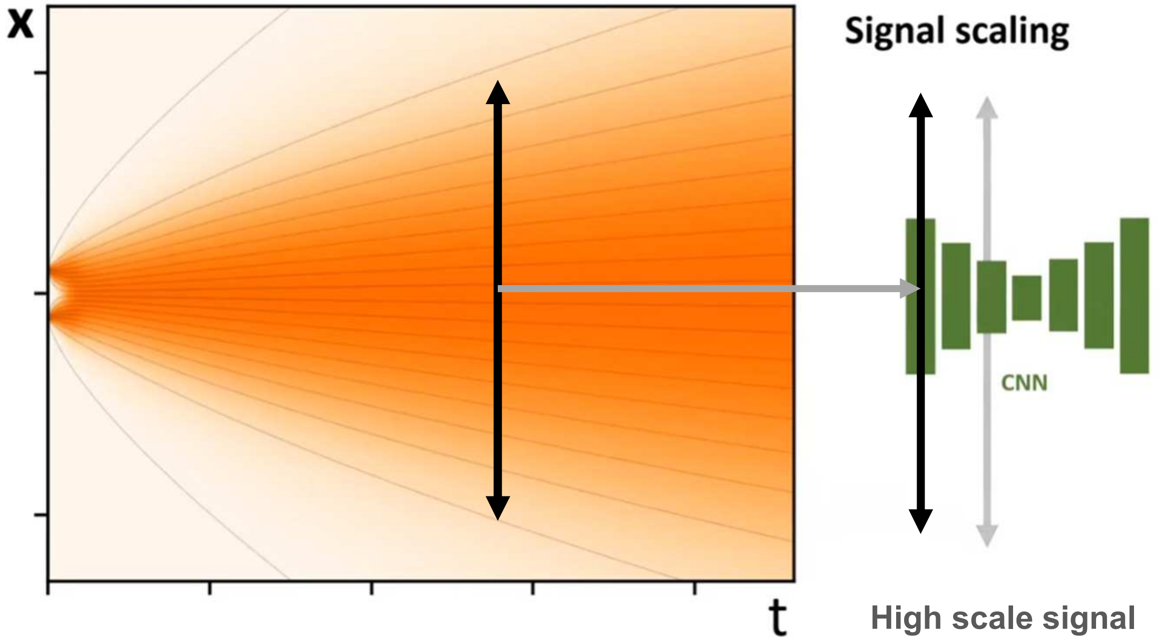
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.



- **If don't do anything, the signal magnitude grows as the noise level grows - Keep piling noise**
- **The signal is quite simply bigger numerically**

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



High scale signal

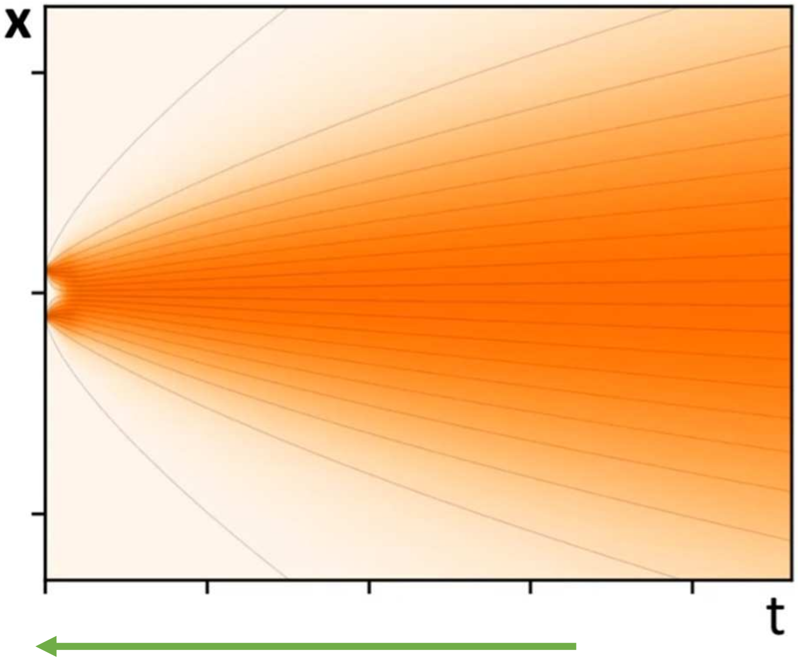
Signals are much larger at the high noise levels than in the low noise levels.

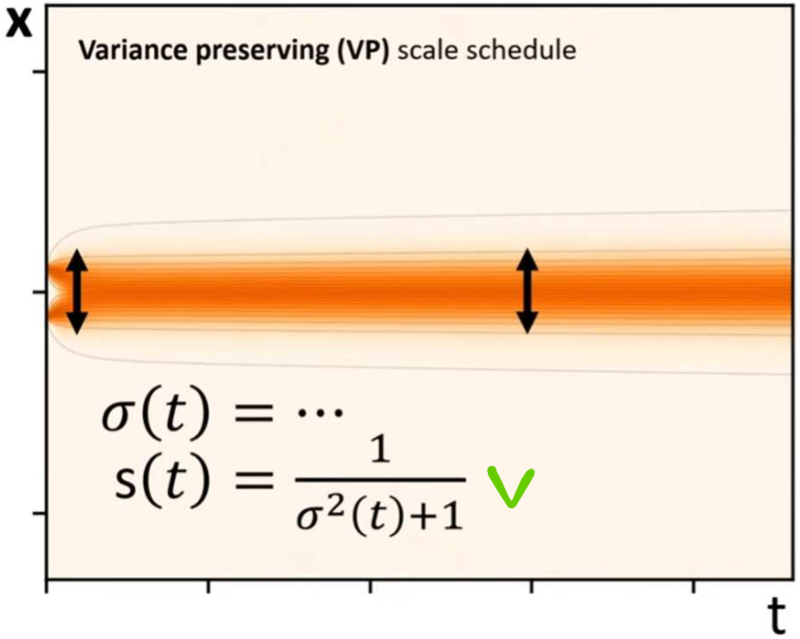
→ To be really bad for neural network training dynamics.

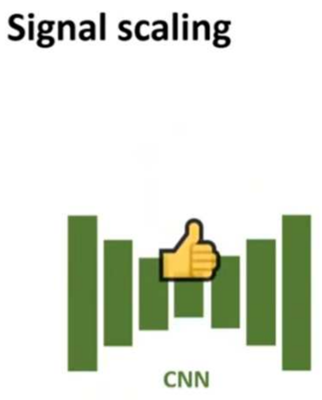
→ Actually critical to deal with to get good performance.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.







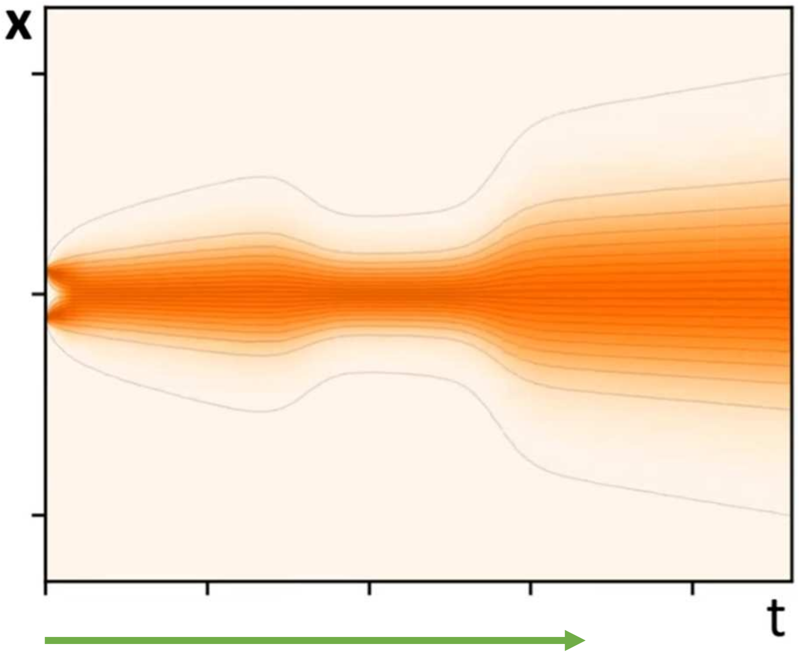
Variance Preserving (VP) Scale Schedule

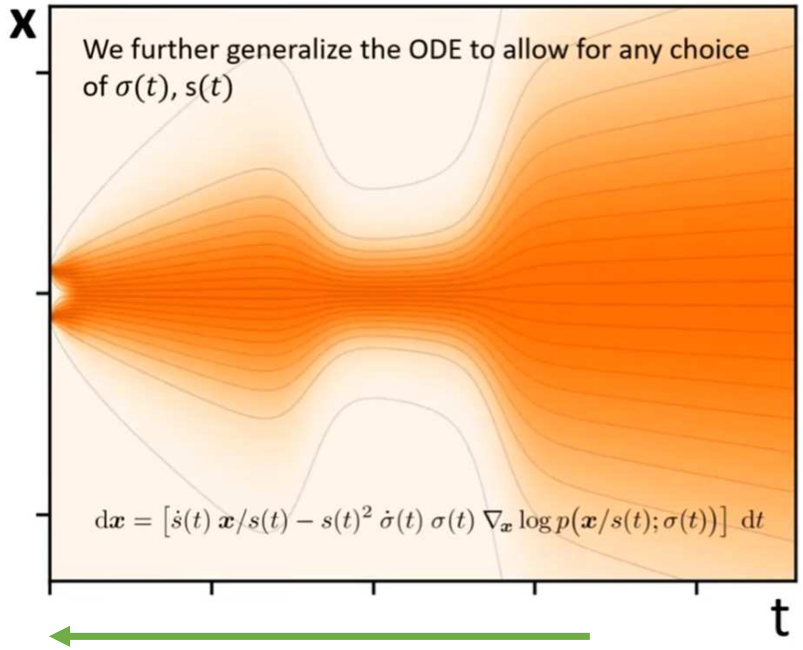
- **Scale schedule : Squeeze the signal magnitude into the constant variance tube so that makes the network happy**

Youtube Presentaiton

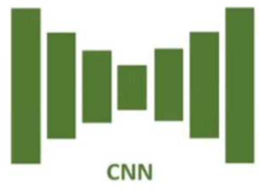
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Scaled ODE





Signal scaling

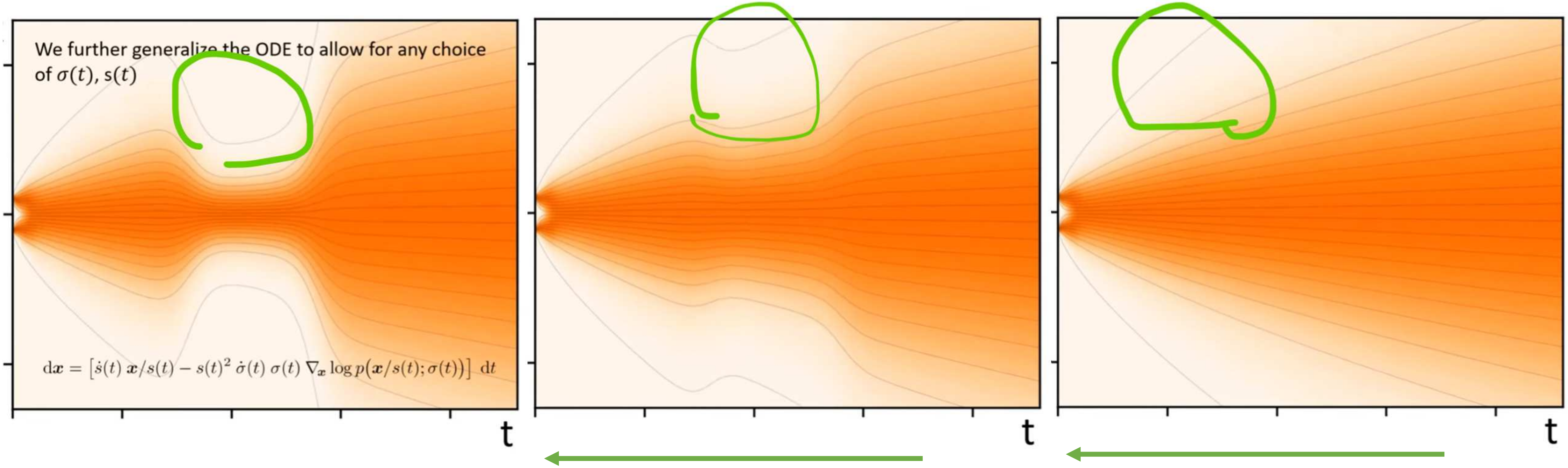


- **Formulating an ODE that allows you to directly specify any arbitrary scale schedule.**
- **The only thing that the scale schedule does is distort these flow lines in some way.**

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

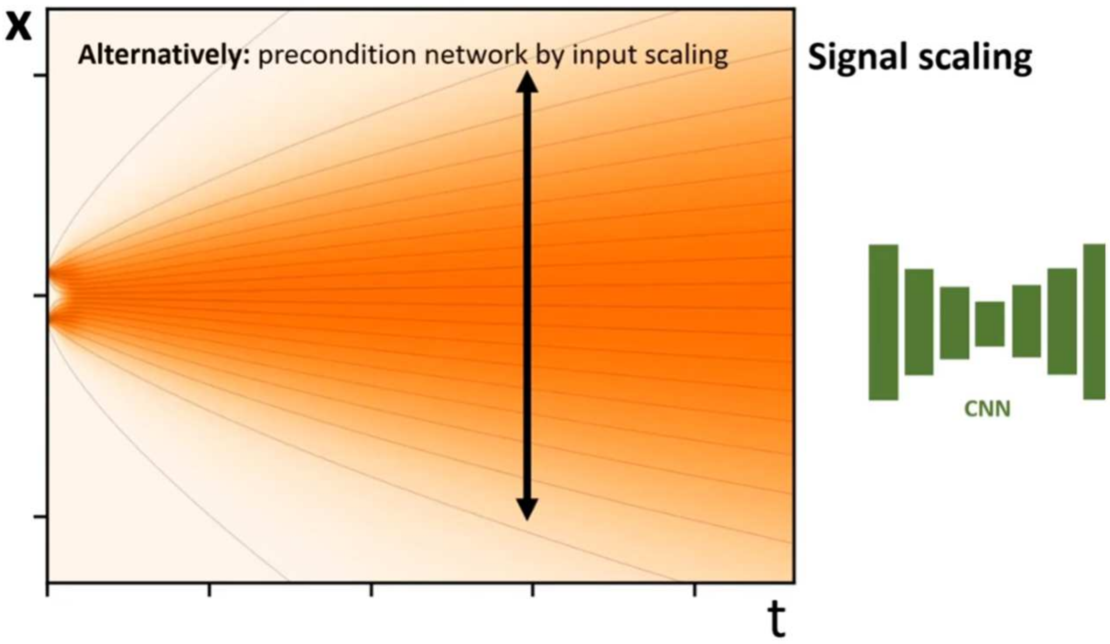
Scaled ODE

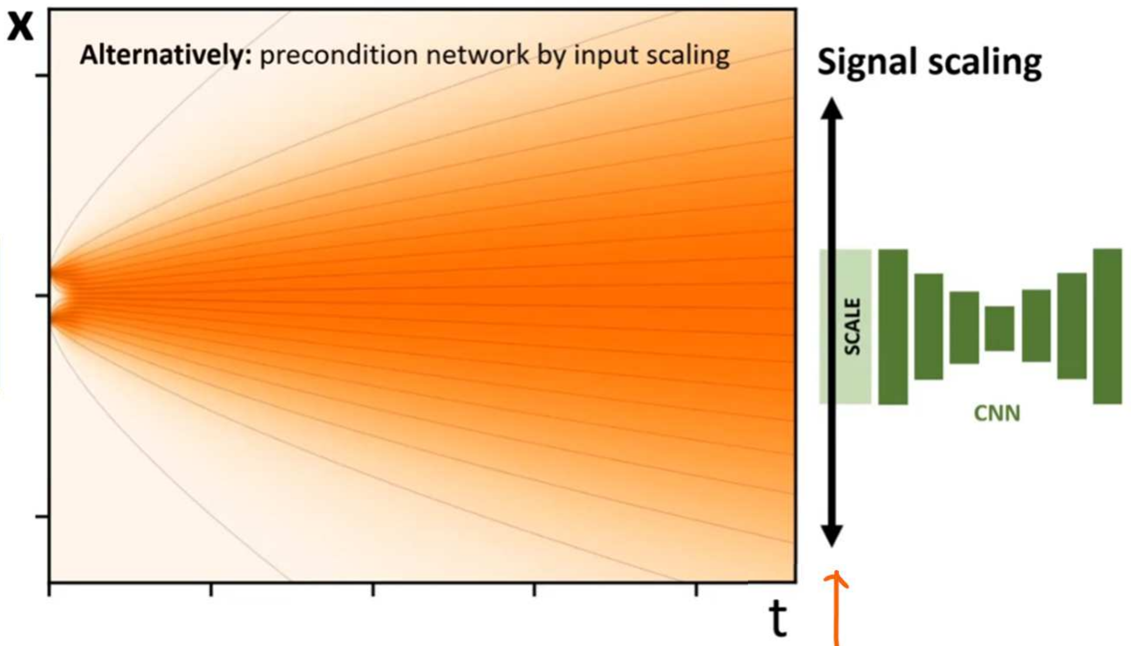


Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Alternatively : Initial Scaling layer



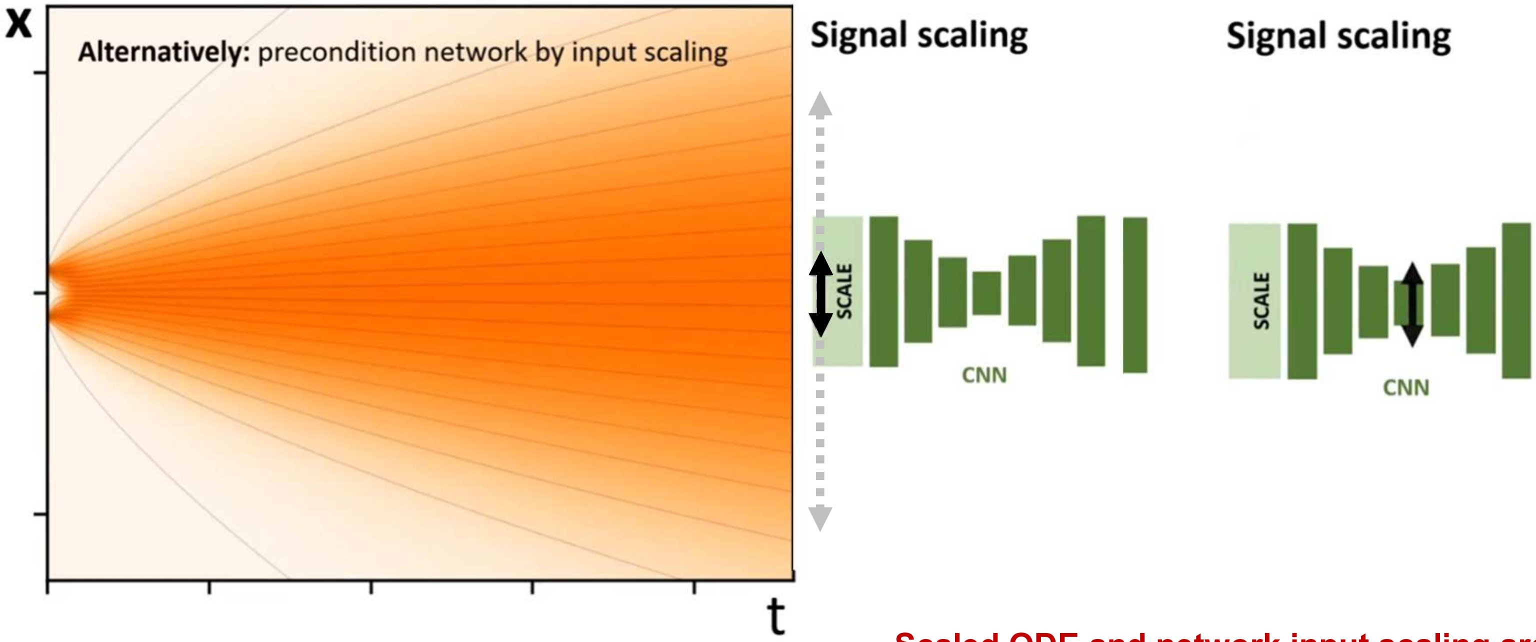


Quite simple Instead of changing ODE

Initial scaling layer tha uses the known signal scale

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Scaled ODE and network input scaling are not equivalent

Because the error characteristics are vastly different between these two cases.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.

Noise schedule

Scaling schedule

		VP	VE	iDDPM + DDIM	Ours
Sampling					?
Time steps	$t_{i<N}$	$1+\frac{i}{N-1}(\epsilon_s-1)$	$\sigma_{\max}^2\left(\sigma_{\min}^2/\sigma_{\max}^2\right)^{\frac{i}{N-1}}$	$u_{\lfloor j_0+\frac{M-1-j_0}{N-1}i+\frac{1}{2}\rfloor}$, where $u_M=0$, $u_{j-1}=\sqrt{\frac{u_j^2+1}{\max(\bar{\alpha}_{j-1}/\bar{\alpha}_j, C_1)}-1}$	
Schedule	$\sigma(t)$	$\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}-1}$	$\sqrt{t}$	t	
Scaling	$s(t)$	$1/\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}}$	1	1	
Network and preconditioning		Skip scaling $c_{\text{skip}}(\sigma)$ Output scaling $c_{\text{out}}(\sigma)$ Input scaling $c_{\text{in}}(\sigma)$ Noise cond. $c_{\text{noise}}(\sigma)$			
Input scaling	$c_{\text{in}}(\sigma)$	$1/\sqrt{\sigma^2+1}$	1	$1/\sqrt{\sigma^2+1}$	

Identified the design choices the scaling & schedule and the scaling that happens inside the neural network itself that we count as a so-called preconditioning of the neural network.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

			VP	VE	iDDPM + DDIM	Ours
	Sampling					?
	Time steps	$t_{i<N}$	$1+\frac{i}{N-1}(\epsilon_s-1)$	$\sigma_{\max}^2\left(\sigma_{\min}^2/\sigma_{\max}^2\right)^{\frac{i}{N-1}}$	$u_{\lfloor j_0+\frac{M-1-j_0}{N-1}i+\frac{1}{2}\rfloor}$, where $u_M=0$ $u_{j-1}=\sqrt{\frac{u_j^2+1}{\max(\bar{\alpha}_{j-1}/\bar{\alpha}_j, C_1)}-1}$	
Noise schedule	Schedule	$\sigma(t)$	$\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}-1}$	$\sqrt{t}$	t	
Scaling schedule	Scaling	$s(t)$	$1/\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}}$	1	1	
	Network and preconditioning					
	Input scaling	$c_{\text{in}}(\sigma)$	$1/\sqrt{\sigma^2+1}$	1	$1/\sqrt{\sigma^2+1}$	

We'll use pre-trained networks from previous work for now...

and return to training in Section IV

We'll just try to improve the sampling;
Deterministic & Stochastic sampling.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

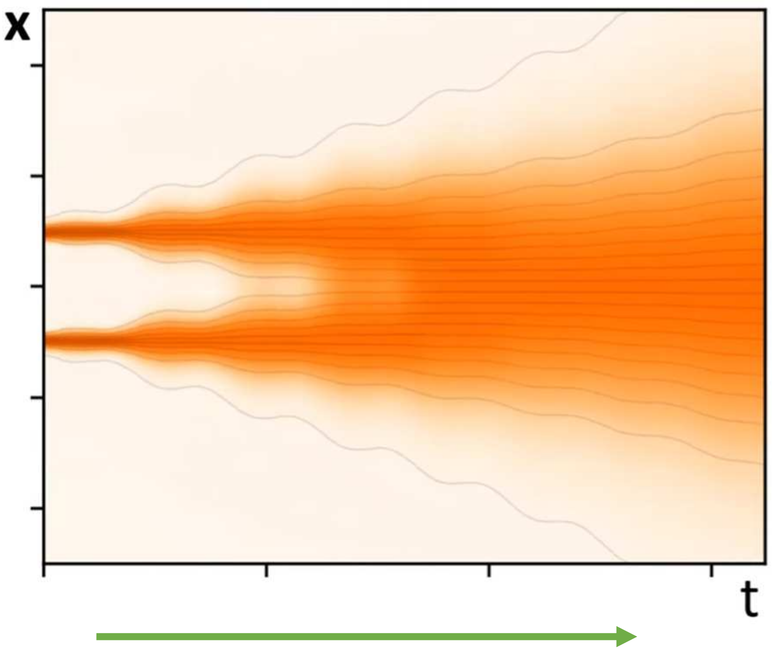
Outline

- Part I: **Common framework**
  - Identifying the moving parts in existing work
- Part II: **Deterministic sampling**
  - Solving the ODE efficiently
- Part III: **Stochastic sampling**
  - Why SDE's? How to do stochastic stepping?
- Part IV: **Preconditioning and training**
  - How to train the CNN used in evaluating a step?

Youtube Presentaiton

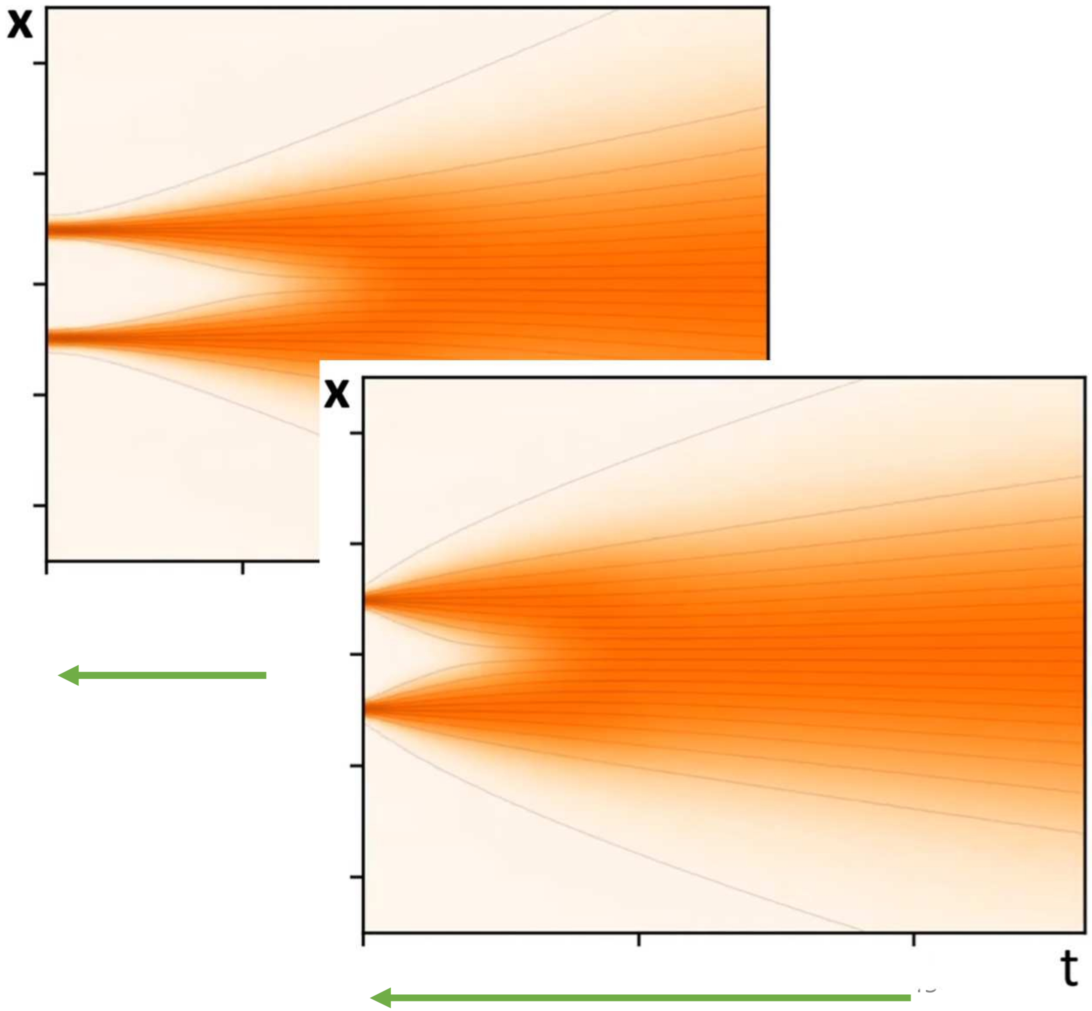
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Noise Schedule $\sigma(t)$



Noise schedule

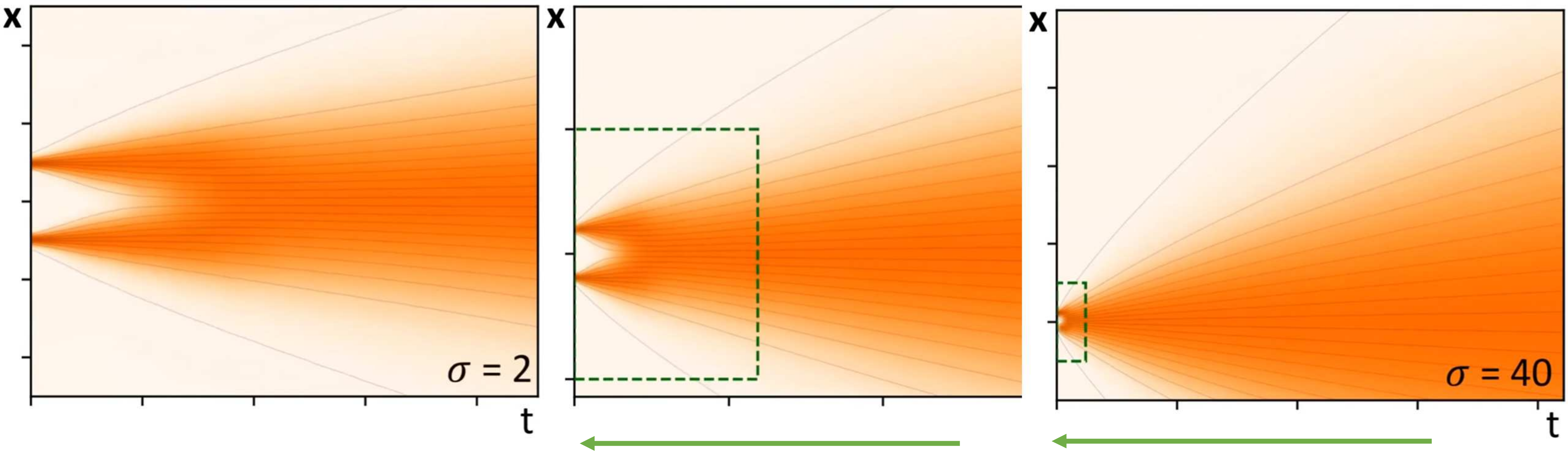
- Why are some better than others?



Youtube Presentaiton

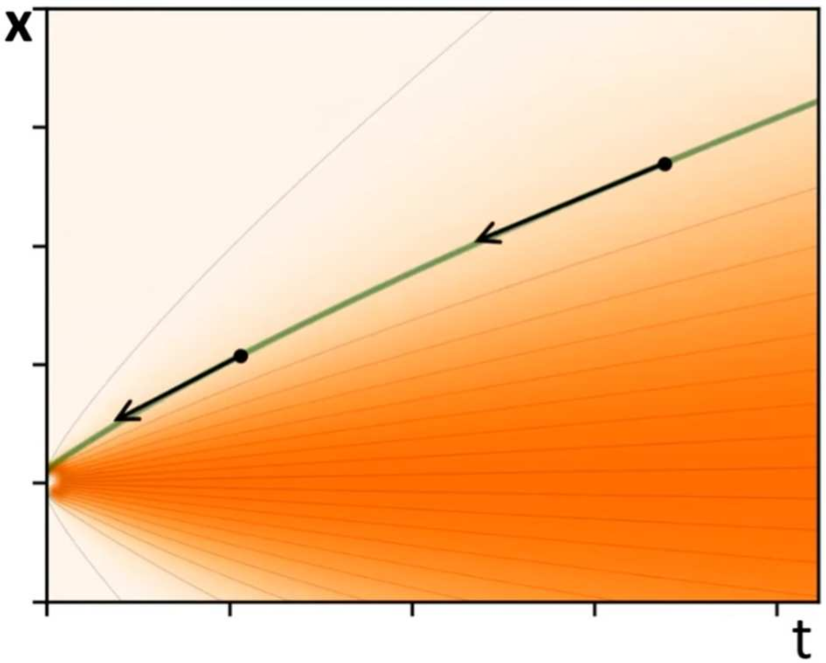
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.

Noise Schedule $\sigma(t)$

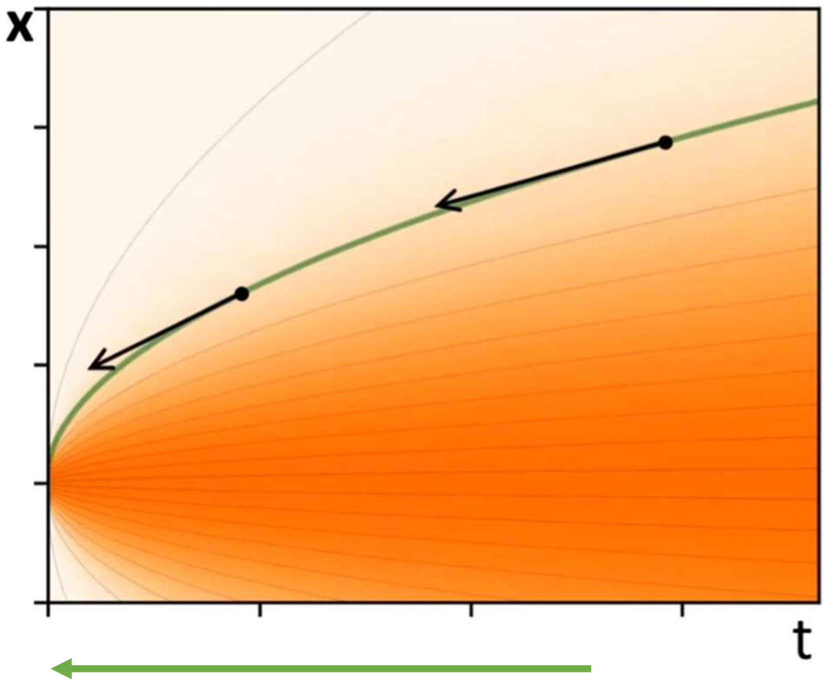


EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Noise Schedule $\sigma(t)$



More successful when the tangents happen to coincide with this curve trajectory and so the trajectory is as straight as possible

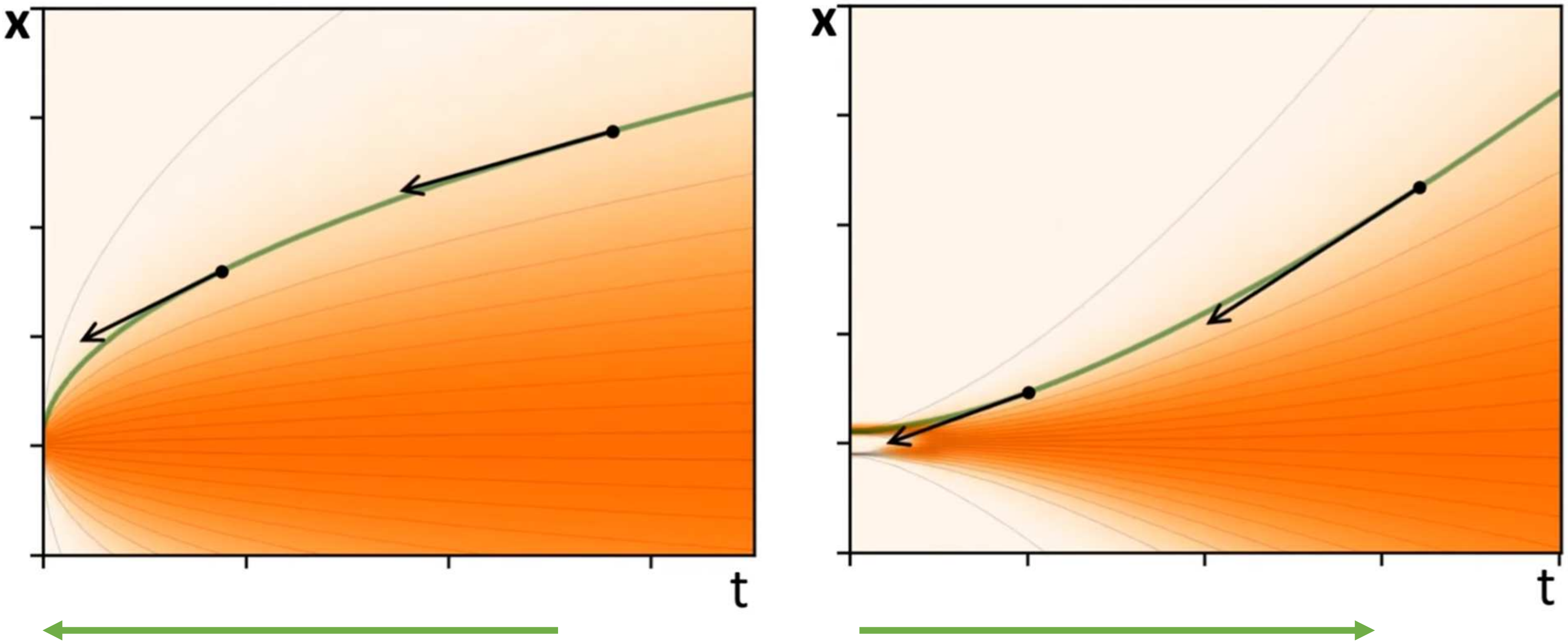


Bad schedule : a visible gap between the tangent and the curve

Easily fall off if you try to step too much.

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Noise Schedule $\sigma(t)$



Noise schedule

- Why are some better than others?
- Straighter trajectory

  → less need to "correct course" along the way

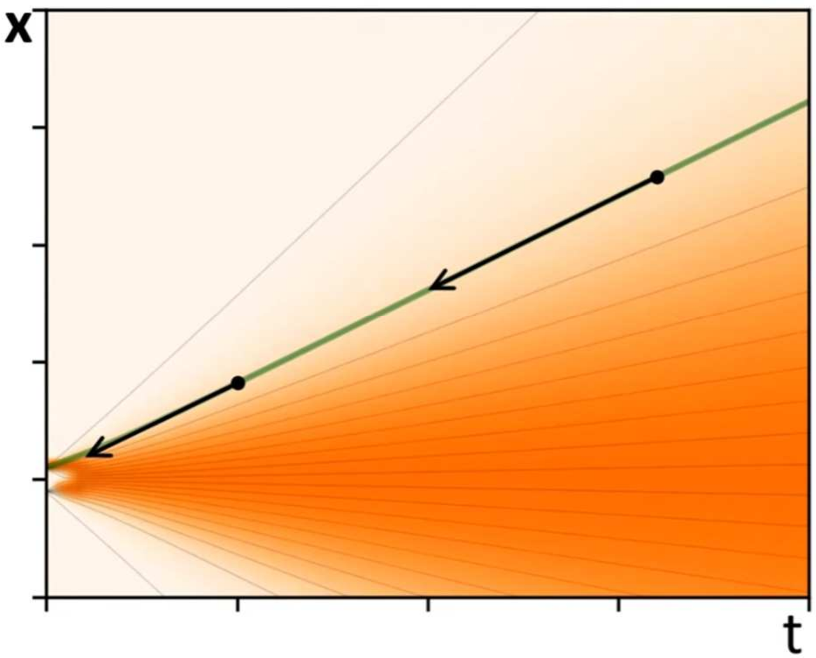
  → fewer steps suffice

Random family of different schedules

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Noise Schedule $\sigma(t)$ **Scling Schedule $s(t)$**



We advocate the "linear" schedule (same as DDIM):

$\sigma(t) = t$

$\boxed{s(t) = 1}$

Note: We'll normalize signals by preconditioning rather than scaling the ODE. But more on that later.

We advocate the "linear" schedule (same as DDIM):

$\sigma(t) = t$

$s(t) = 1$

$$d\mathbf{x} = -t\,\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\,dt$$

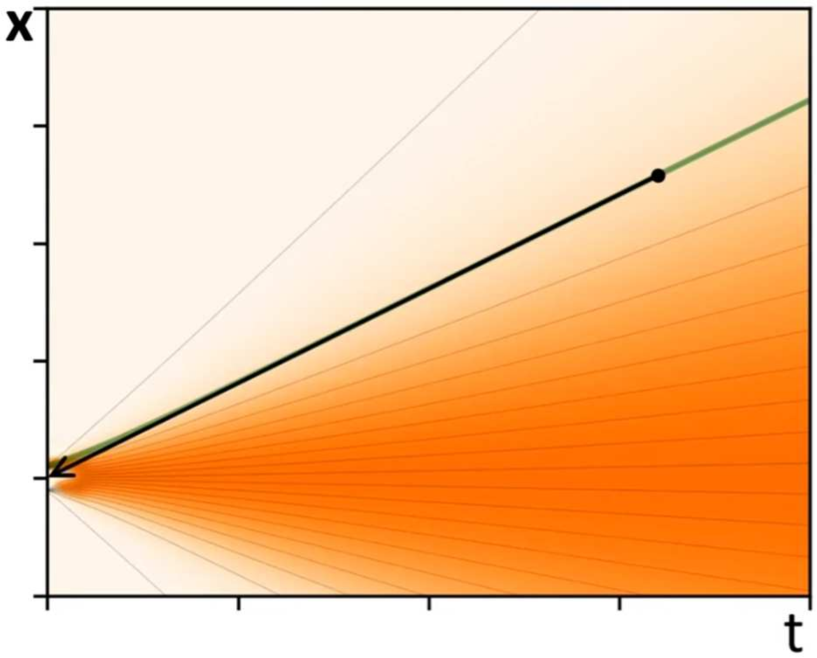
We'll be leaving the scaling for Neural Network parameterization. The reason for that is that the scaling also introduces unwanted curvature into these lines.

As a further with this the ODE becomes very simple

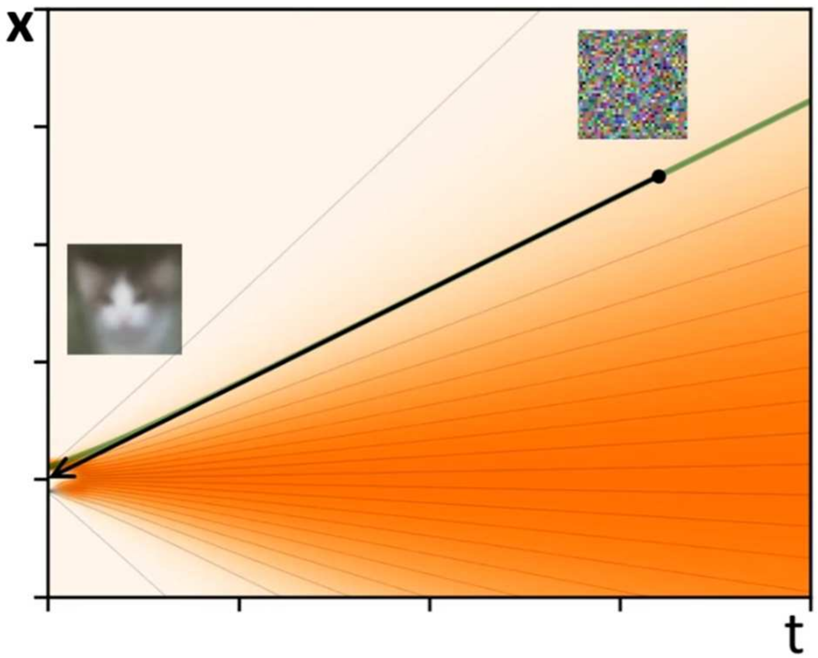
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Noise Schedule $\sigma(t)$ **Scaling Schedule $s(t)$**



This schedule allows us to take long steps



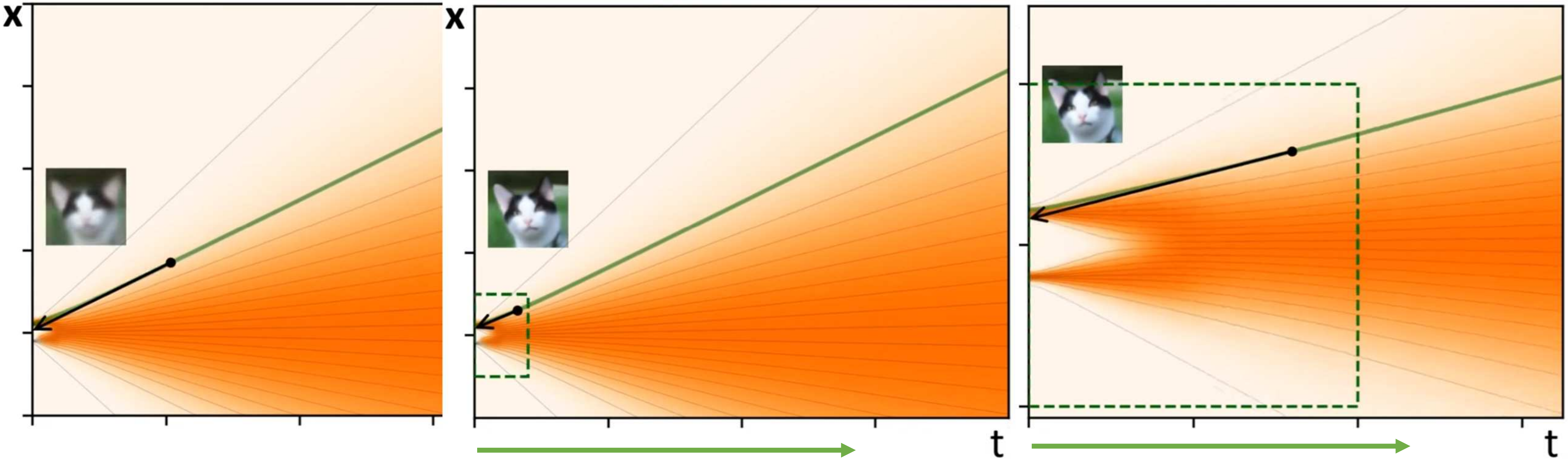
If I take a step directly to time zero, then with only these schedules, the tangent is pointing directly to the denoise output.

We advocate the “linear” schedule (same as DDIM):

$\sigma(t) = t$
$s(t) = 1$

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.



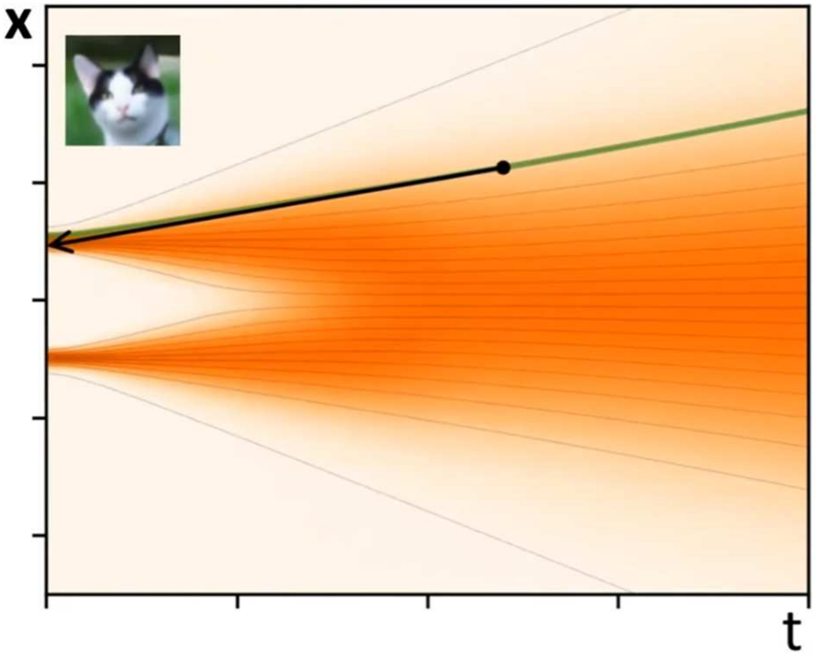
The denoise output changes only very slowly during the sampling process.

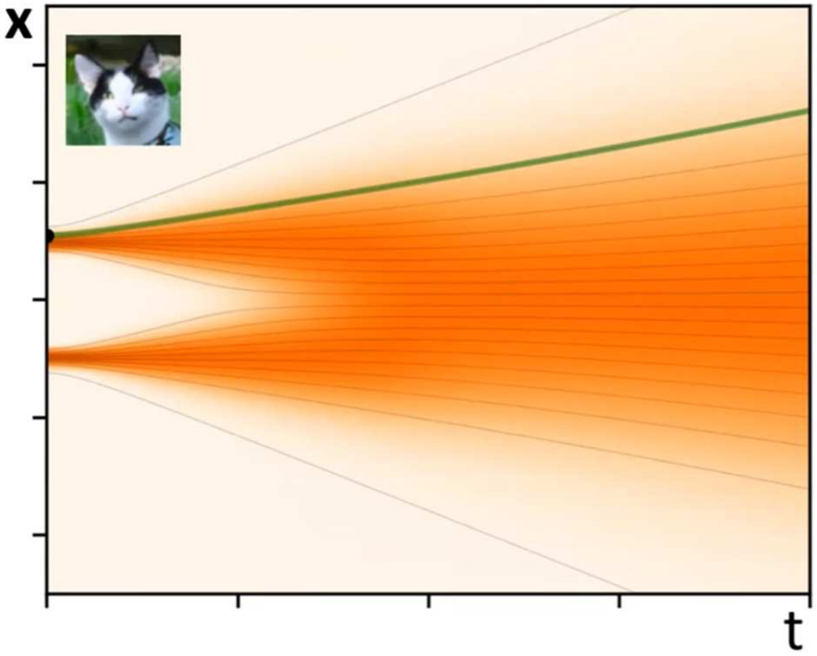
Mean that the direction you are going to doesn't change almost at all

So it means you can take long bold steps and you can consequently only take a few steps or many fewer steps than with the alternatives.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



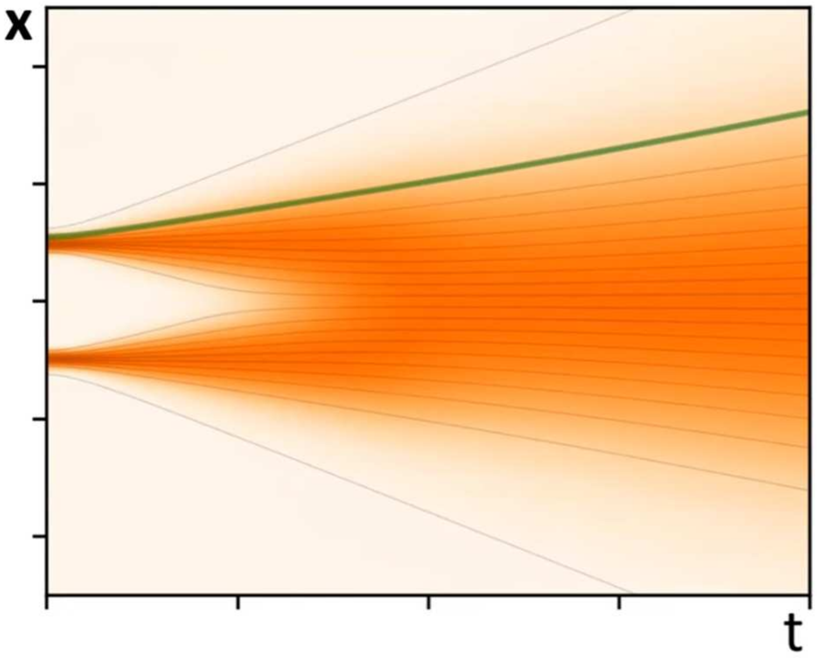


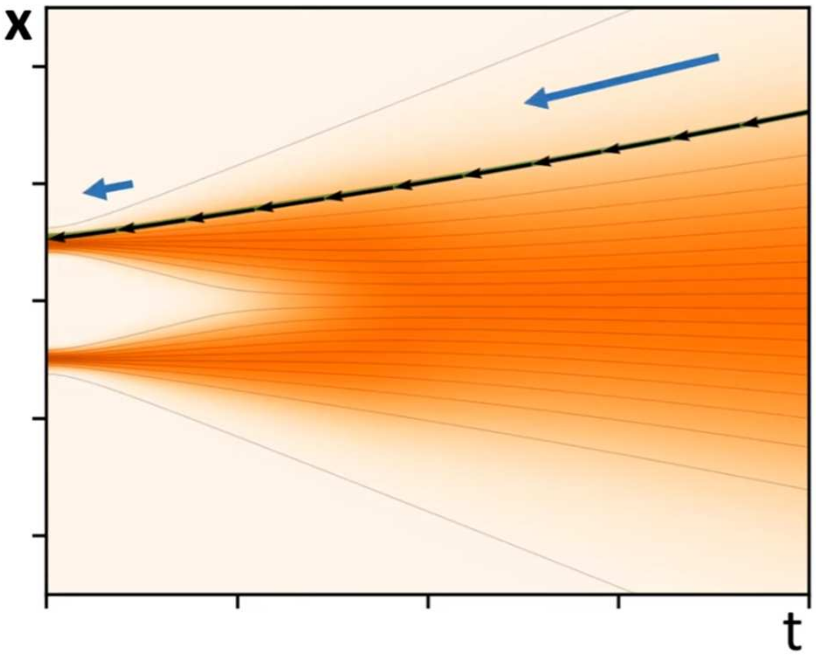
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Varying step length

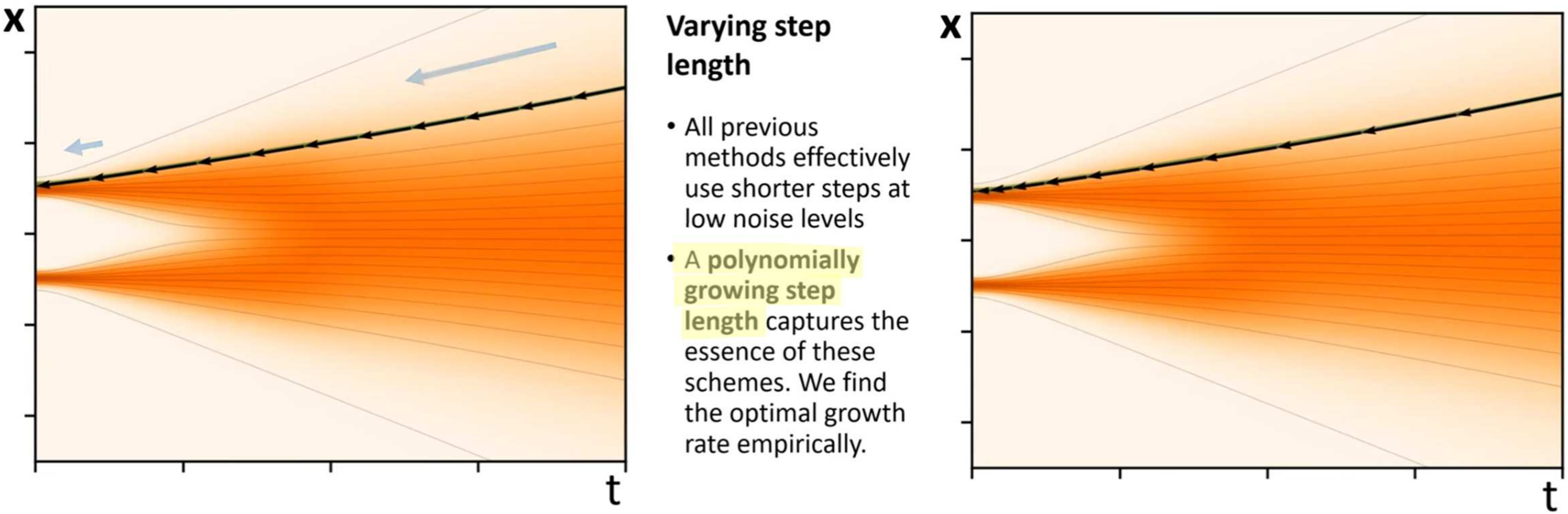
Take different length steps at different stages of the generation





EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

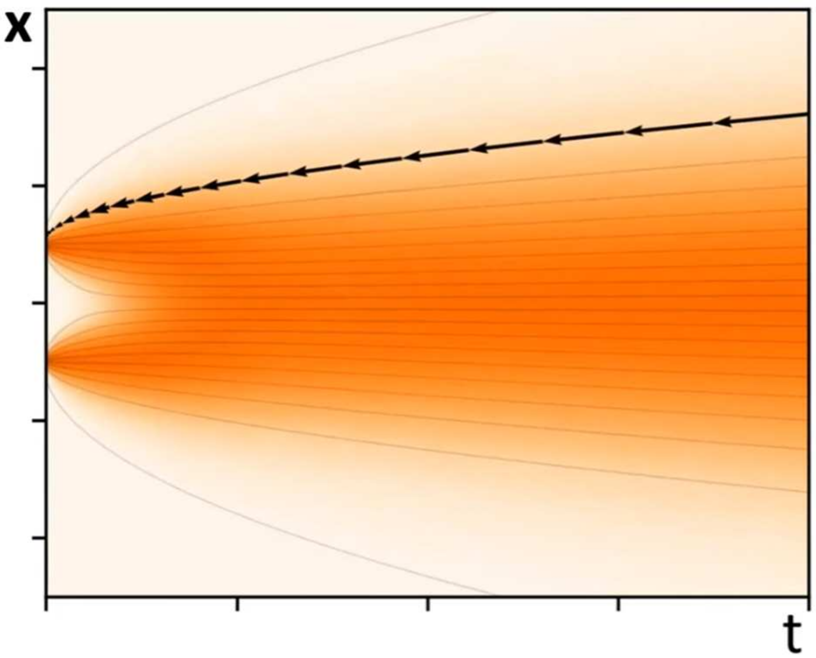
Varying step length



Youtube Presentaiton

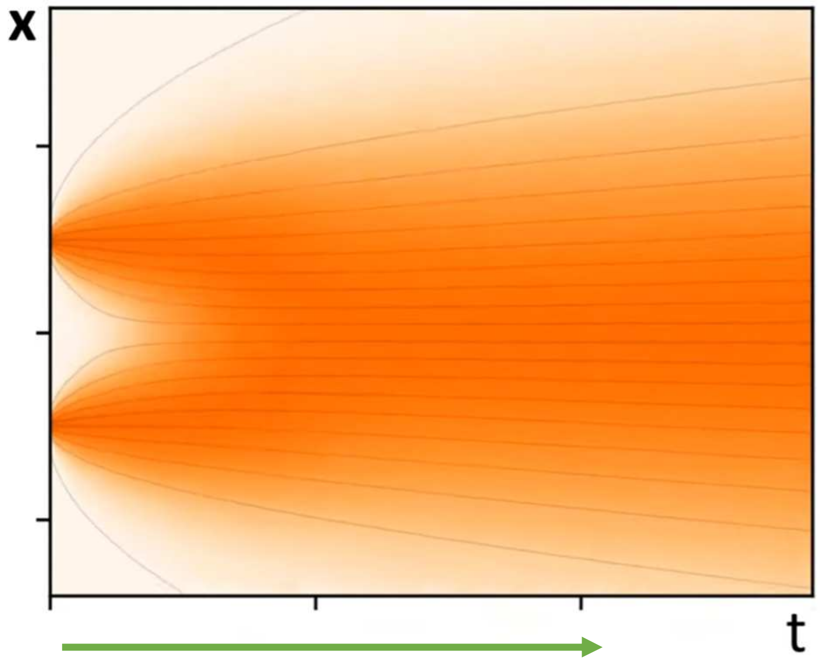
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Higher-order solvers



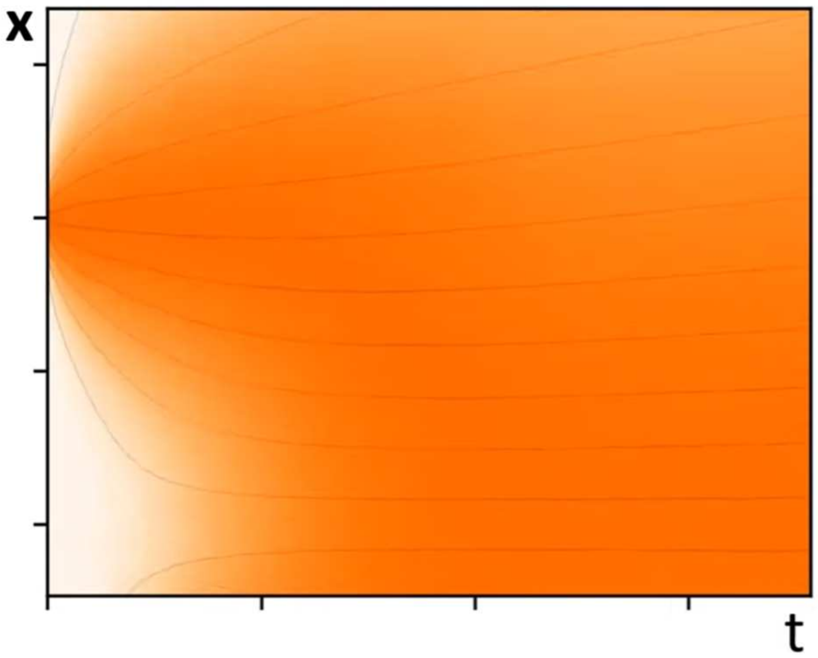
Higher-order solvers

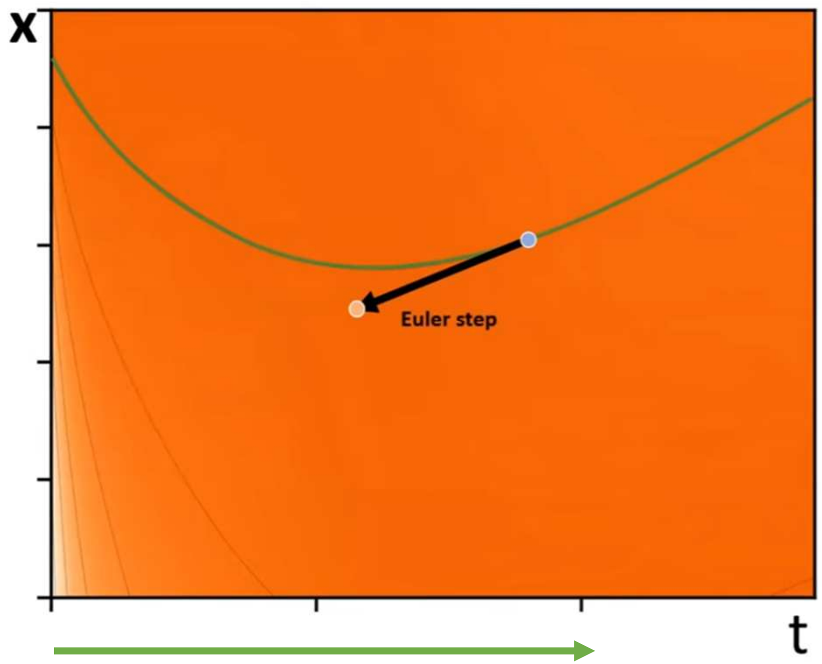
- Clever sub-steps → higher accuracy, higher cost
- Empirically, 2nd order **Heun method** strikes best balance



ODE framework allows you to do which the Markov chain formulas uses the higher-order solvers, so there is going to be curvature.

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.





Higher-order solvers

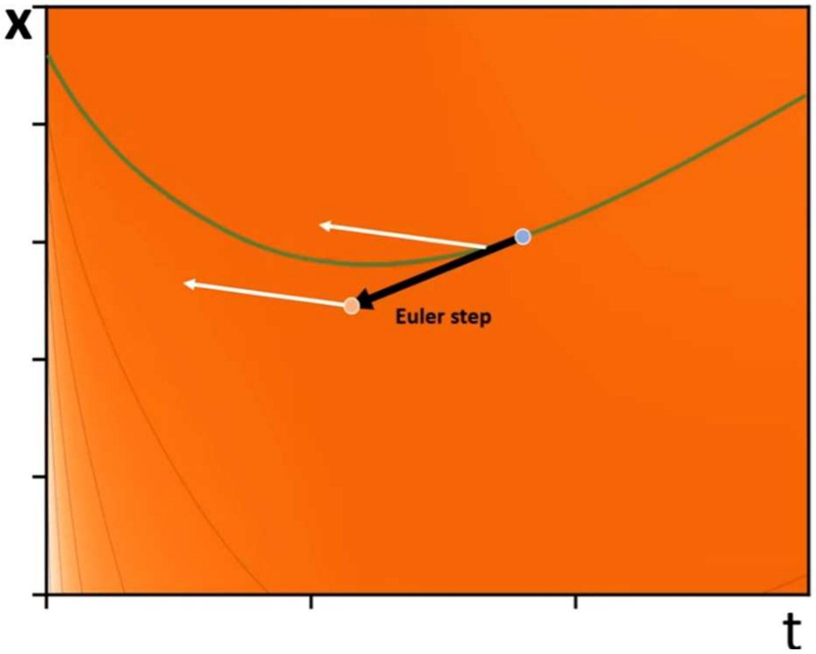
- Clever sub-steps → higher accuracy, higher cost
- Empirically, 2nd order **Heun method** strikes best balance

It can be rapid at places, so you can fall of the track if you just follow the tangent by using the Euler step

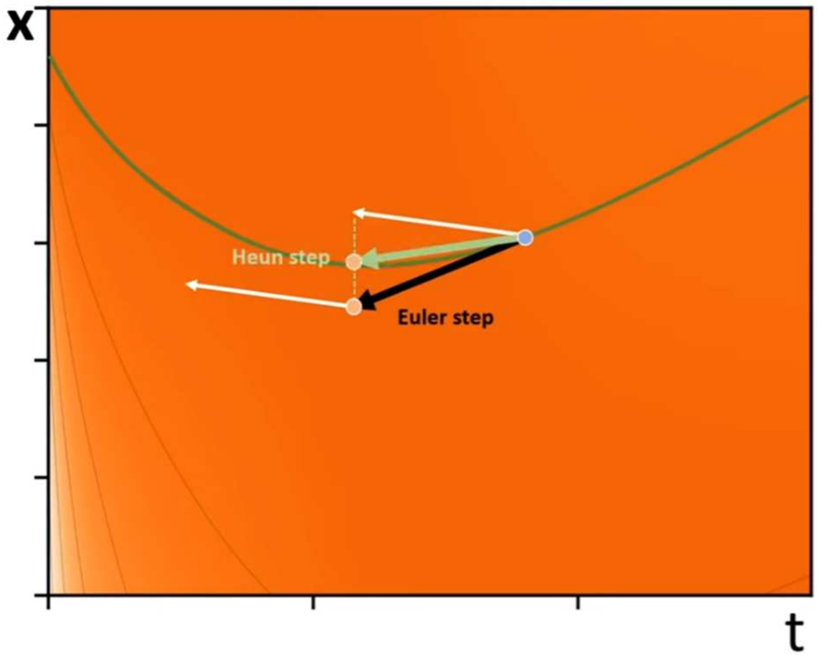
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models]," NeurIPS 2022.

2nd order Heun step



Heun's method : 1) Take the second tentative step and move it back to where you started from, 2) Take average of that and the initial one



Strike the best balance between these higher-order methods

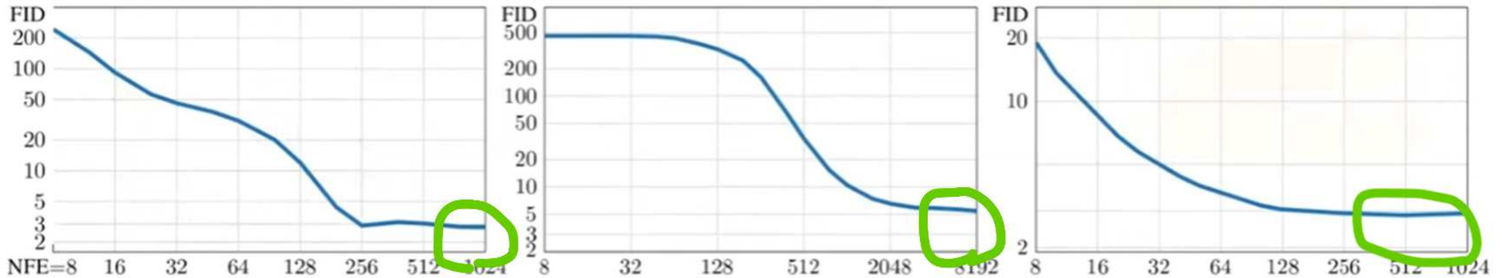
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

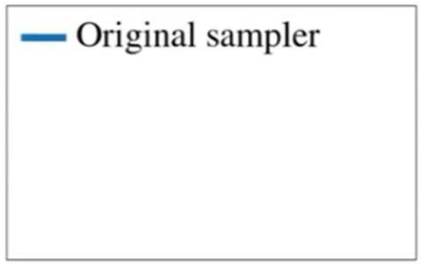
		VP	VE	iDDPM + DDIM	Ours
Sampling					
ODE solver		Euler	Euler	Euler	2nd order Heun
Time steps	$t_{i<N}$	$1+\frac{i}{N-1}(\epsilon_s-1)$	$\sigma_{\max}^2\left(\sigma_{\min}^2/\sigma_{\max}^2\right)^{\frac{i}{N-1}}$	$u_{\lfloor j_0+\frac{M-1-j_0}{N-1}i+\frac{1}{2}\rfloor}$, where $u_M=0$, $u_{j-1}=\sqrt{\frac{u_j^2+1}{\max(\bar{\alpha}_{j-1}/\bar{\alpha}_j,C_1)}-1}$	$\left(\sigma_{\max}{}^{\frac{1}{\rho}}+\frac{i}{N-1}\left(\sigma_{\min}{}^{\frac{1}{\rho}}-\sigma_{\max}{}^{\frac{1}{\rho}}\right)\right)^{\rho}$
Schedule	$\sigma(t)$	$\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}-1}$	$\sqrt{t}$	t	t
Scaling	$s(t)$	$1/\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}}$	1	1	1
Network and preconditioning					
Output scaling $c_{\text{out}}(\sigma)$					

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Evaluation of discretization & solver



(a) Uncond. CIFAR-10, VP ODE (b) Uncond. CIFAR-10, VE ODE (c) Class-cond. ImageNet-64, DDIM

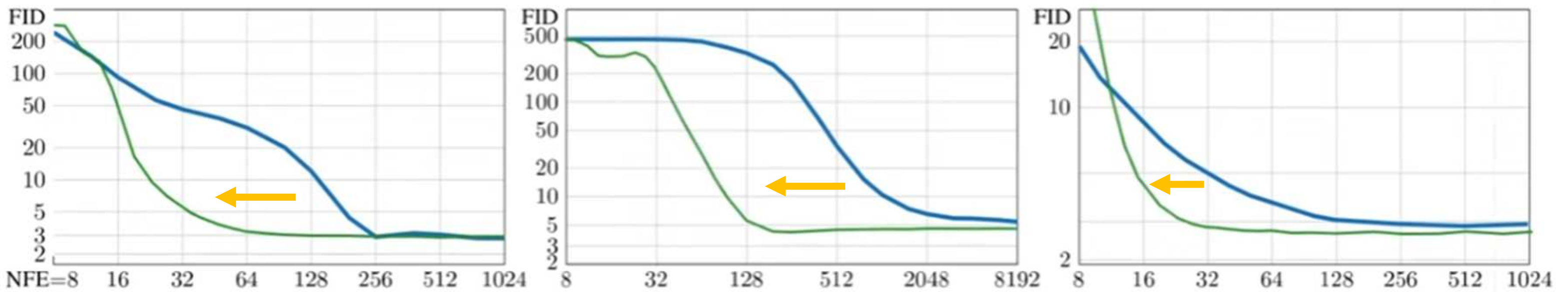


Need to take something like a hundreds or even thousands of steps to get kind of saturated quality and to get the best quality that model gives you

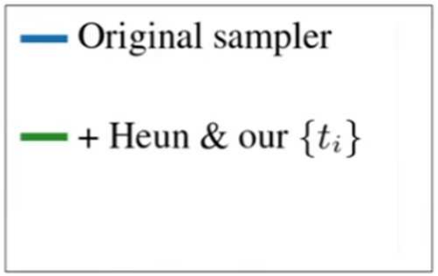
NFE (the number of neural function evaluations) : Forward pass 중에 전체 모델 파라미터가 몇번이나 계산되었는가를 의미하는 지표

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Evaluation of discretization & solver



(a) Uncond. CIFAR-10, VP ODE (b) Uncond. CIFAR-10, VE ODE (c) Class-cond. ImageNet-64, DDIM

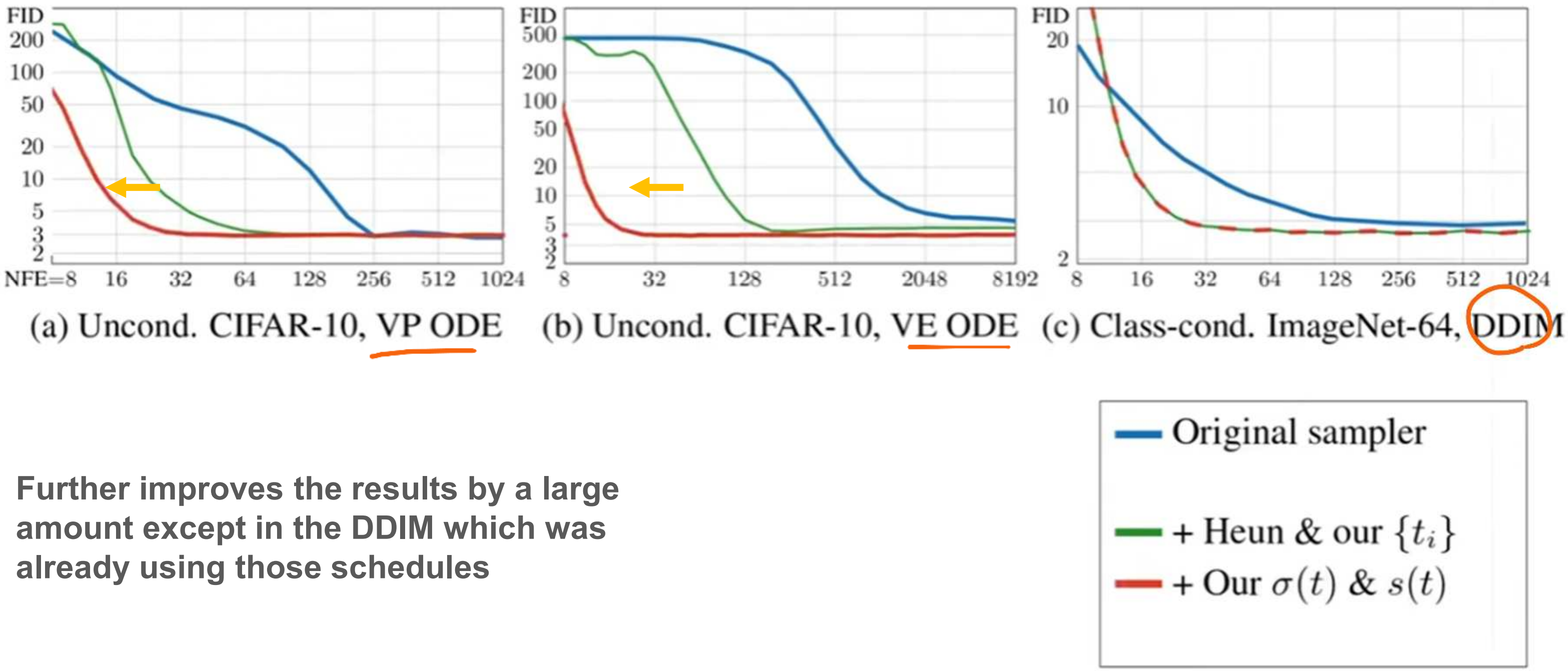


Heun & Our discretization schedule : Go to from hundreds to dozens of evaluations

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Evaluation of schedule

Noise schedule and Scaling schedule

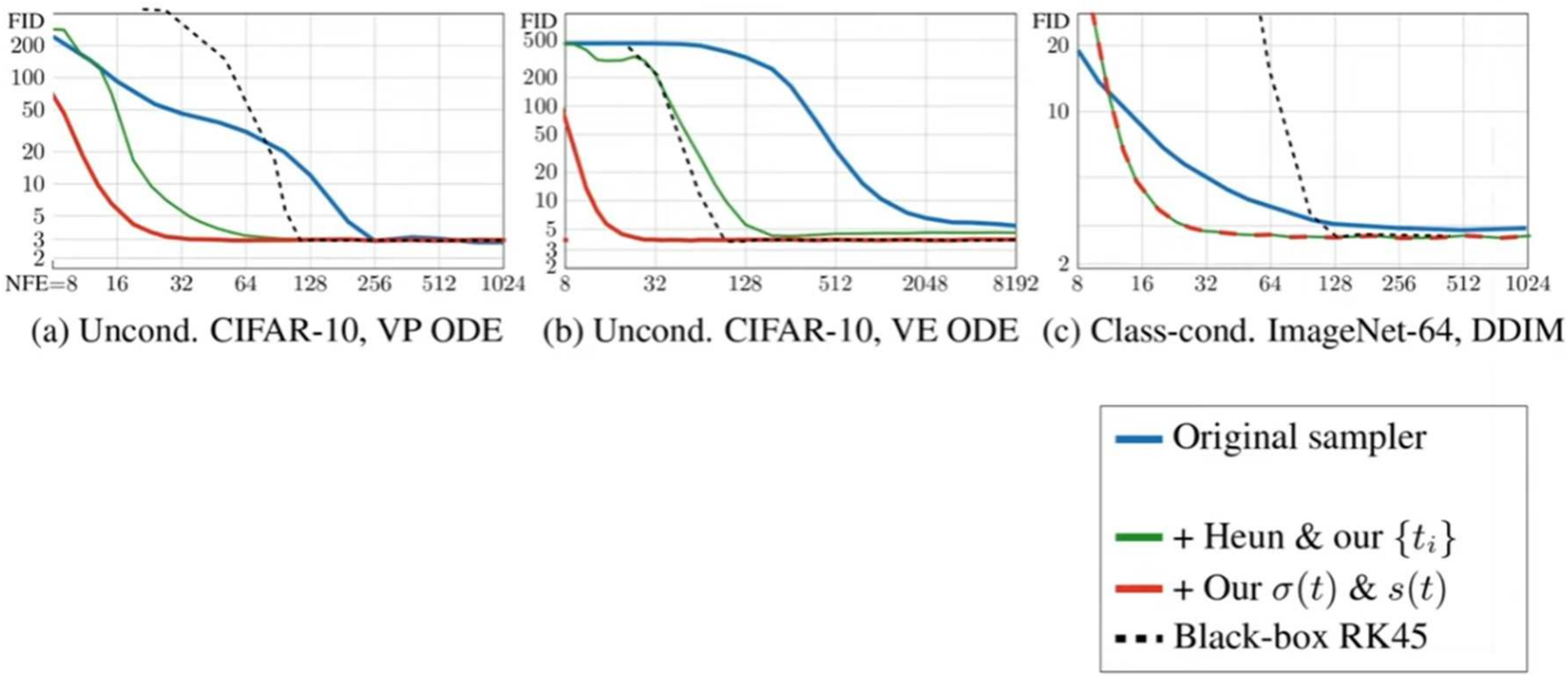


Further improves the results by a large amount except in the DDIM which was already using those schedules

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Evaluation of schedule



Youtube Presentaiton

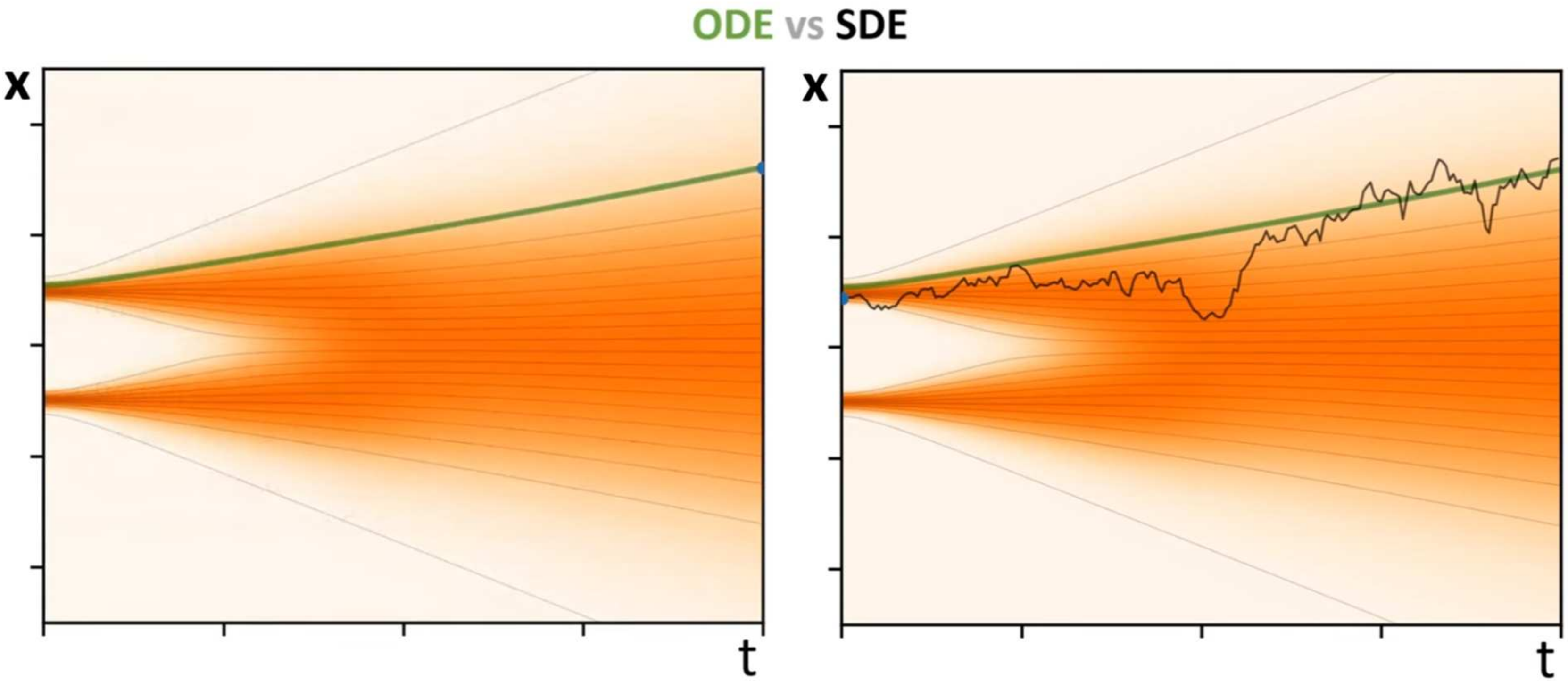
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models]," NeurIPS 2022.

Outline

- Part I: **Common framework**
  - Identifying the moving parts in existing work
- Part II: **Deterministic sampling**
  - Solving the ODE efficiently
- Part III: **Stochastic sampling**
  - Why SDE's? How to do stochastic stepping?
- Part IV: **Preconditioning and training**
  - How to train the CNN used in evaluating a step?

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



ODE vs SDE

- ODE gradually fades the noise out without changing it
- SDE injects fresh noise throughout evolution
  → *noise* is not only reduced, but also *replaced* at some rate
- Let's first pick this apart, and then see why it's useful

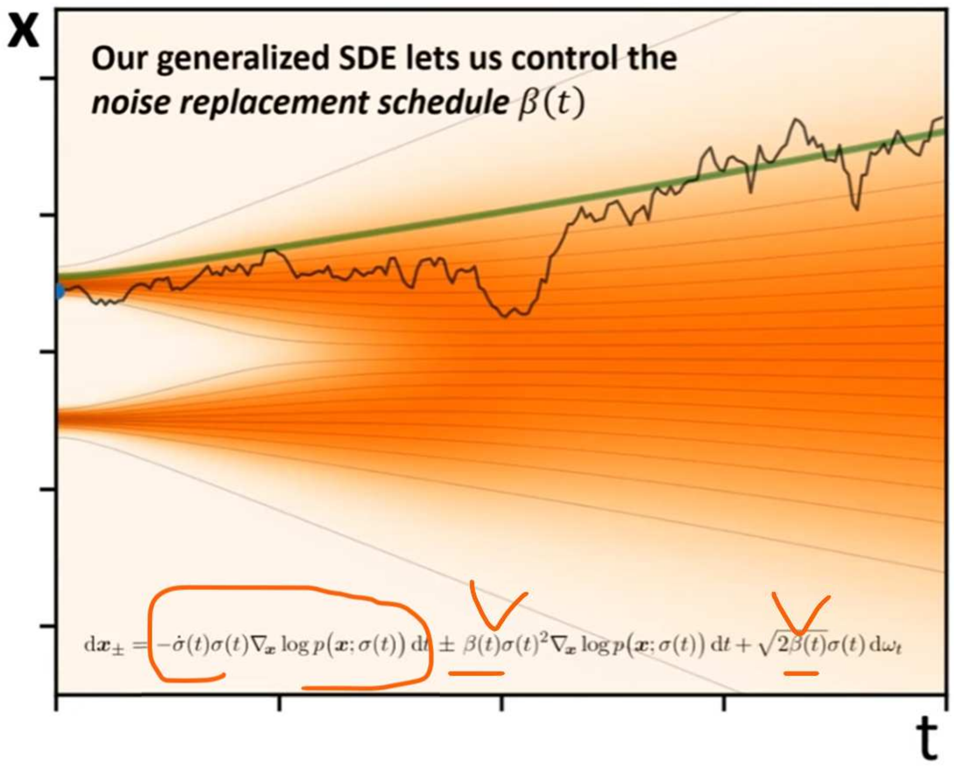
Instead of following these nice smooth flow trajectories, the SDE is sort as some kind of exploration around that baseline so it can be interpreted as replacing the noise and reducing it.

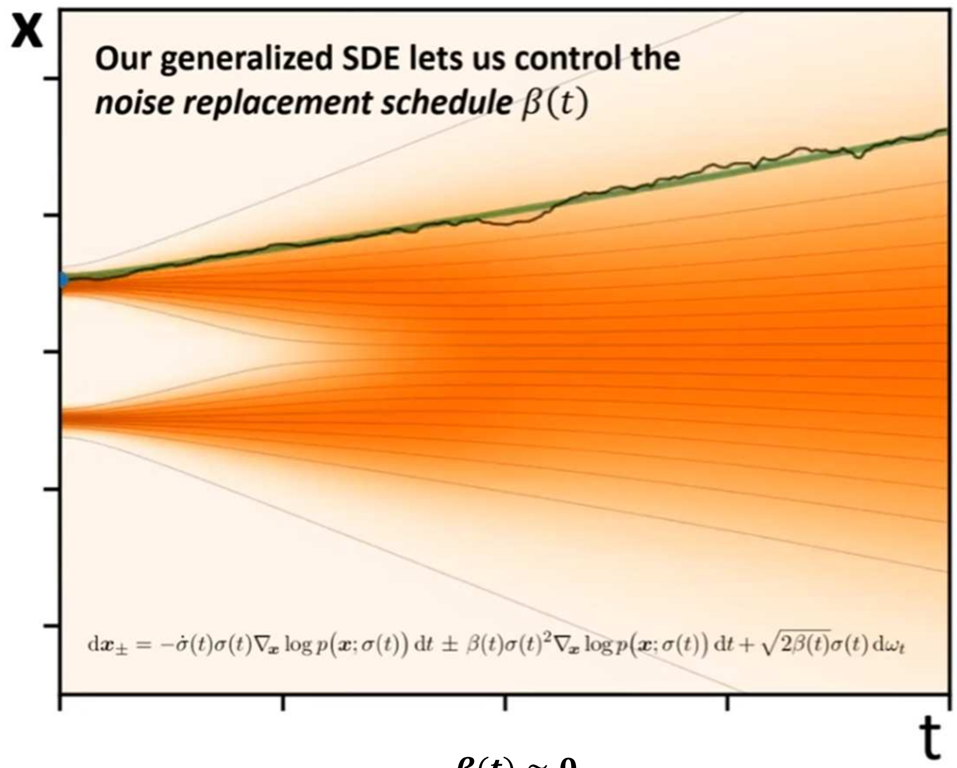
In practice you tend to get better results when you use the SDE instead of the ODE at least in previous works

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models]," NeurIPS 2022.

Generalized SDE





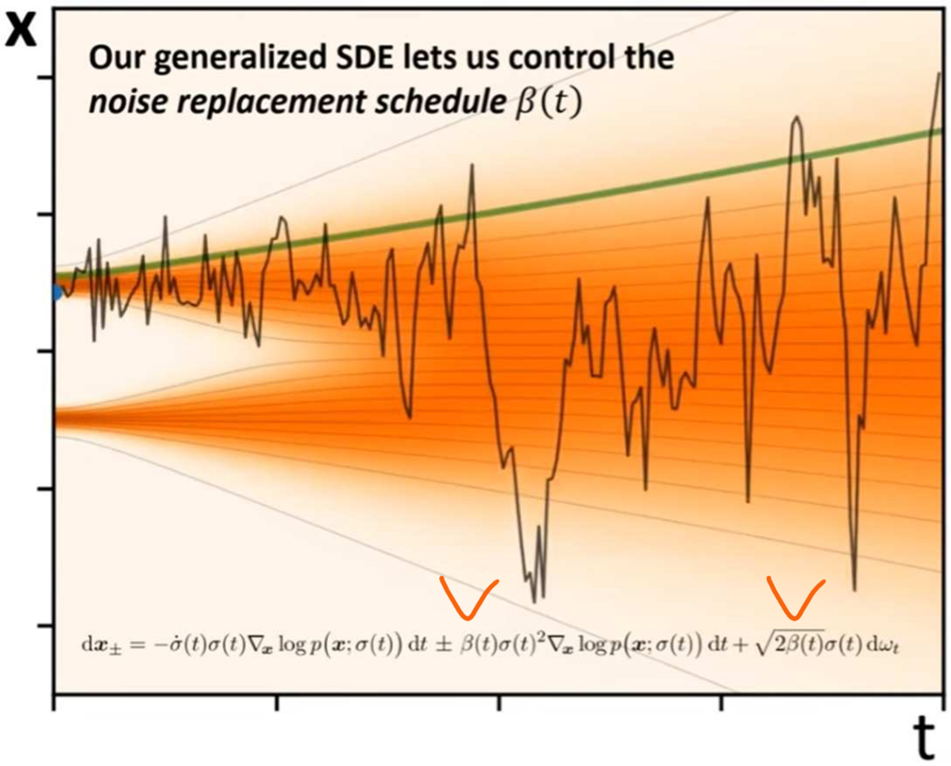
$\boldsymbol{\beta(t) \approx 0}$

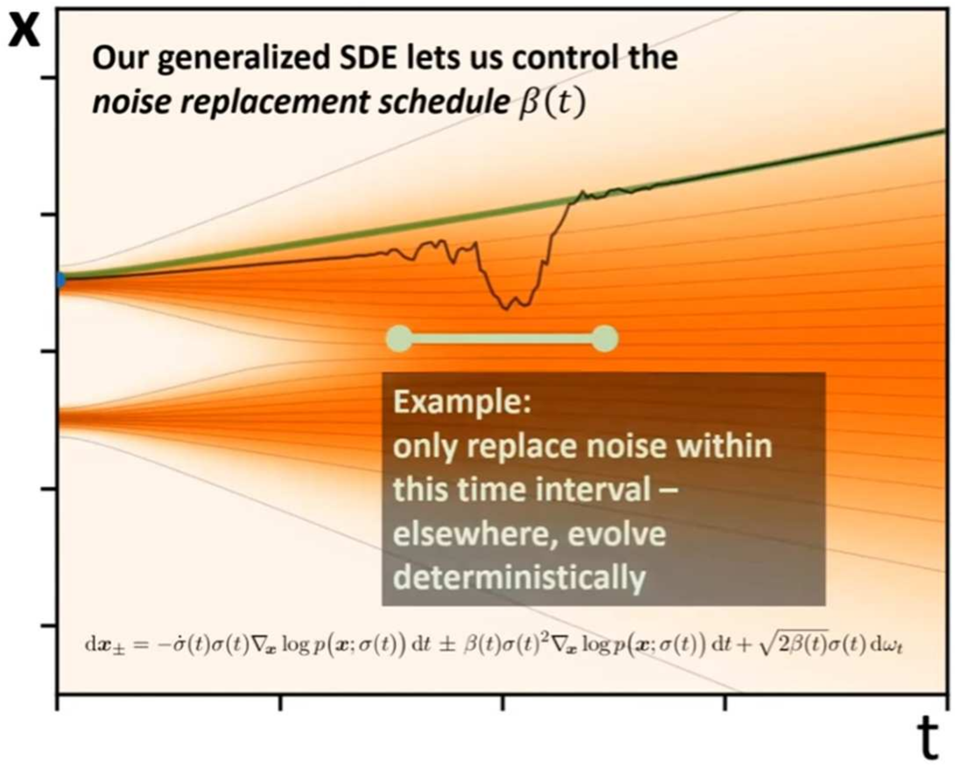
Generalized SDE allows you to specify the strength of this exploration by this noise replacement schedule $\boldsymbol{\beta(t)}$

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Generalized SDE

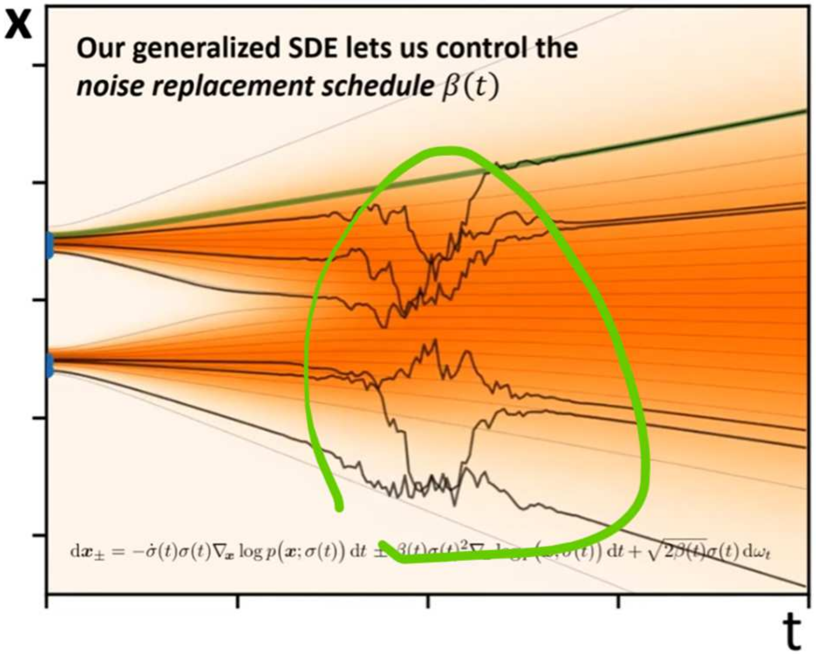


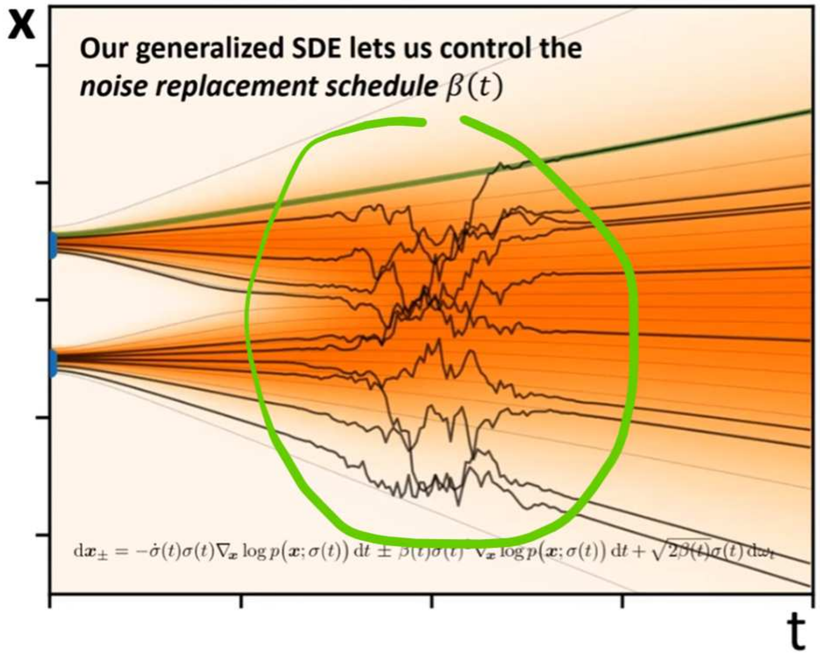


$\boldsymbol{\beta(t)}$ ↑ : Boosting → More exploration

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.





Generalized SDE

- Nice trick, but doesn't it still just go through the same distributions as ODE?
- Empirically, stochasticity does improve results. Why?

$$\mathrm{d}\boldsymbol{x}_\pm = \underbrace{-\dot\sigma(t)\sigma(t)\nabla_{\boldsymbol{x}} \log p\big(\boldsymbol{x};\sigma(t)\big)\,\mathrm{d}t}_{\text{probability flow ODE (Eq. 1)}} \boxed{\pm \underbrace{\underbrace{\beta(t)\sigma(t)^2\nabla_{\boldsymbol{x}} \log p\big(\boldsymbol{x};\sigma(t)\big)\,\mathrm{d}t}_{\text{deterministic noise decay}} + \underbrace{\sqrt{2\beta(t)}\sigma(t)\,\mathrm{d}\omega_t}_{\text{noise injection}}}_{\text{Langevin diffusion SDE}}},$$

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

$$d\boldsymbol{x}_{\pm} = \underbrace{-\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{x}} \log p\big(\boldsymbol{x};\sigma(t)\big)\,dt}_{\text{probability flow ODE}} \pm \underbrace{\underbrace{\beta(t)\sigma(t)^2\nabla_{\boldsymbol{x}} \log p\big(\boldsymbol{x};\sigma(t)\big)\,dt}_{\text{deterministic noise decay}} + \underbrace{\sqrt{2\beta(t)}\sigma(t)\,d\omega_t}_{\text{noise injection}}}_{\text{Langevin diffusion SDE}}$$

Shapes the trajectories, such that they pass through the desired distributions p_t at time t

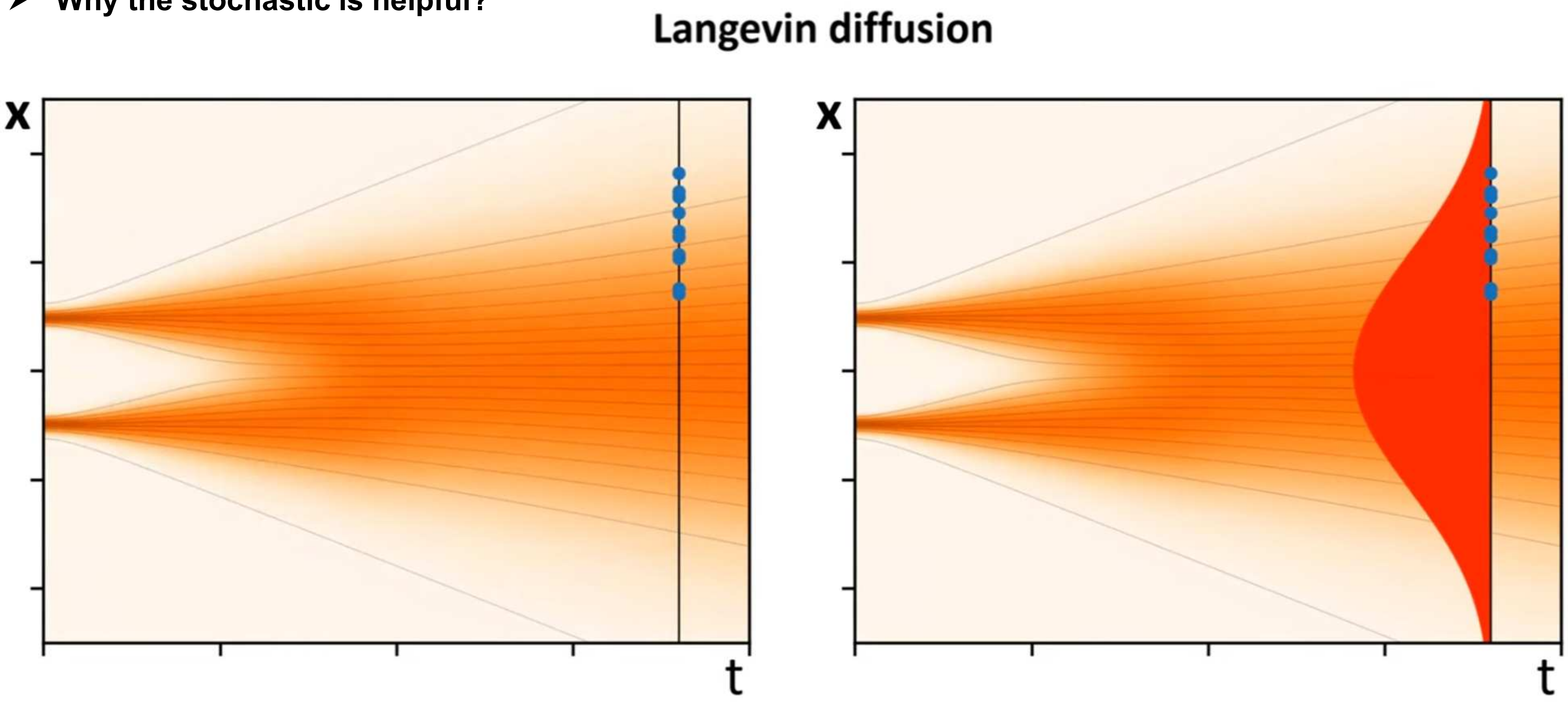
Randomly explores the distribution p_t at time t, driving the samples towards it

Driving towards the distribution and making it follow the flow lines

It makes the samples explore the distribution.
If the samples are not distributed correctly, it will reduce that error. Healing property
→ Because we do make errors during the sampling, it can actively corrects for them

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.

- **Why the stochastic is helpful?**

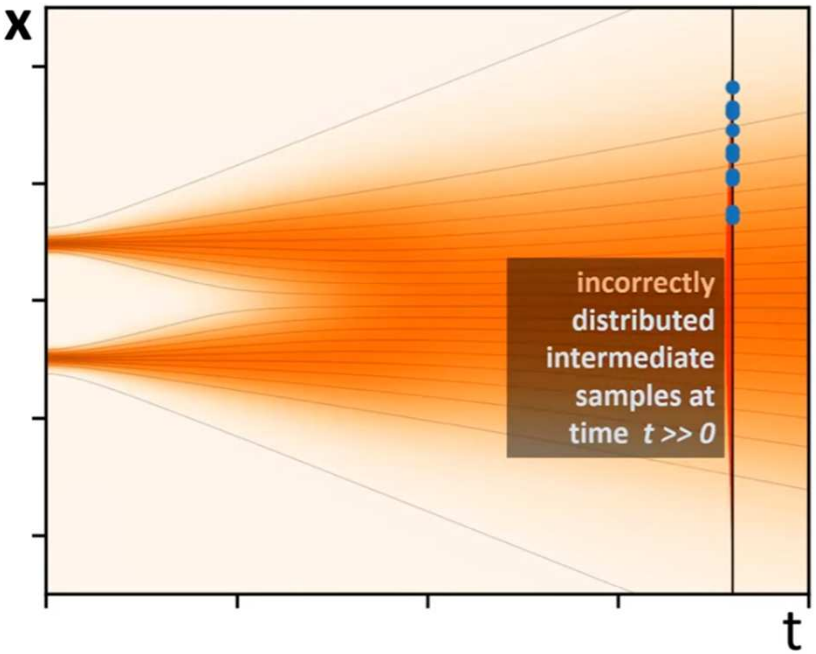


Samples blue dots in Bad case (not follow the underlying distribution at all)

Youtube Presentaiton

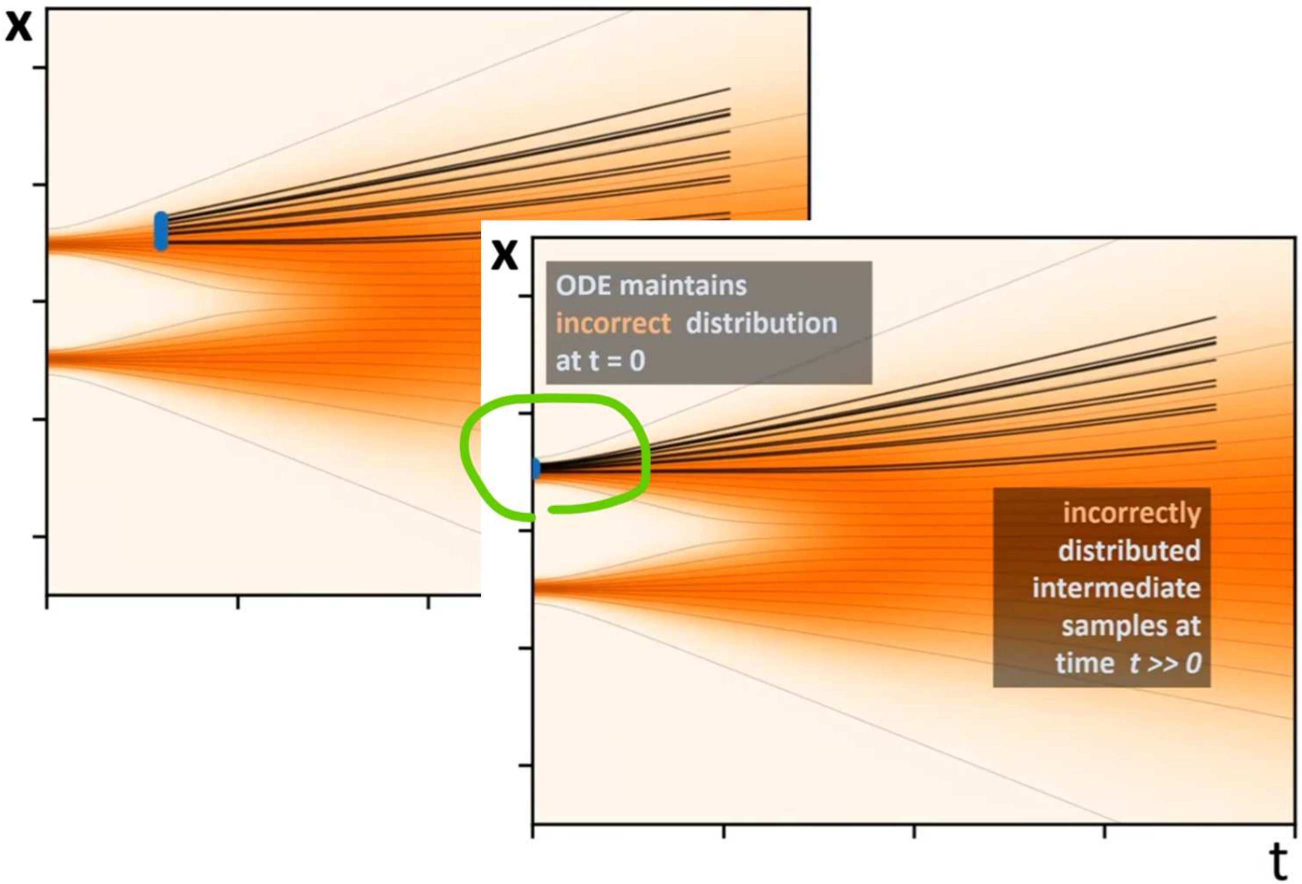
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

- **Why the stochastic is helpful?**



Skewed to one side

If keep following ODE, do nothing to correct that skew and completely miss the other base data

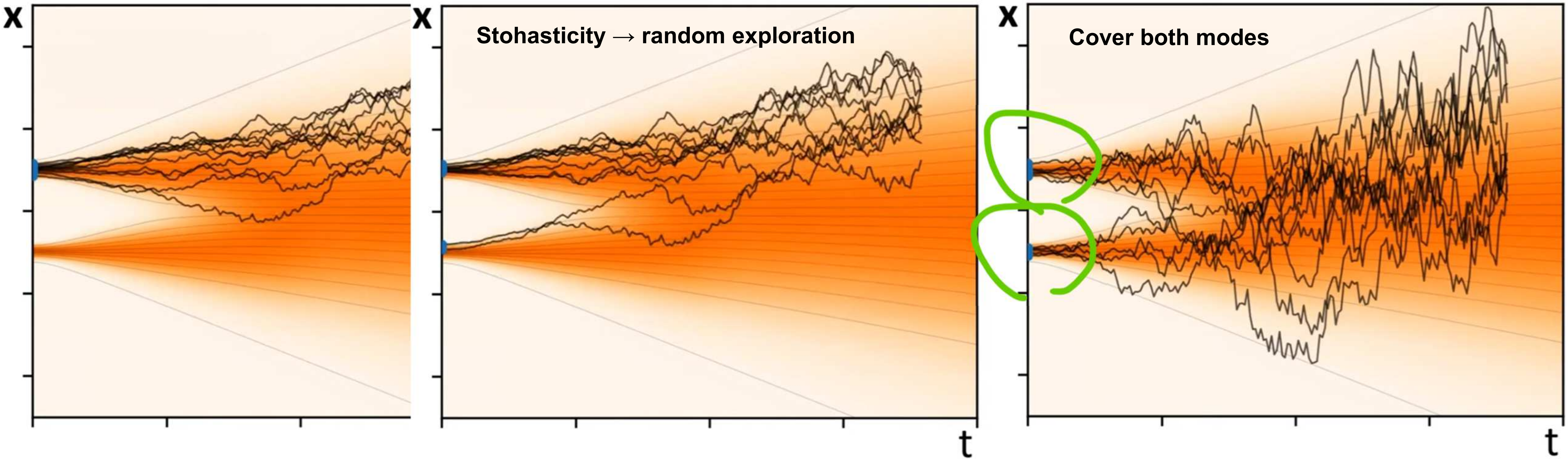


Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

➢ **Why the stochastic is helpful?**

Langevin diffusion

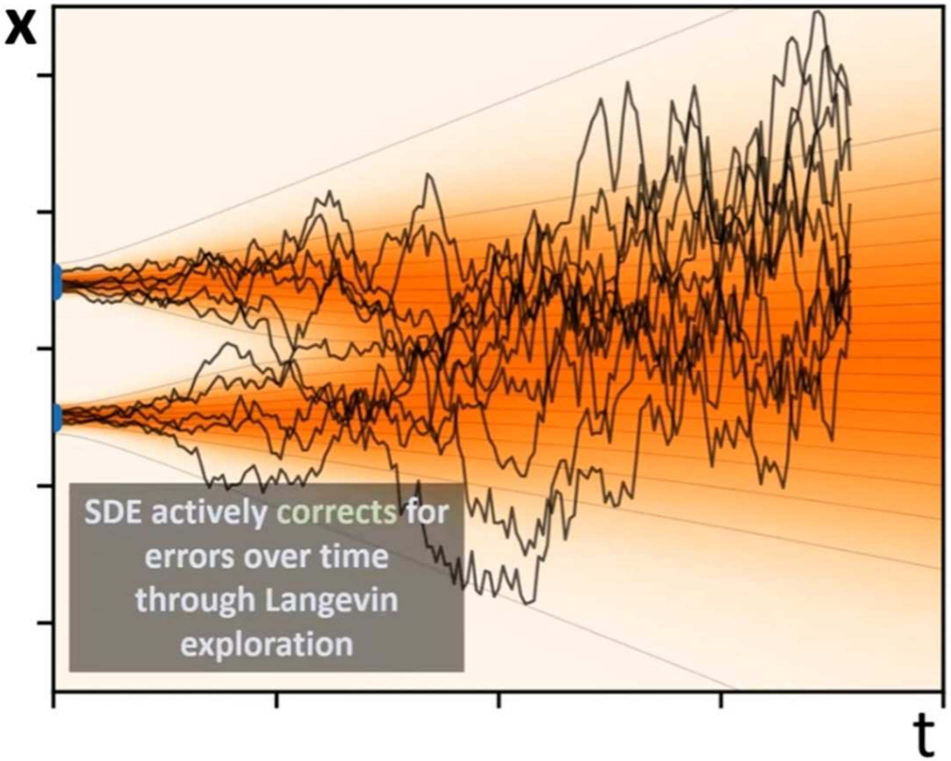


So these samples do this kind of random exploration and gradually forget where they came from and forget the error and initial position.

And now we've covered both modes for example in the generated images on left edge.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Langevin diffusion

- The main benefit of SDE over ODE is the implicit Langevin exploration
- Could we instead simply combine the higher-order ODE solver and Langevin diffusion?

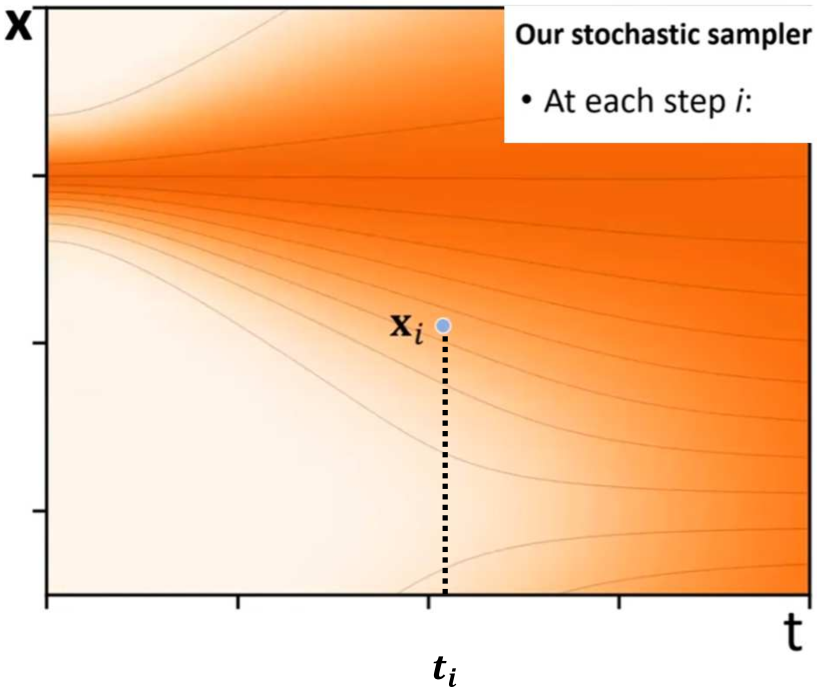
Answer → **Our stochastic sampler**

Youtube Presentaiton

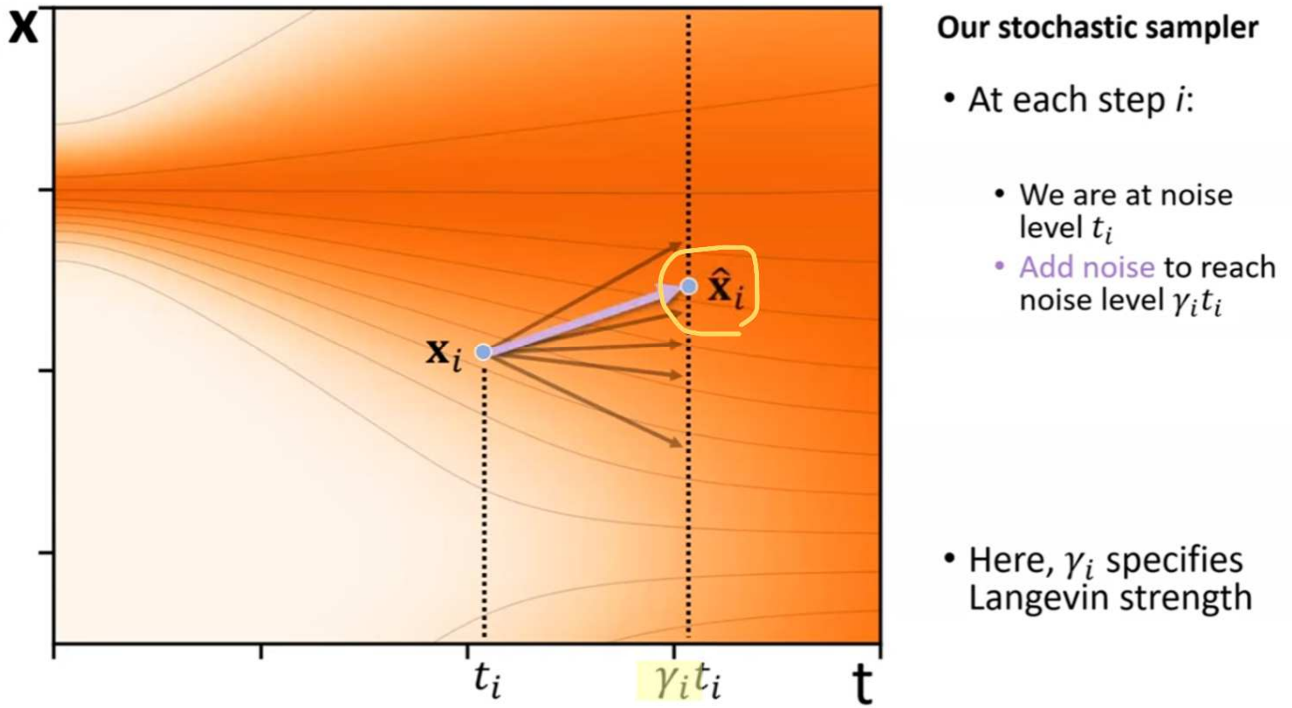
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Stochastic Sampler

Two substeps



1)

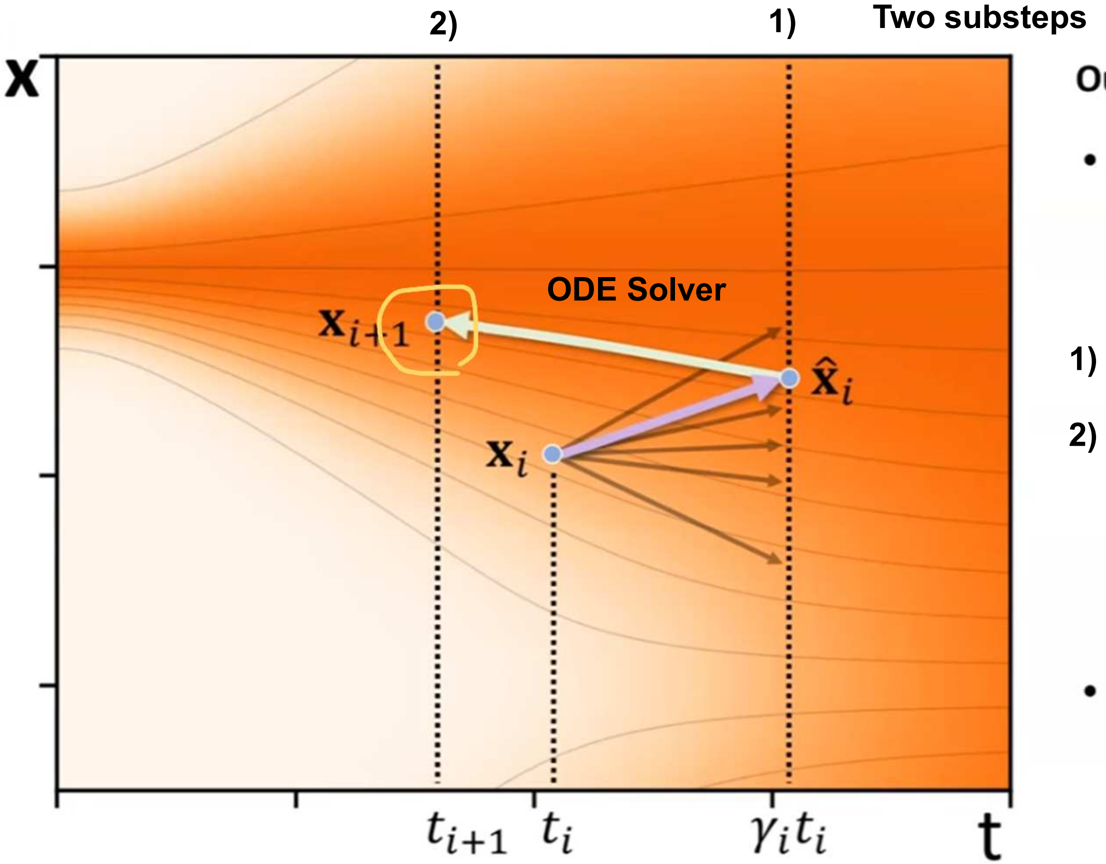


Noise level t_i → Act completely equivalent with the time

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Stochastic Sampler



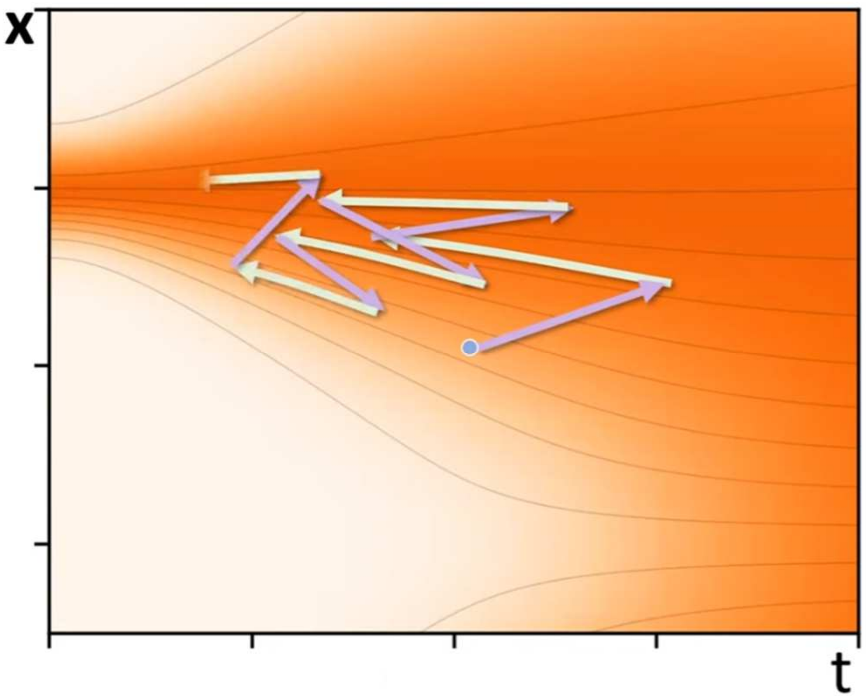
Our stochastic sampler

- At each step *i*:
  - We are at noise level t_i
  - 1) Add noise to reach noise level $\gamma_i t_i$
  - 2) Solve ODE backwards to the next time step t_{i+1} (with single Heun step)
- Here, γ_i specifies Langevin strength

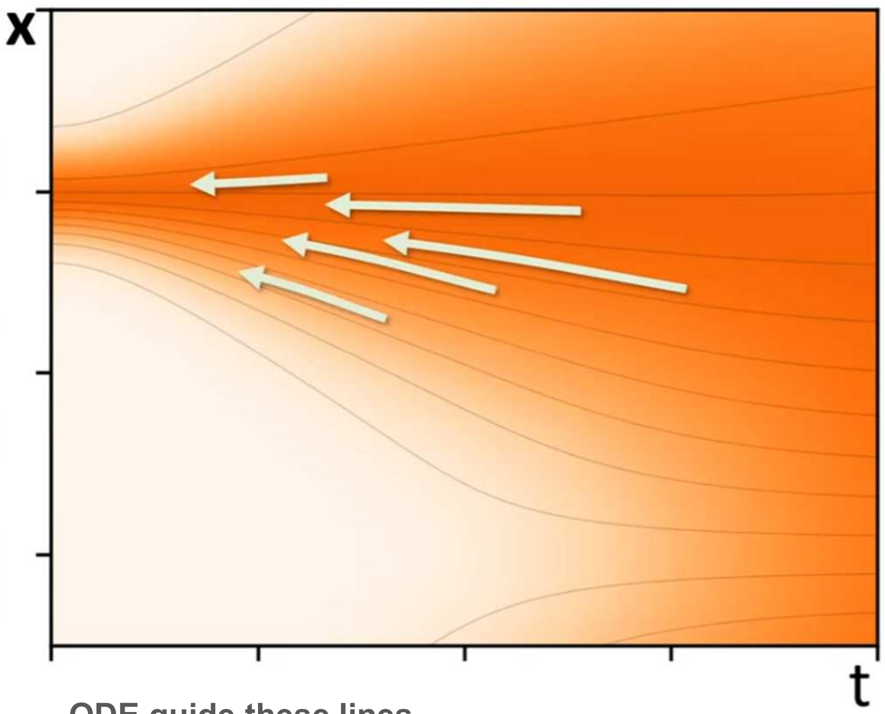
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Stochastic Sampler



Alternating between the noise addition and the Heun step as closer to time zero.



ODE guide these lines.
We now have the juttering which corrects errors.

Our stochastic sampler

- At each step i:
  - We are at noise level t_i
  - Add noise to reach noise level $\gamma_i t_i$
  - Solve ODE backwards to the next time step t_{i+1} (with single Heun step)
- Here, γ_i specifies Langevin strength

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Get these errors correct reaction but it's not actually free because the Langevin diffusion is also an approximation of some continuous thing.

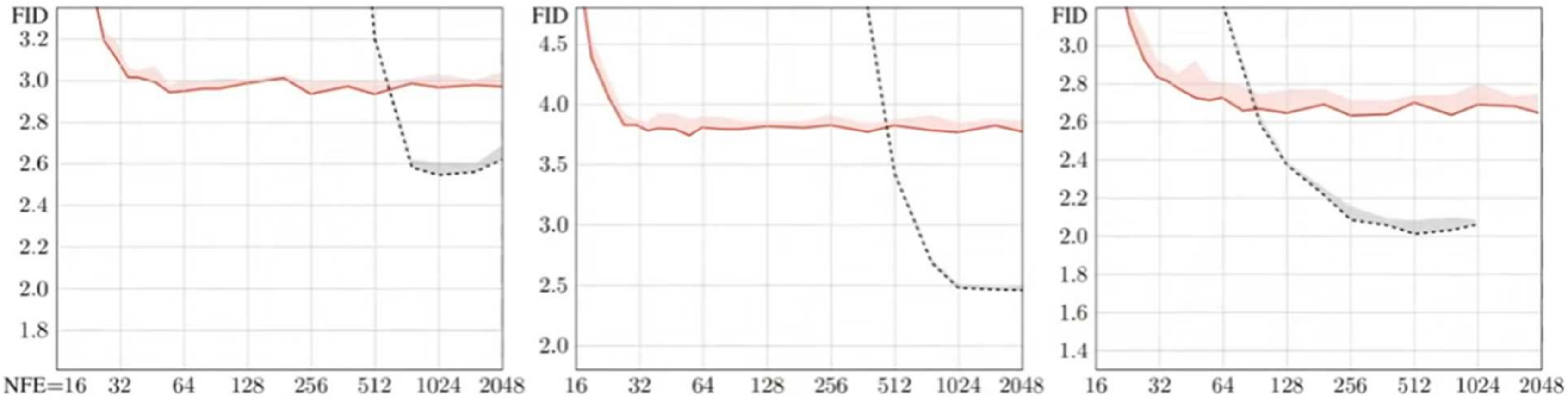
Quite delicate balance how much make error

Need to tune the amount of stochasticity on a data set per architectural basis

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Evaluation



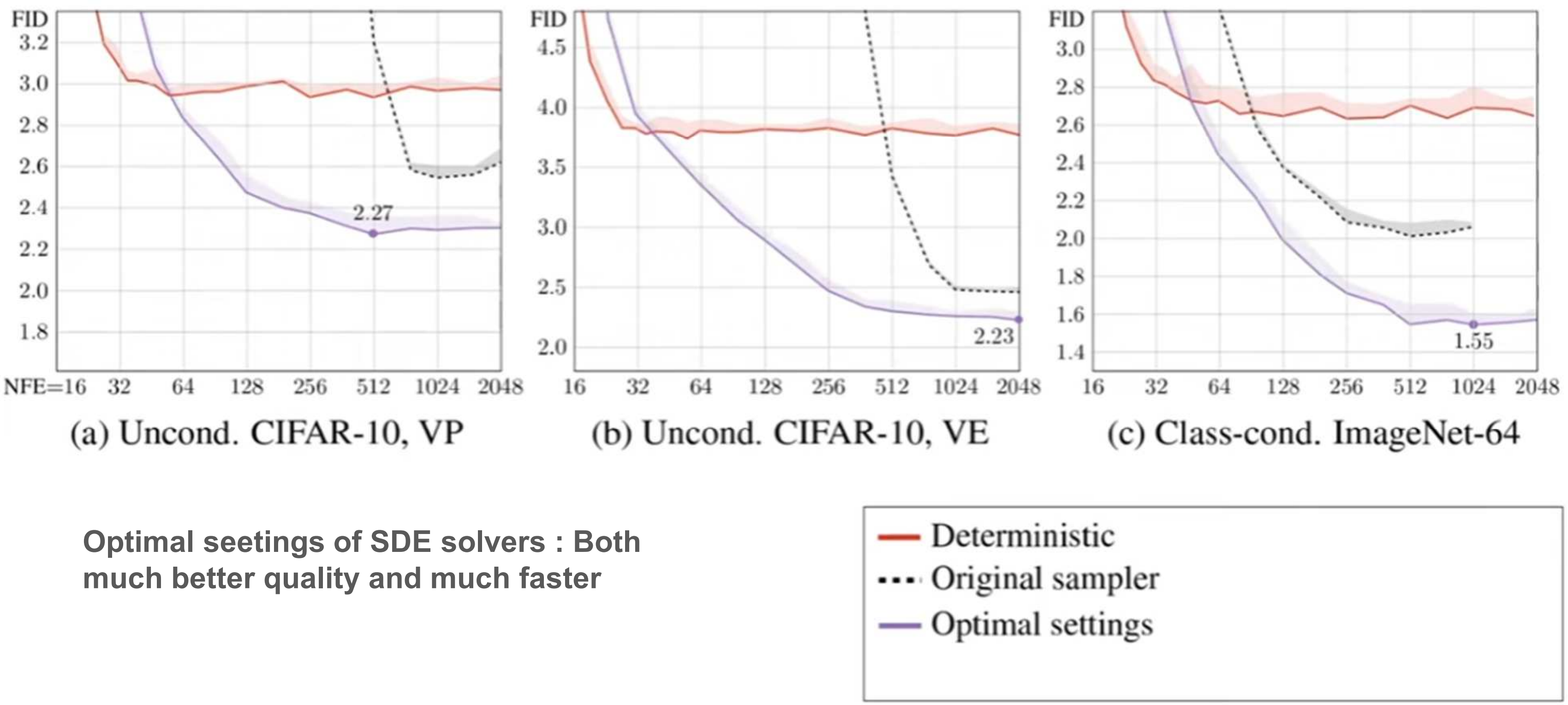
(a) Uncond. CIFAR-10, VP (b) Uncond. CIFAR-10, VE (c) Class-cond. ImageNet-64

SDE solvers are better but very slow

Deterministic **(ODE)**
Original sampler **(SDE)**

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Evaluation

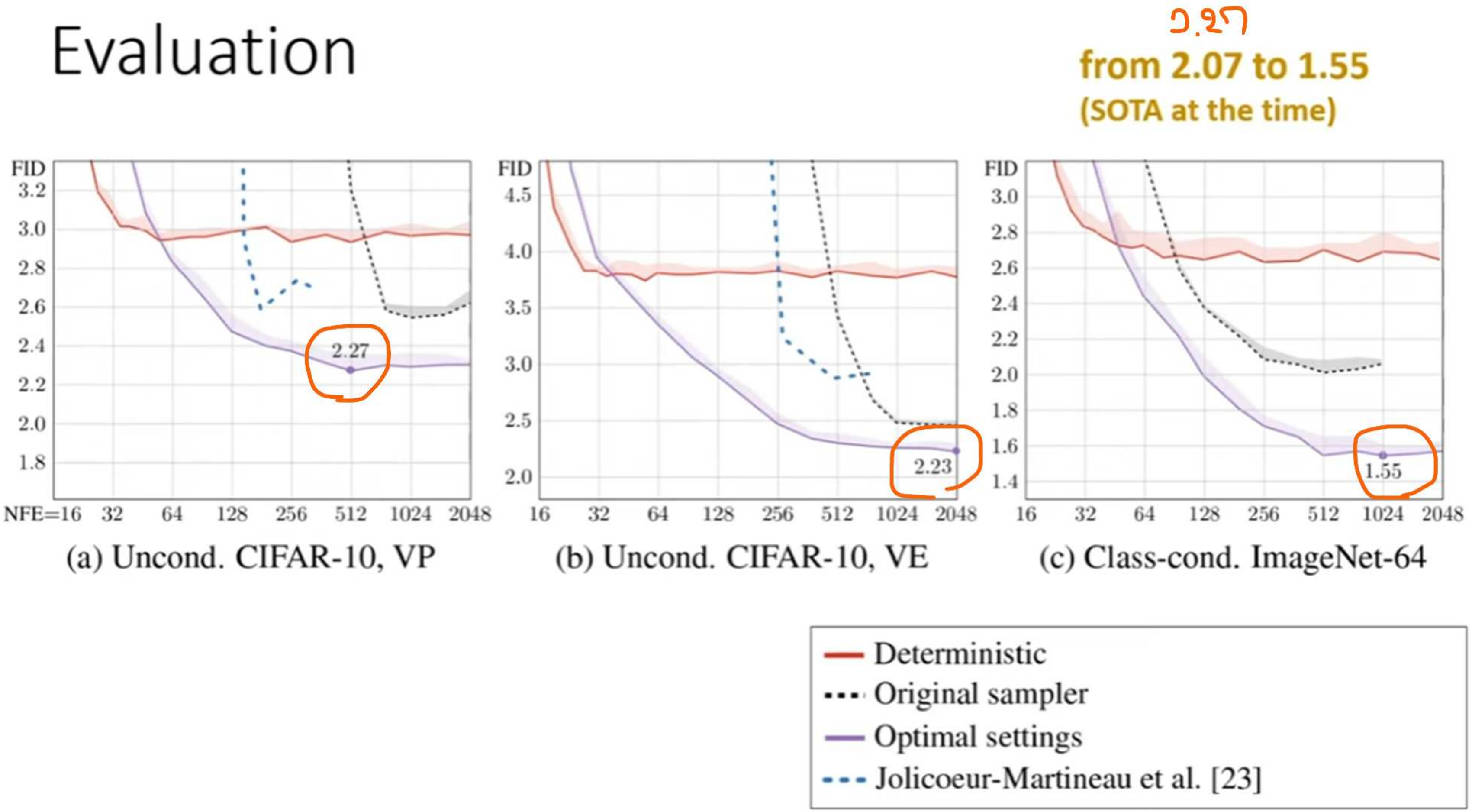


(a) Uncond. CIFAR-10, VP (b) Uncond. CIFAR-10, VE (c) Class-cond. ImageNet-64

Optimal seetings of SDE solvers : Both much better quality and much faster

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Outline

- Part I: **Common framework**
  - Identifying the moving parts in existing work
- Part II: **Deterministic sampling**
  - Solving the ODE efficiently
- Part III: **Stochastic sampling**
  - Why SDE's? How to do stochastic stepping?
- Part IV: **Preconditioning and training**
  - How to train the CNN used in evaluating a step?

Summary (정리)

Outline

- Part I: **Common framework**
  - Identifying the moving parts in existing work
- Part II: **Deterministic sampling**
  - Solving the ODE efficiently
- Part III: **Stochastic sampling**
  - Why SDE's? How to do stochastic stepping?
- Part IV: **Preconditioning and training**
  - How to train the CNN used in evaluating a step?

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Recall two sources of error

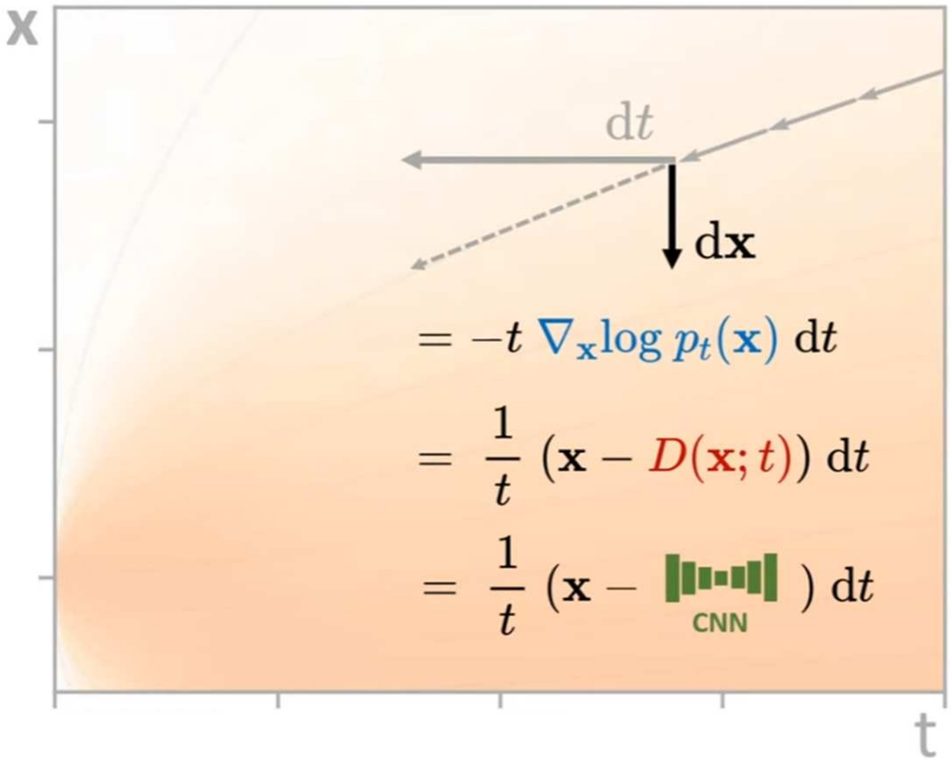
- Discretized steps in sampling
  - We studied this with pre-trained networks
- Inaccurate neural denoiser, a.k.a. score function

  Next up:
  - Improved network preconditioning (e.g., input and output scales)
  - Improved training (loss scaling, and what noise levels to train at?)

- We will not change the layer architecture, etc. (much)

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.



Recall…

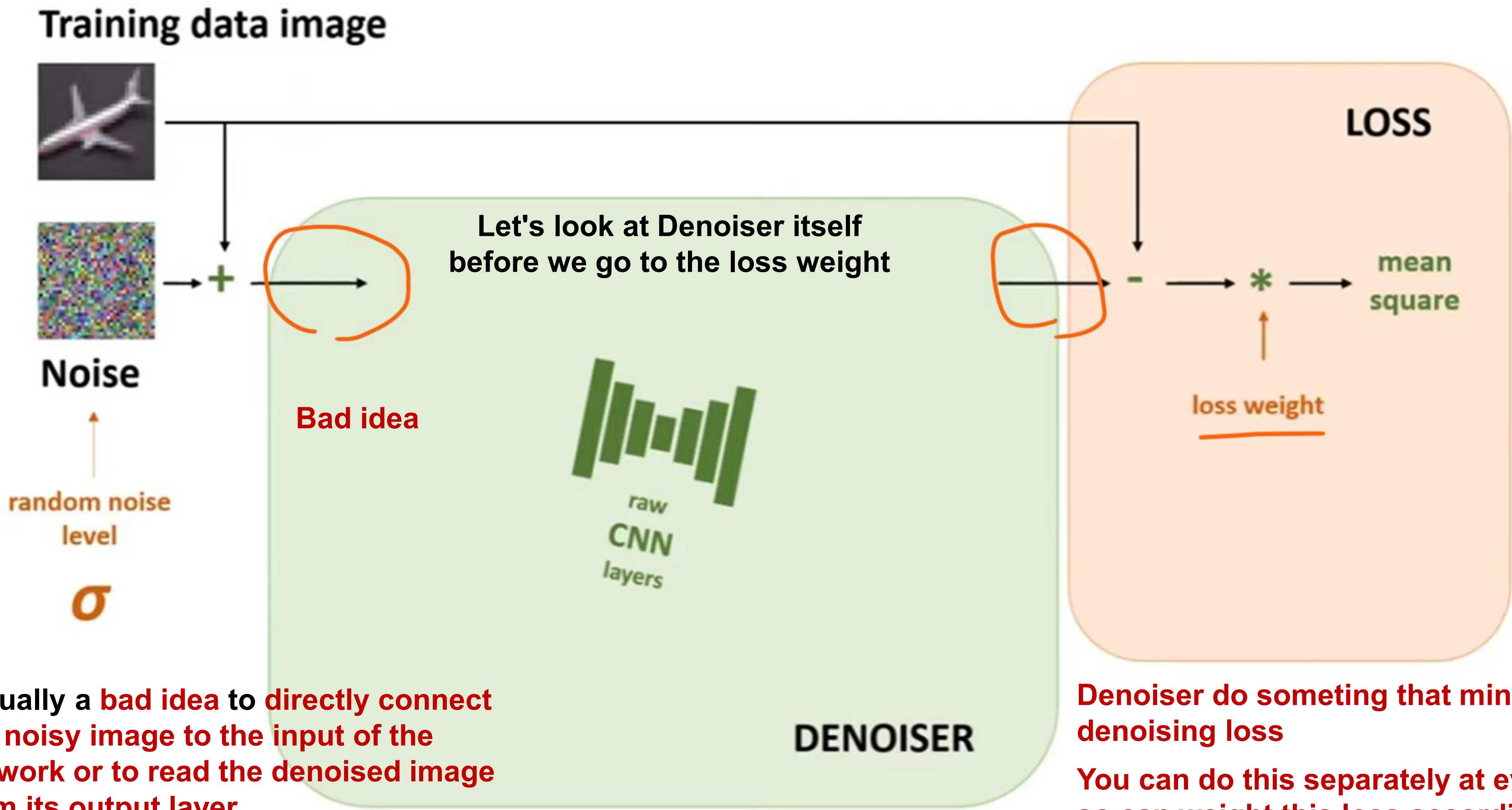
- ODE step uses the *score function*
- … which can be computed using a denoiser
- … which we approximate with a neural network

ODE Role Give us the step direction by the score function, evaluated using a Denoiser which can be approximated using neural network

The role of neural networks tells where to go in a single step or what direction you need to go to

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.



Actually a bad idea to directly connect the noisy image to the input of the network or to read the denoised image from its output layer.

Denoiser do someting that minimizes the L2 denoising loss

You can do this separately at every noise level, so can weight this loss according to the noise level

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Preconditioning

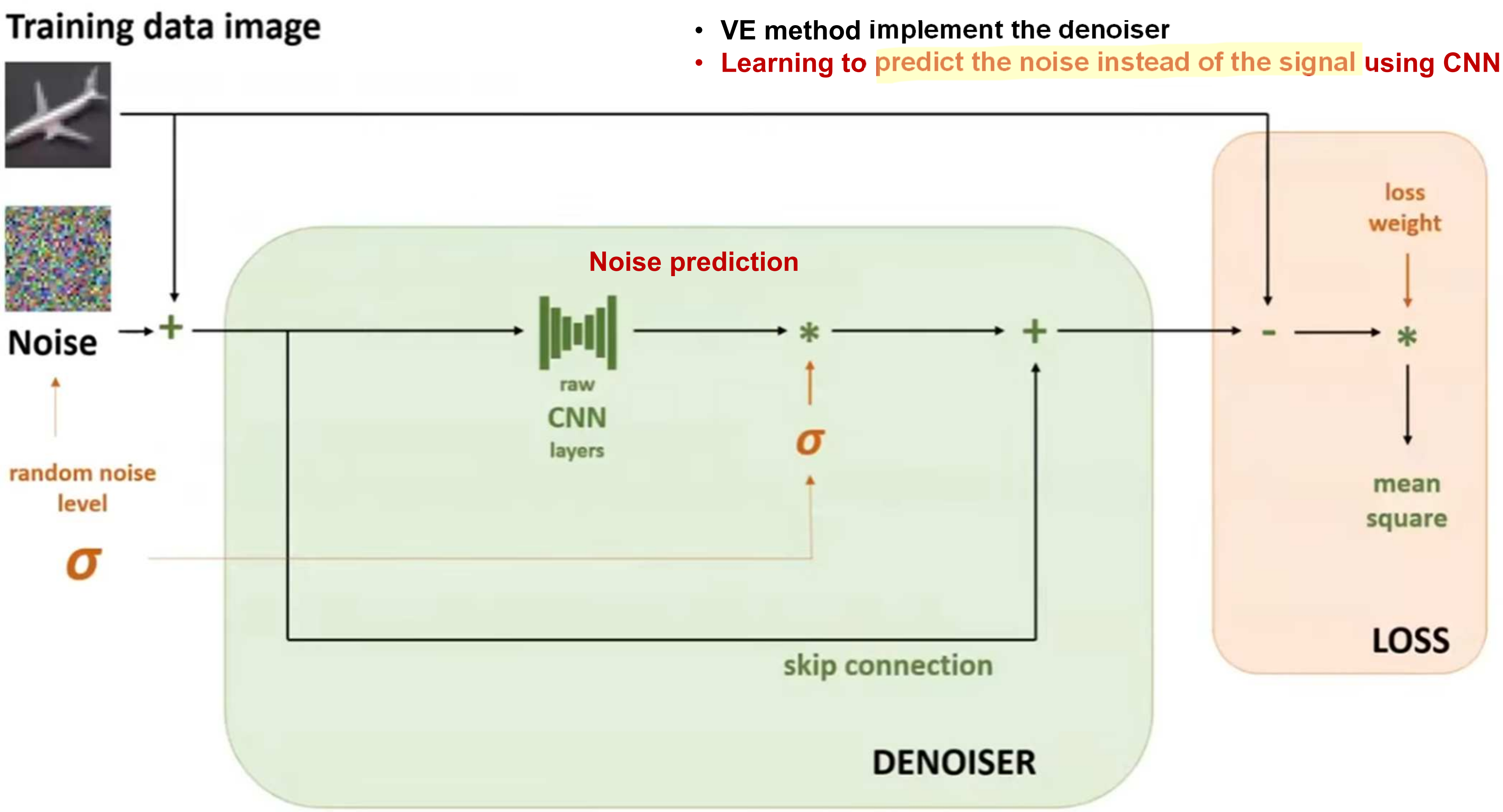
- To make things easy for the CNN:
  - (A) Always feed unit stdev inputs to networks
  - (B) .. and train with unit stdev targets
- Networks make errors. We should (C) minimize network's contribution to output of the denoiser
- Our noise levels vary wildly, so this is critical!

- **Want to wrap it between some kind of a signal management layers to manage those signal scales of both the input and the output to standardize them.**
- **Often recycle from the input because if the input image is almost noise free, then we don't really need to denoise much.**
- **Should copy just what we know and only fix the remainder we're going to come to that soon.**

Youtube Presentaiton

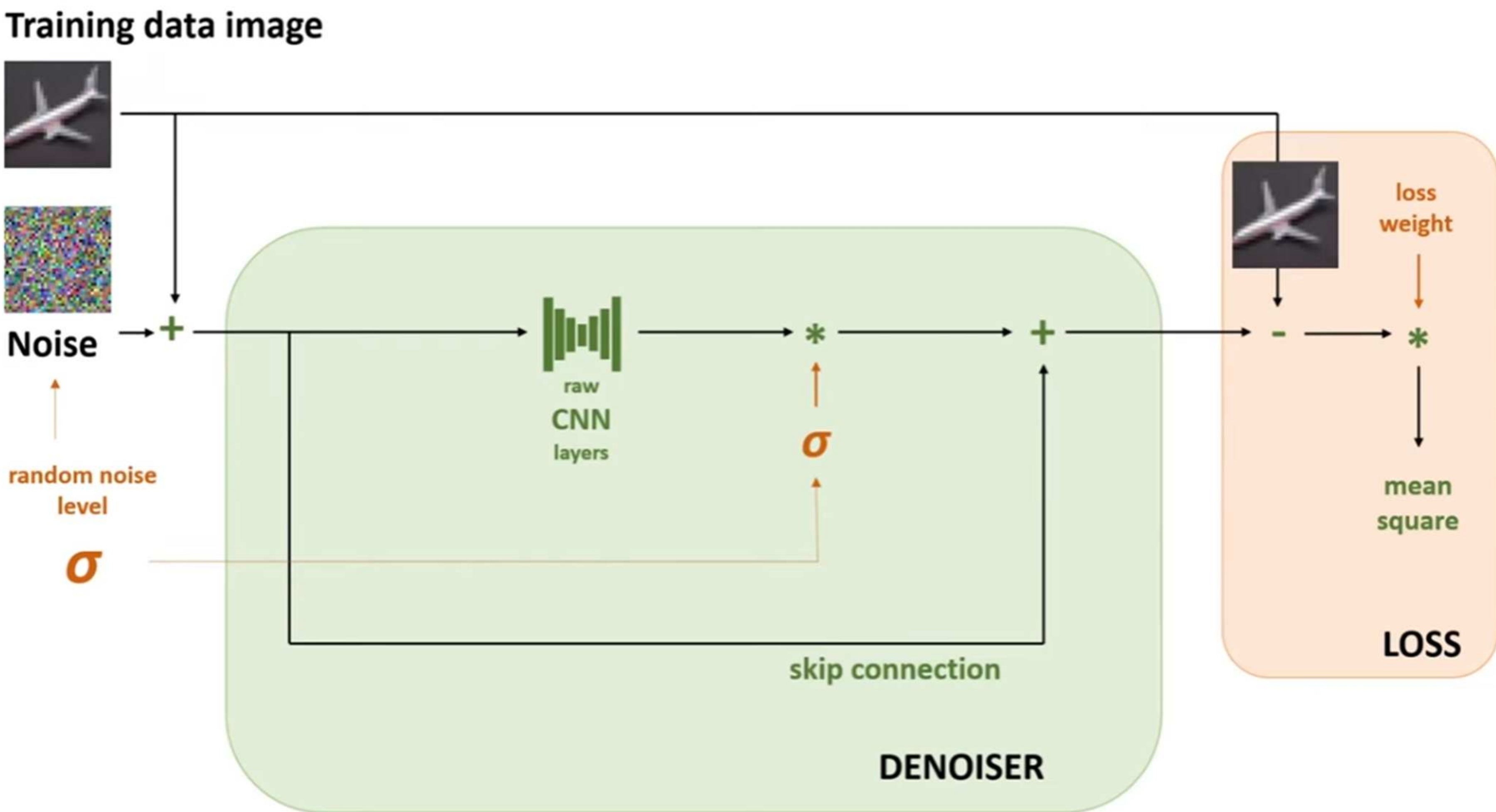
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.

- VE method implement the denoiser
- Learning to predict the noise instead of the signal using CNN layers



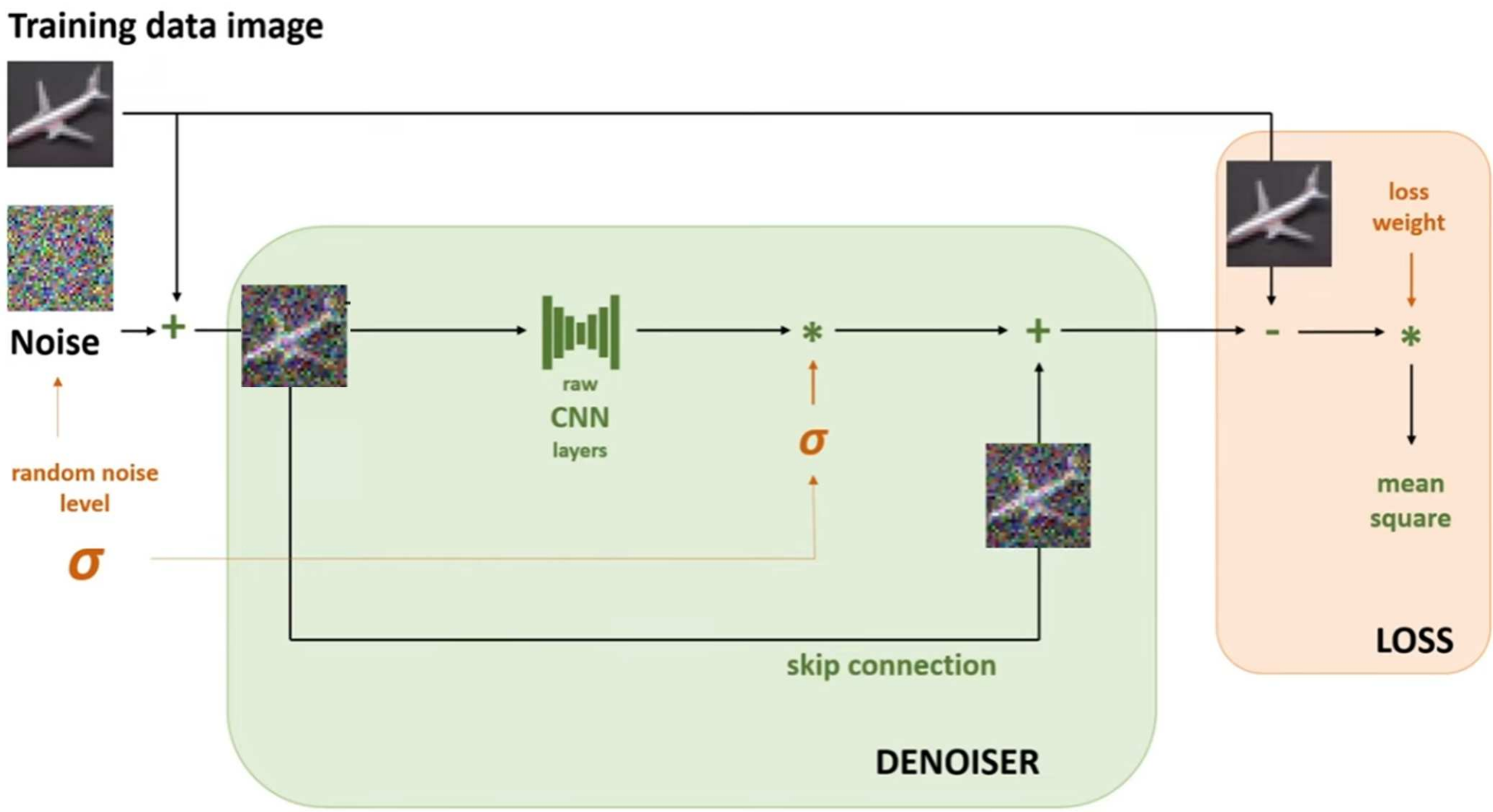
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.



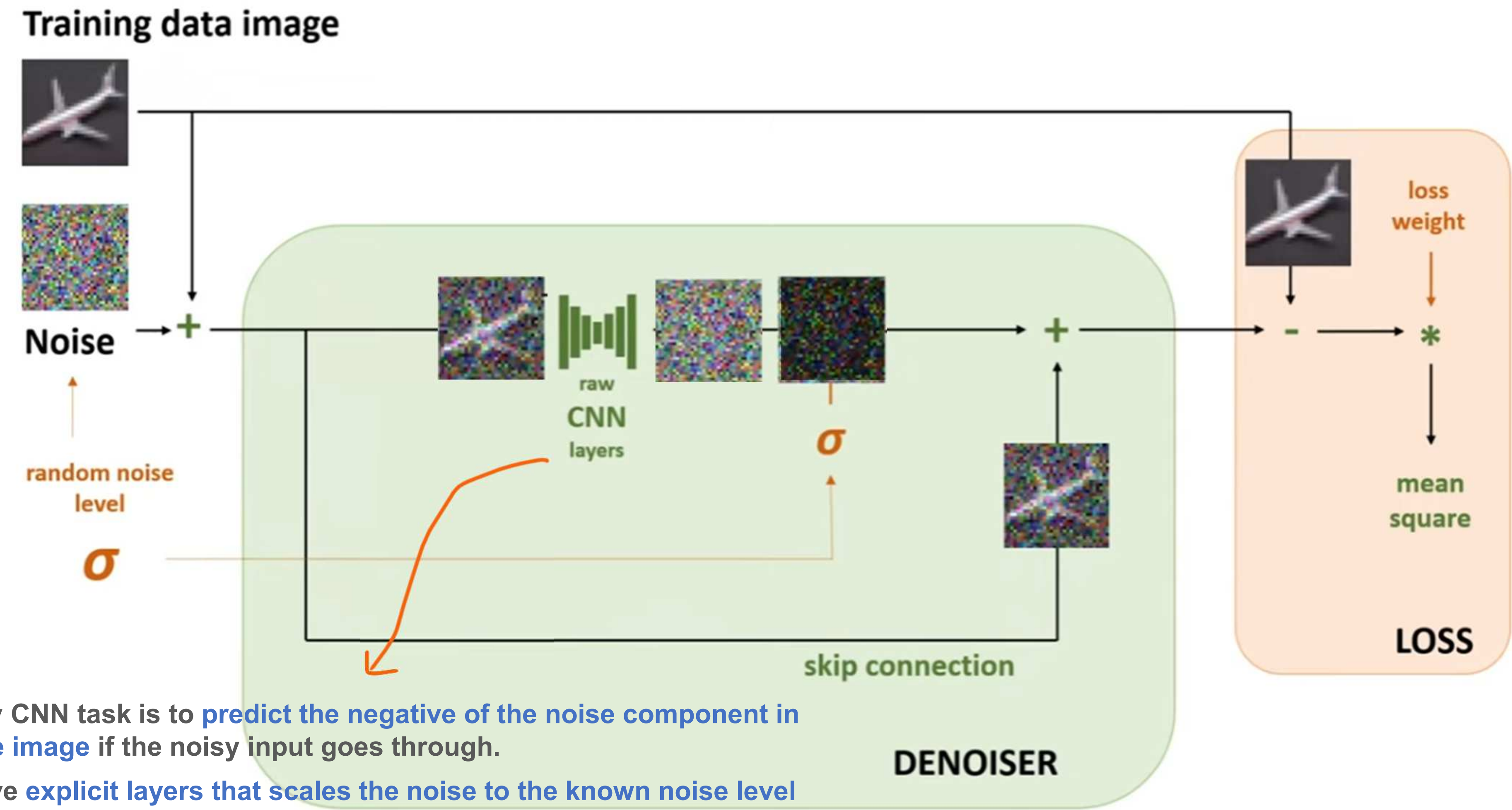
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

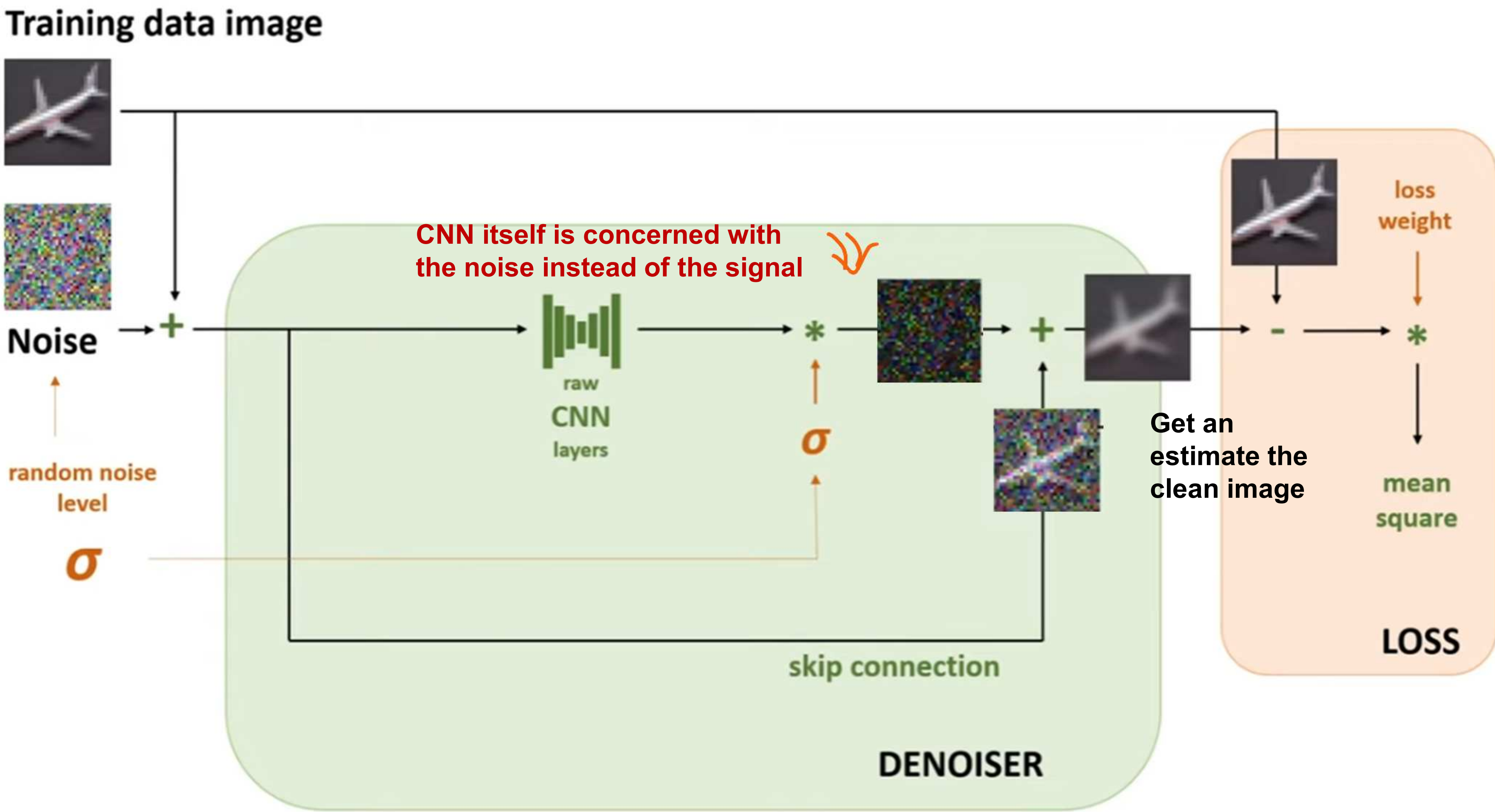


Implicitly CNN task is to predict the negative of the noise component in the noise image if the noisy input goes through.

They have explicit layers that scales the noise to the known noise level

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

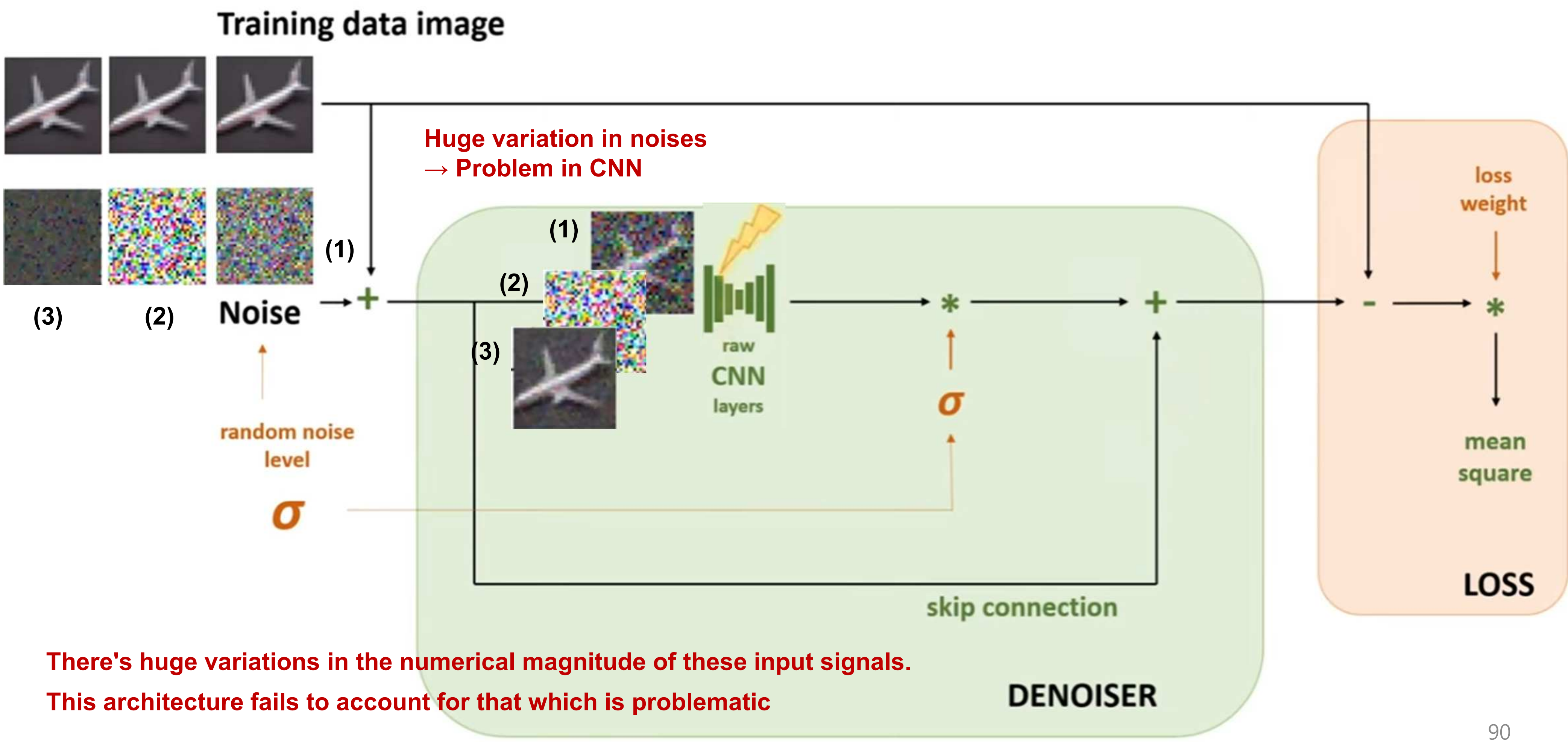


CNN itself is concerned with the noise instead of the signal

Get an estimate the clean image

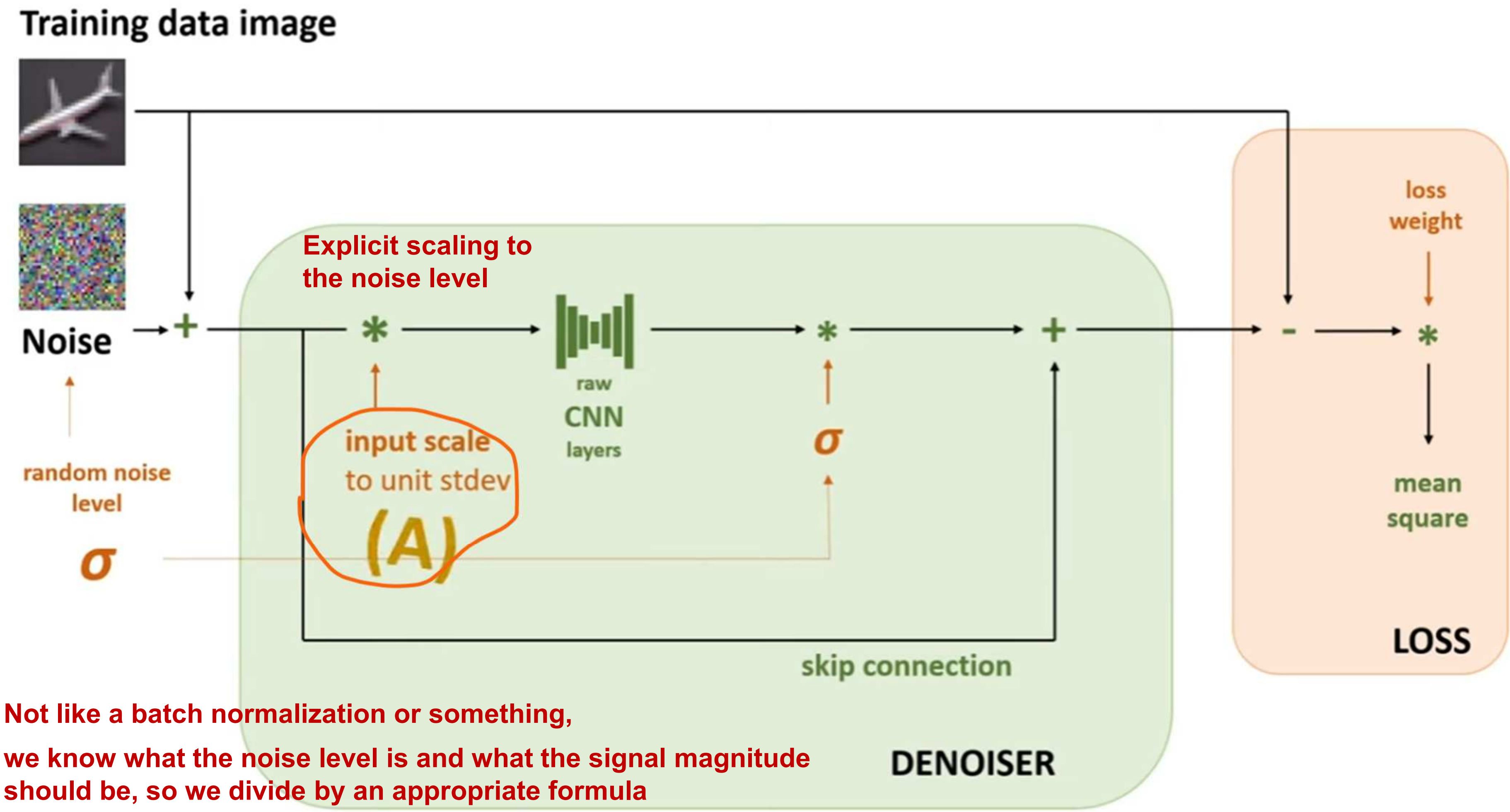
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

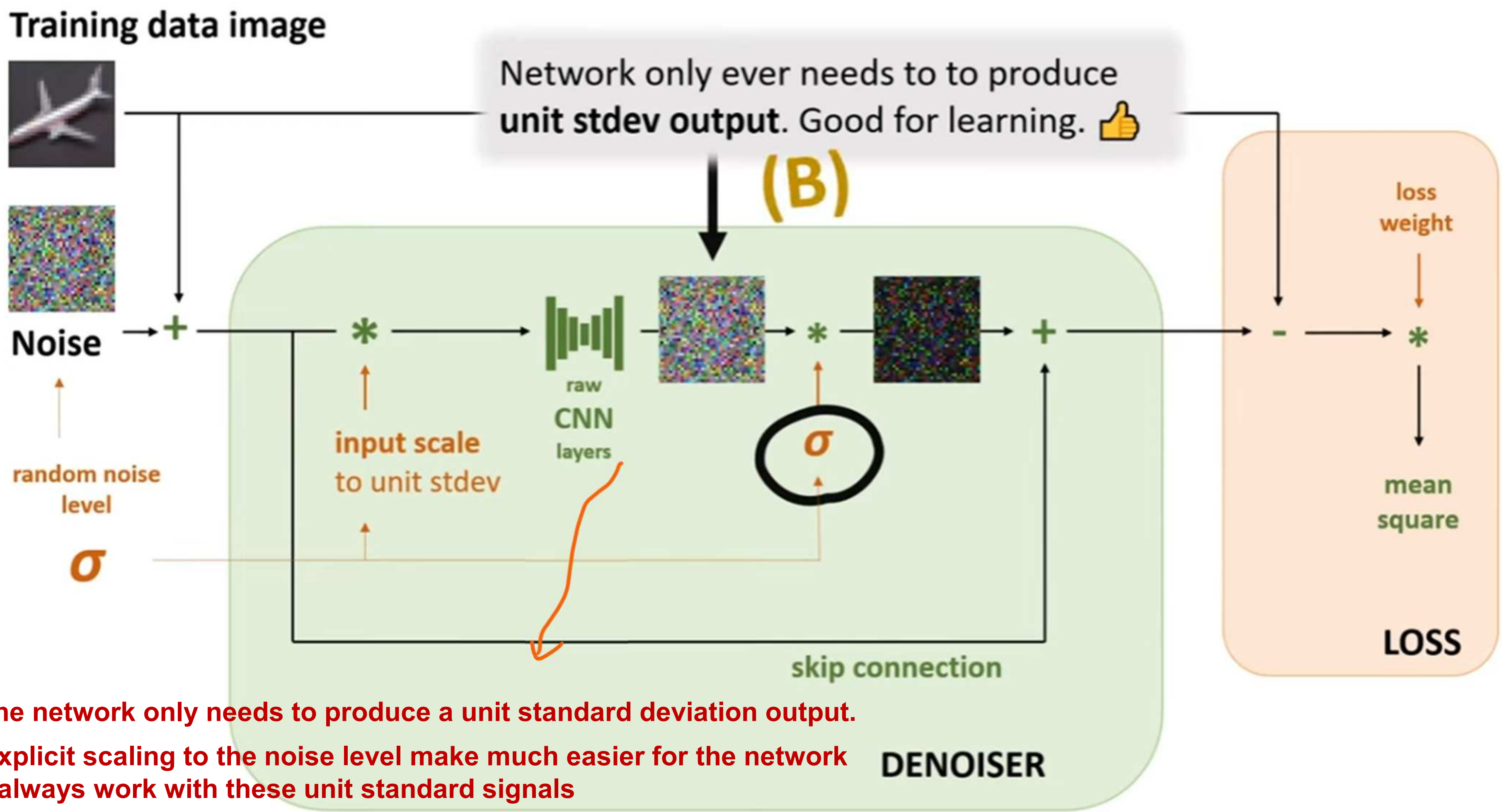


Not like a batch normalization or something,

we know what the noise level is and what the signal magnitude should be, so we divide by an appropriate formula

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models]," NeurIPS 2022.



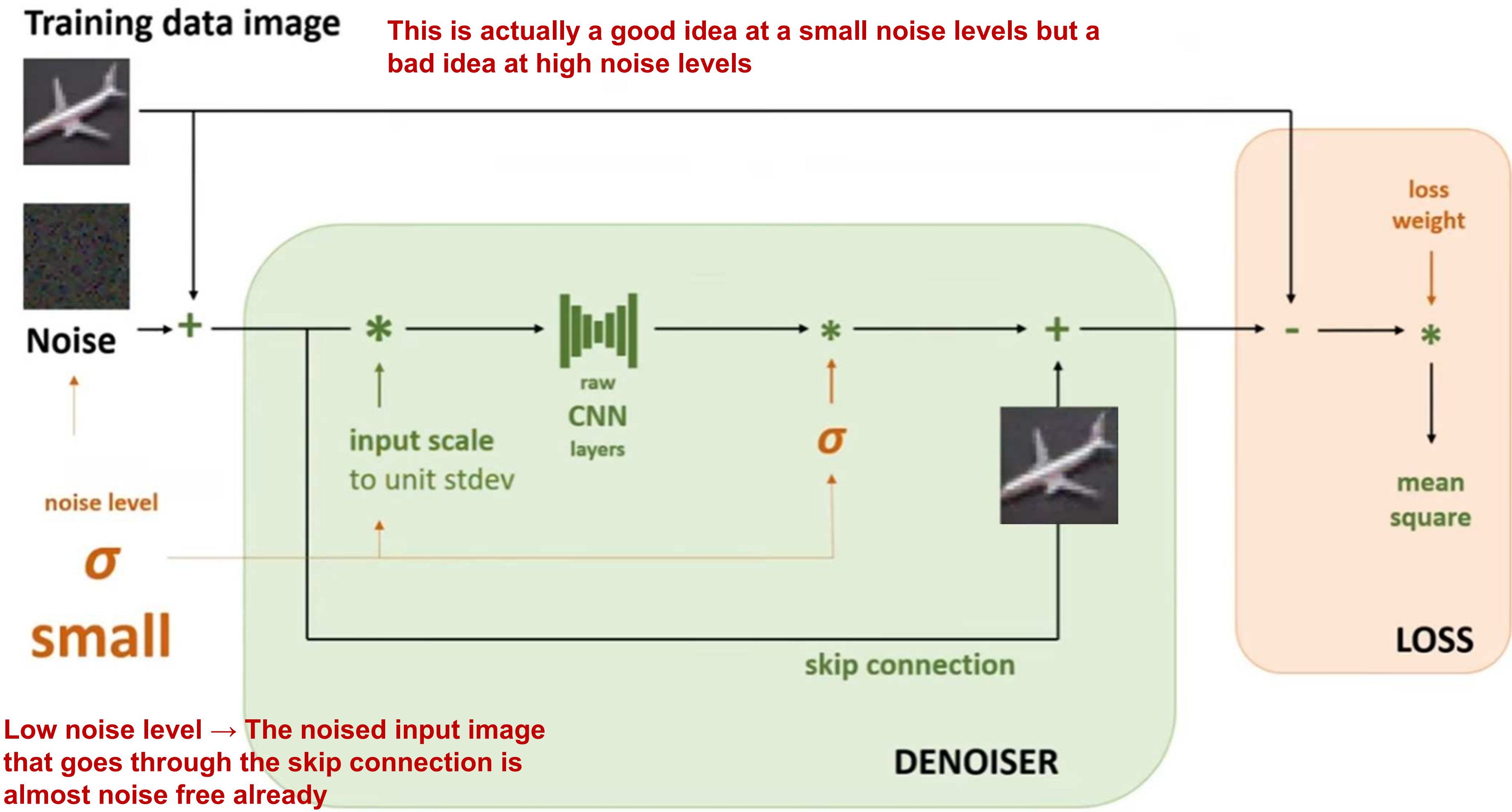
Now the network only needs to produce a unit standard deviation output.

This explicit scaling to the noise level make much easier for the network it can always work with these unit standard signals

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.

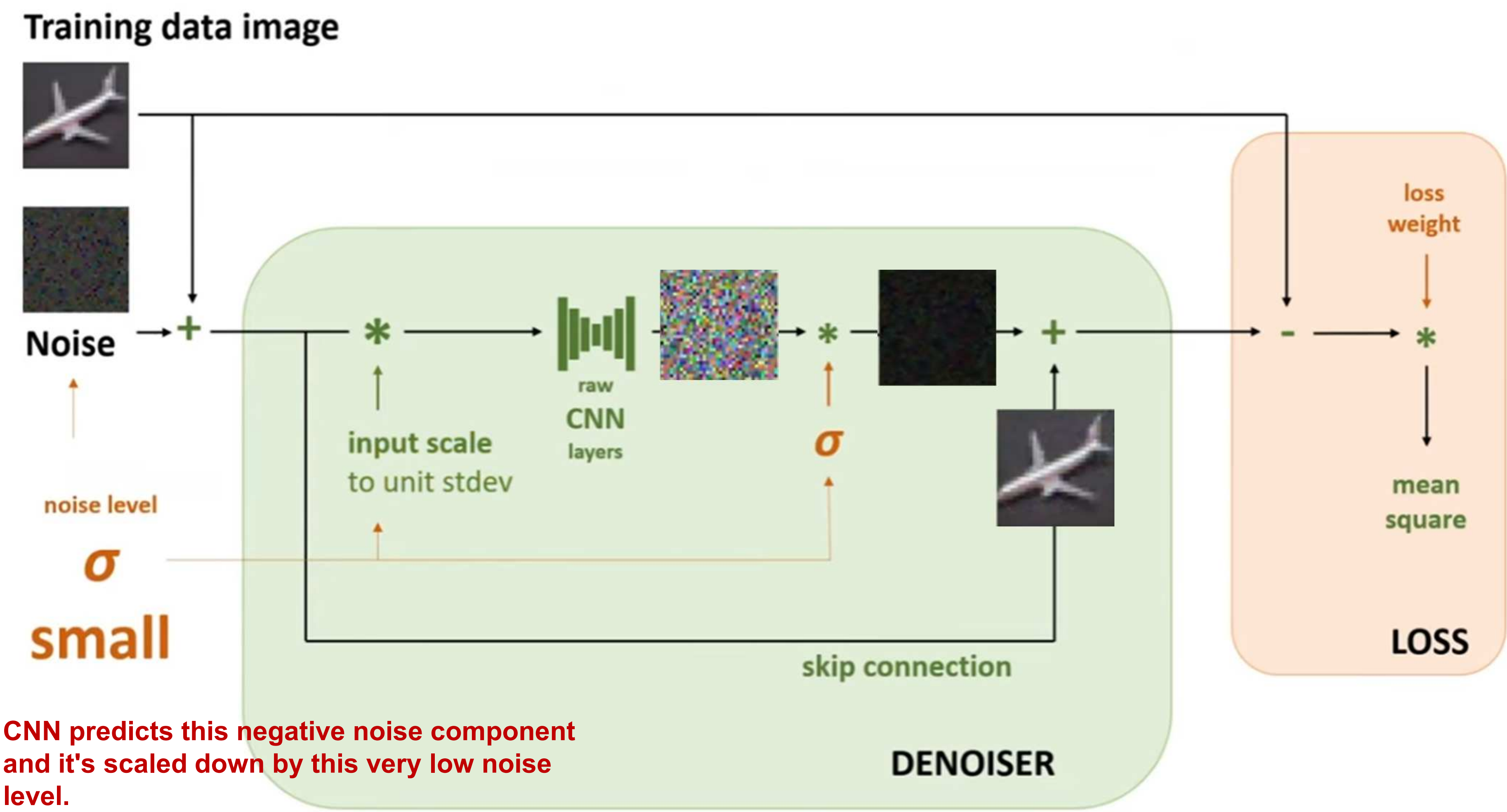
This is actually a good idea at a small noise levels but a bad idea at high noise levels



Low noise level → The noised input image that goes through the skip connection is almost noise free already

Youtube Presentaiton

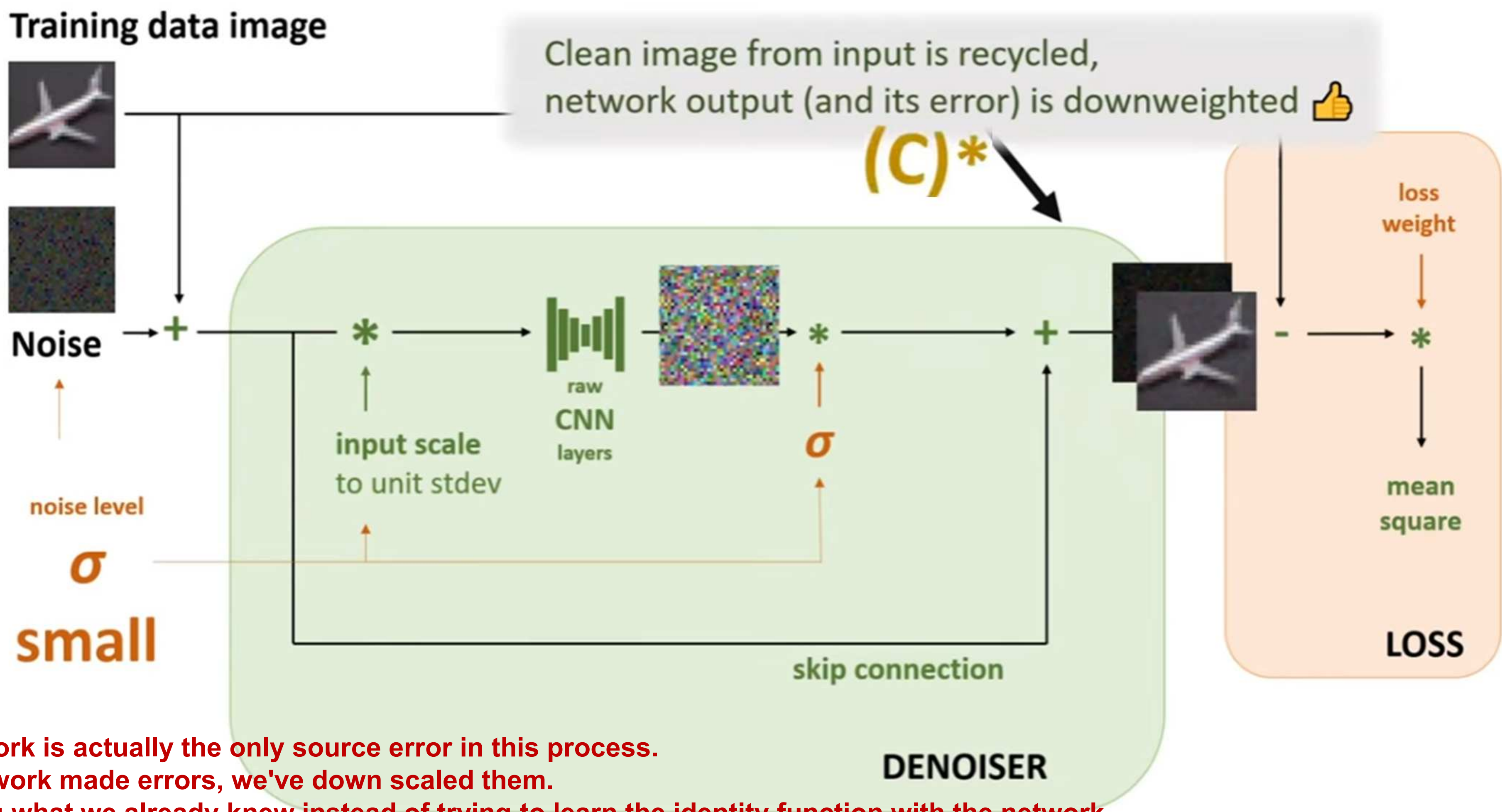
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



CNN predicts this negative noise component and it's scaled down by this very low noise level.

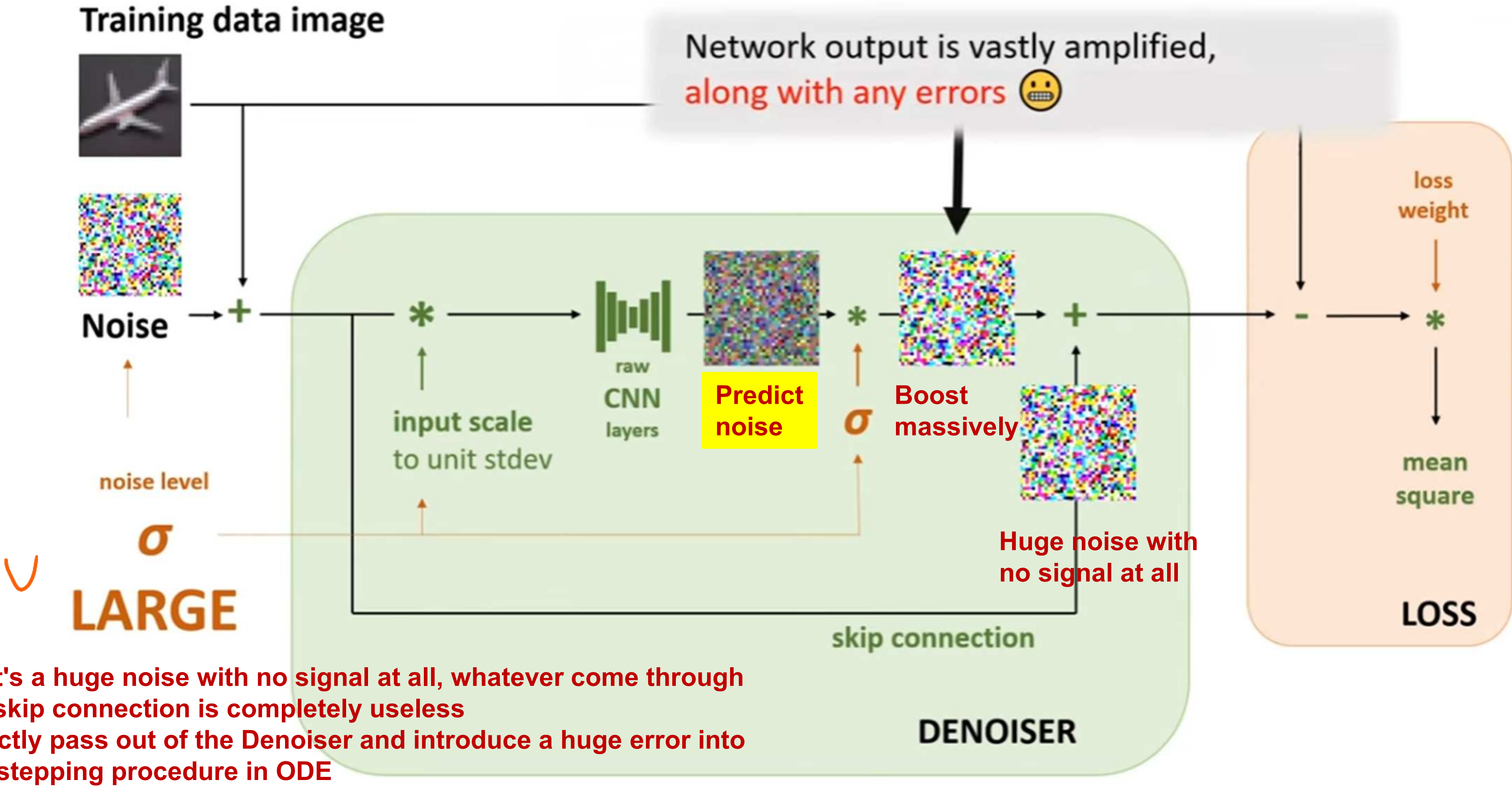
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.



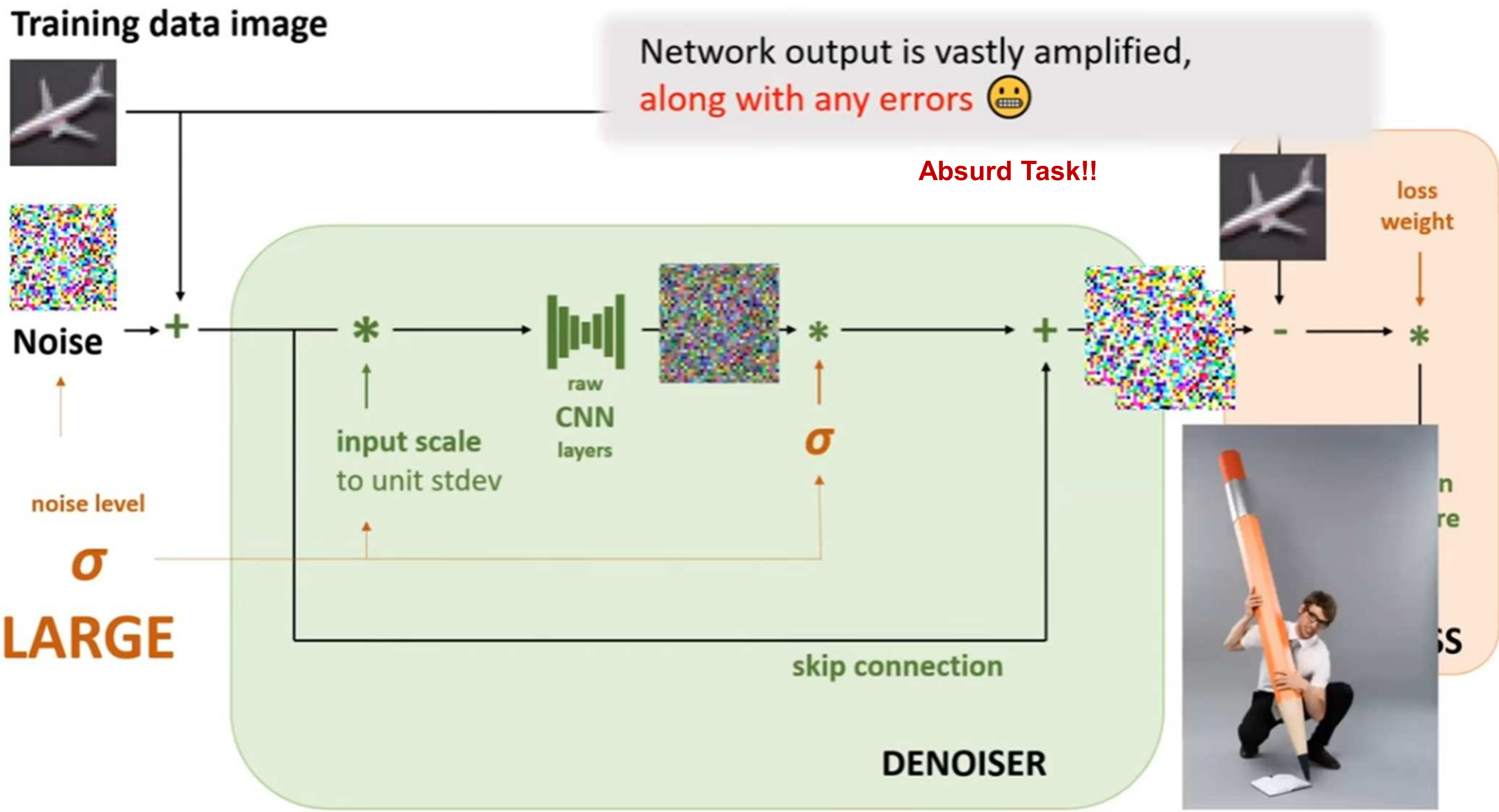
The network is actually the only source error in this process.
If the network made errors, we've down scaled them.
Recycling what we already knew instead of trying to learn the identity function with the network

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models]," NeurIPS 2022.



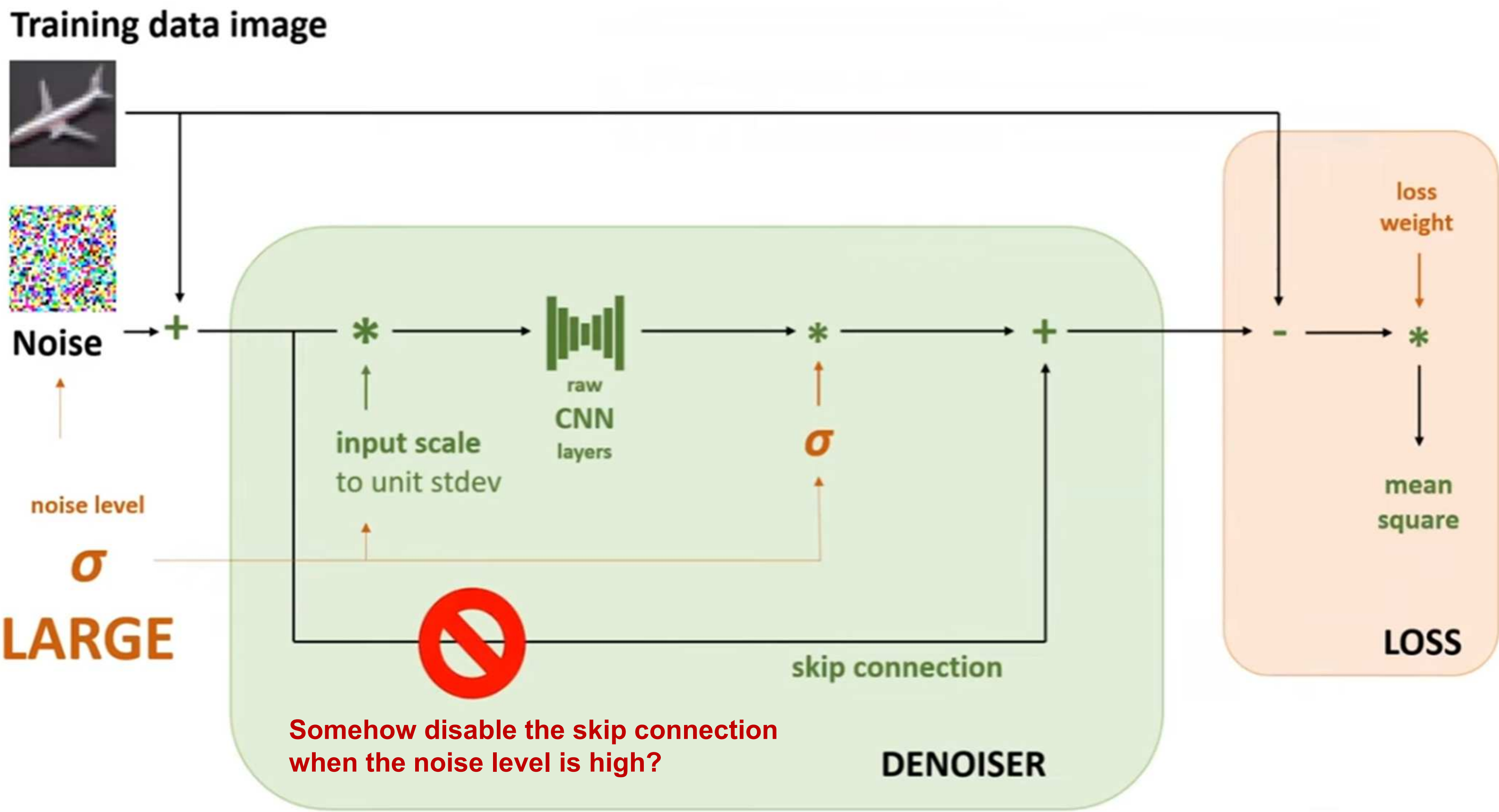
As it's a huge noise with no signal at all, whatever come through the skip connection is completely useless
Directly pass out of the Denoiser and introduce a huge error into our stepping procedure in ODE

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.



Youtube Presentaiton

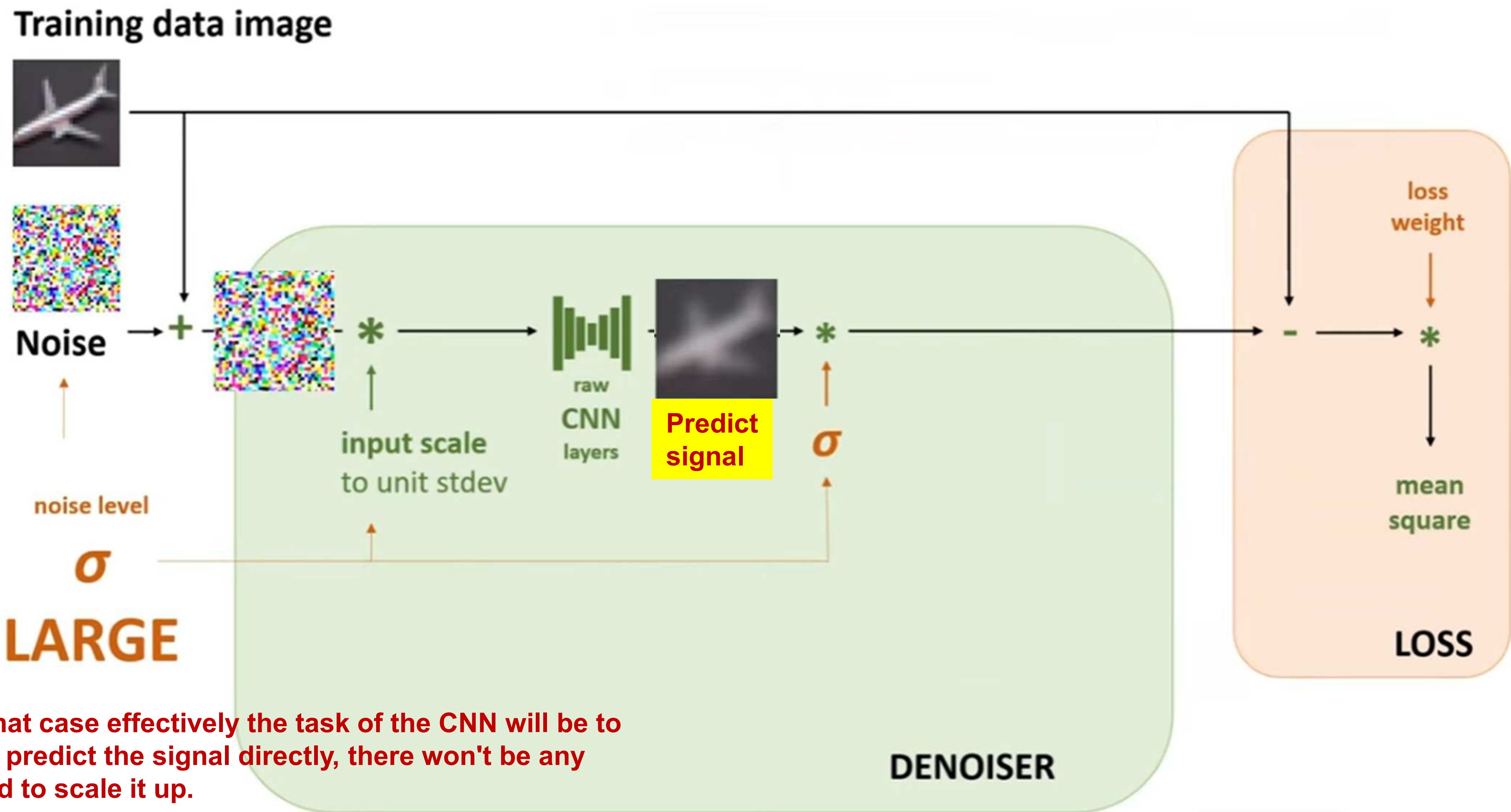
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.



Somehow disable the skip connection when the noise level is high?

Youtube Presentaiton

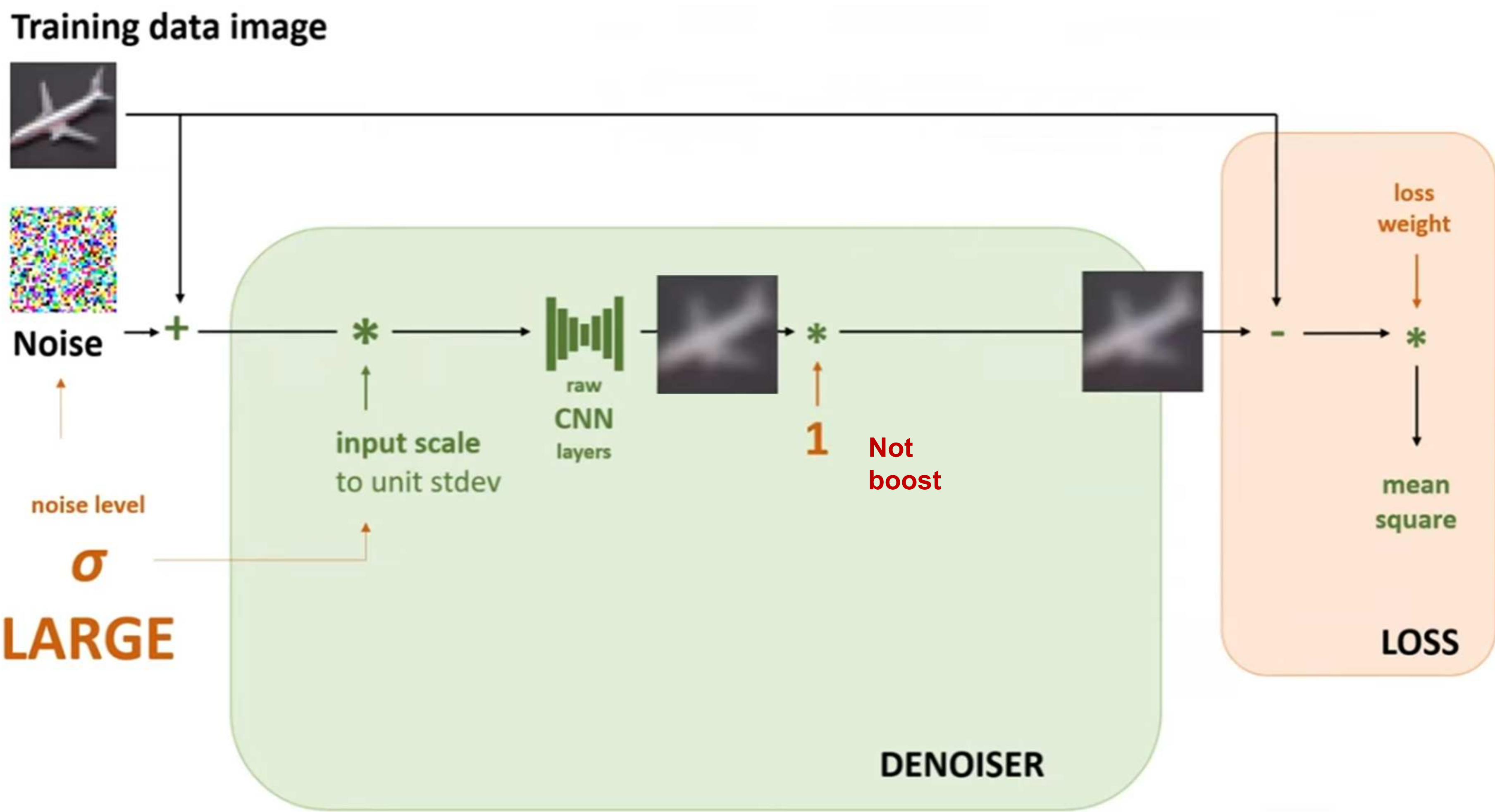
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.



In that case effectively the task of the CNN will be to just predict the signal directly, there won't be any need to scale it up.

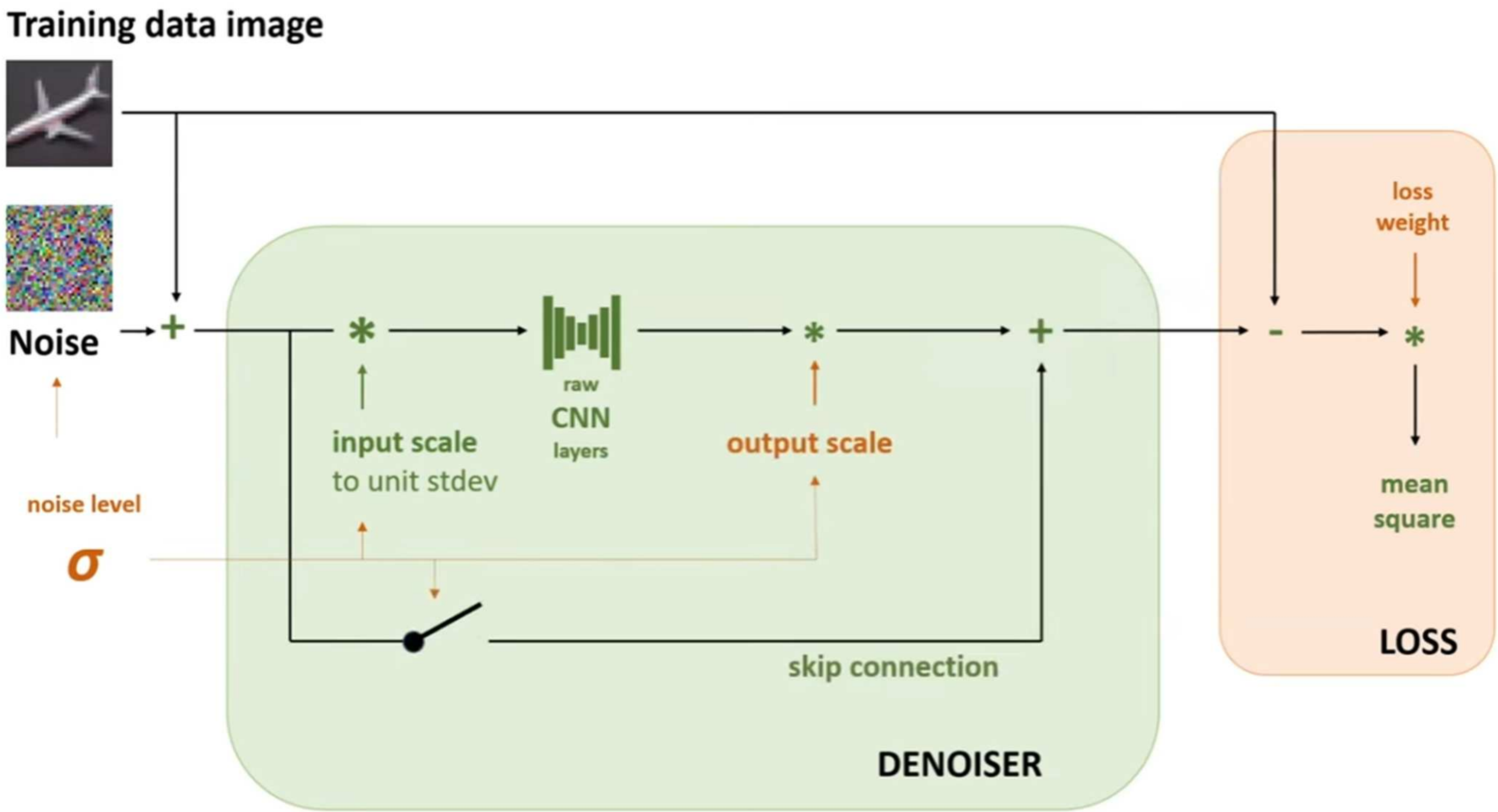
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.



Youtube Presentaiton

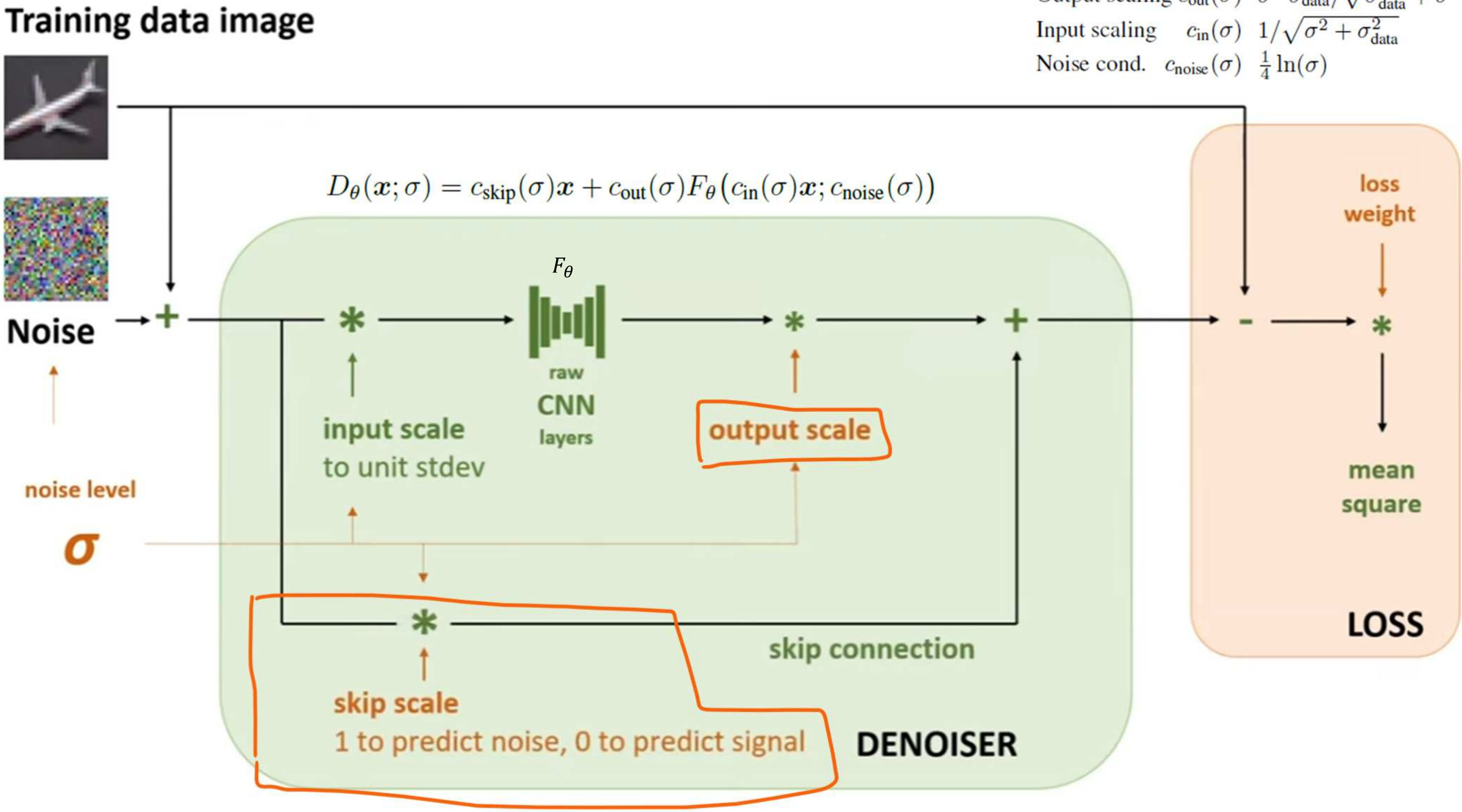
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based**

Architecture of F_θ		(any)
Skip scaling	$c_{\text{skip}}(\sigma)$	$\sigma_{\text{data}}^2 / (\sigma^2 + \sigma_{\text{data}}^2)$
Output scaling	$c_{\text{out}}(\sigma)$	$\sigma \cdot \sigma_{\text{data}} / \sqrt{\sigma_{\text{data}}^2 + \sigma^2}$
Input scaling	$c_{\text{in}}(\sigma)$	$1/\sqrt{\sigma^2 + \sigma_{\text{data}}^2}$
Noise cond.	$c_{\text{noise}}(\sigma)$	$\frac{1}{4}\ln(\sigma)$

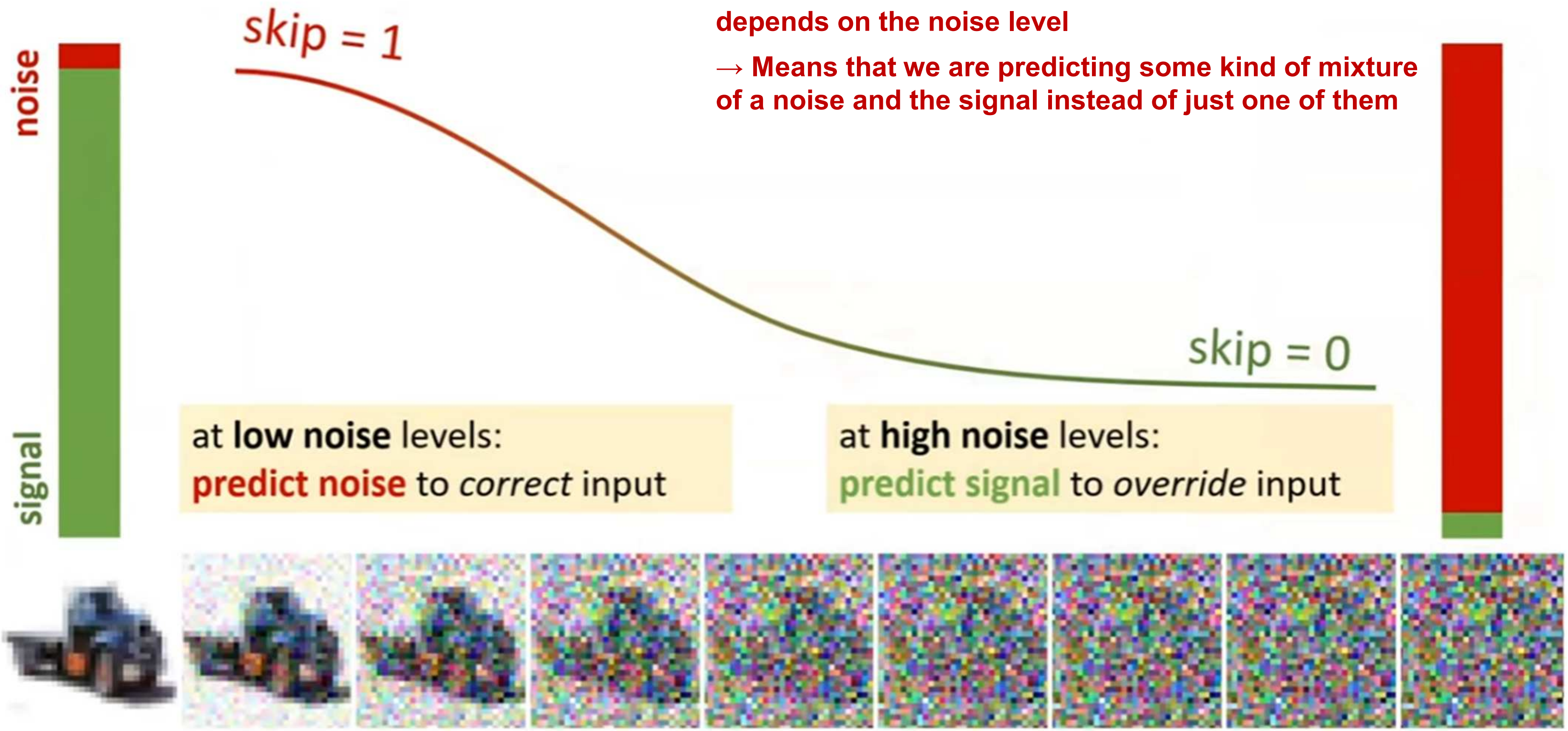


Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

A continuous value between zero and one that depends on the noise level

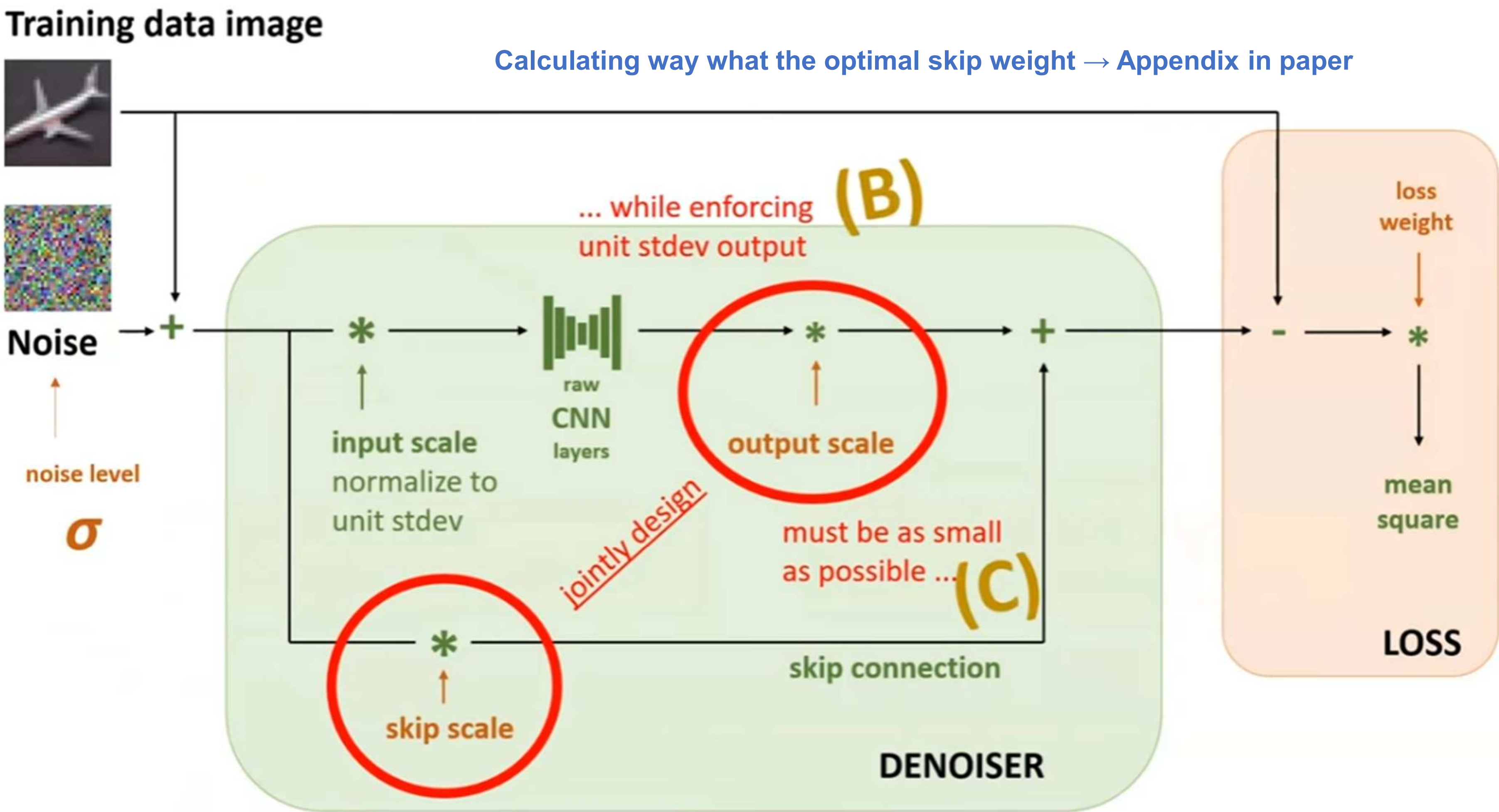
→ Means that we are predicting some kind of mixture of a noise and the signal instead of just one of them



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Calculating way what the optimal skip weight → Appendix in paper



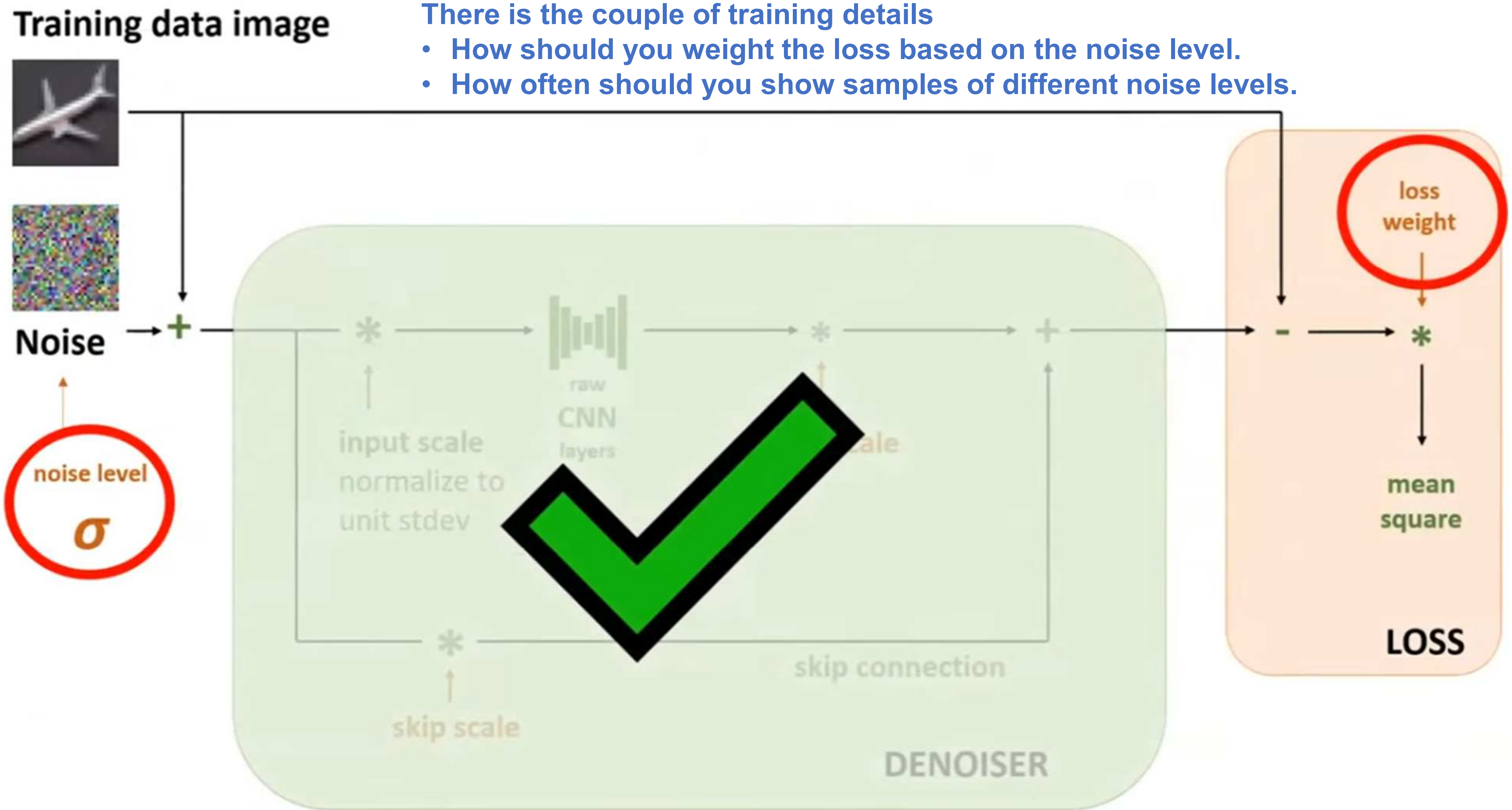
EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.

		VP	VE	iDDPM + DDIM	Ours
Sampling					
ODE solver		Euler	Euler	Euler	2nd order Heun
Time steps	$t_{i<N}$	$1+\frac{i}{N-1}(\epsilon_s-1)$	$\sigma_{\max}^2\left(\sigma_{\min}^2/\sigma_{\max}^2\right)^{\frac{i}{N-1}}$	$u_{\lfloor j_0+\frac{M-1-j_0}{N-1}i+\frac{1}{2}\rfloor}$, where $u_M=0$, $u_{j-1}=\sqrt{\frac{u_j^2+1}{\max(\bar{\alpha}_{j-1}/\bar{\alpha}_j,C_1)}-1}$	$\left(\sigma_{\max}{}^{\frac{1}{\rho}}+\frac{i}{N-1}\left(\sigma_{\min}{}^{\frac{1}{\rho}}-\sigma_{\max}{}^{\frac{1}{\rho}}\right)\right)^{\rho}$
Schedule	$\sigma(t)$	$\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}-1}$	$\sqrt{t}$	t	t
Scaling	$s(t)$	$1/\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}}$	1	1	1
Network and preconditioning					
Skip scaling	$c_{\text{skip}}(\sigma)$	1	1	1	$\sigma_{\text{data}}^2/\left(\sigma^2+\sigma_{\text{data}}^2\right)$
Output scaling	$c_{\text{out}}(\sigma)$	$-\sigma$	σ	$-\sigma$	$\sigma\cdot\sigma_{\text{data}}/\sqrt{\sigma_{\text{data}}^2+\sigma^2}$
Input scaling	$c_{\text{in}}(\sigma)$	$1/\sqrt{\sigma^2+1}$	1	$1/\sqrt{\sigma^2+1}$	$1/\sqrt{\sigma^2+\sigma_{\text{data}}^2}$
? Noise cond.	$c_{\text{noise}}(\sigma)$	$(M-1)\,\sigma^{-1}(\sigma)$	$\ln(\frac{1}{2}\sigma)$	$M-1-\arg\min_j \lvert u_j-\sigma\rvert$	$\frac{1}{4}\ln(\sigma)$
Training					
Noise distribution					
Loss weighting	$\lambda(\sigma)$				

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.

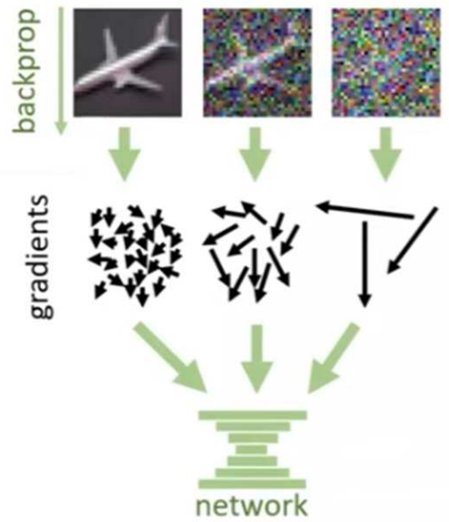
There is the couple of training details
- How should you weight the loss based on the noise level.
- How often should you show samples of different noise levels.



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "[Elucidating the Design Space of Diffusion-Based Generative Models](#)," NeurIPS 2022.

Loss weighting and noise level distribution



1. Baseline: frequent small updates on some noise levels, infrequent large updates on others. Unhealthy training dynamics.

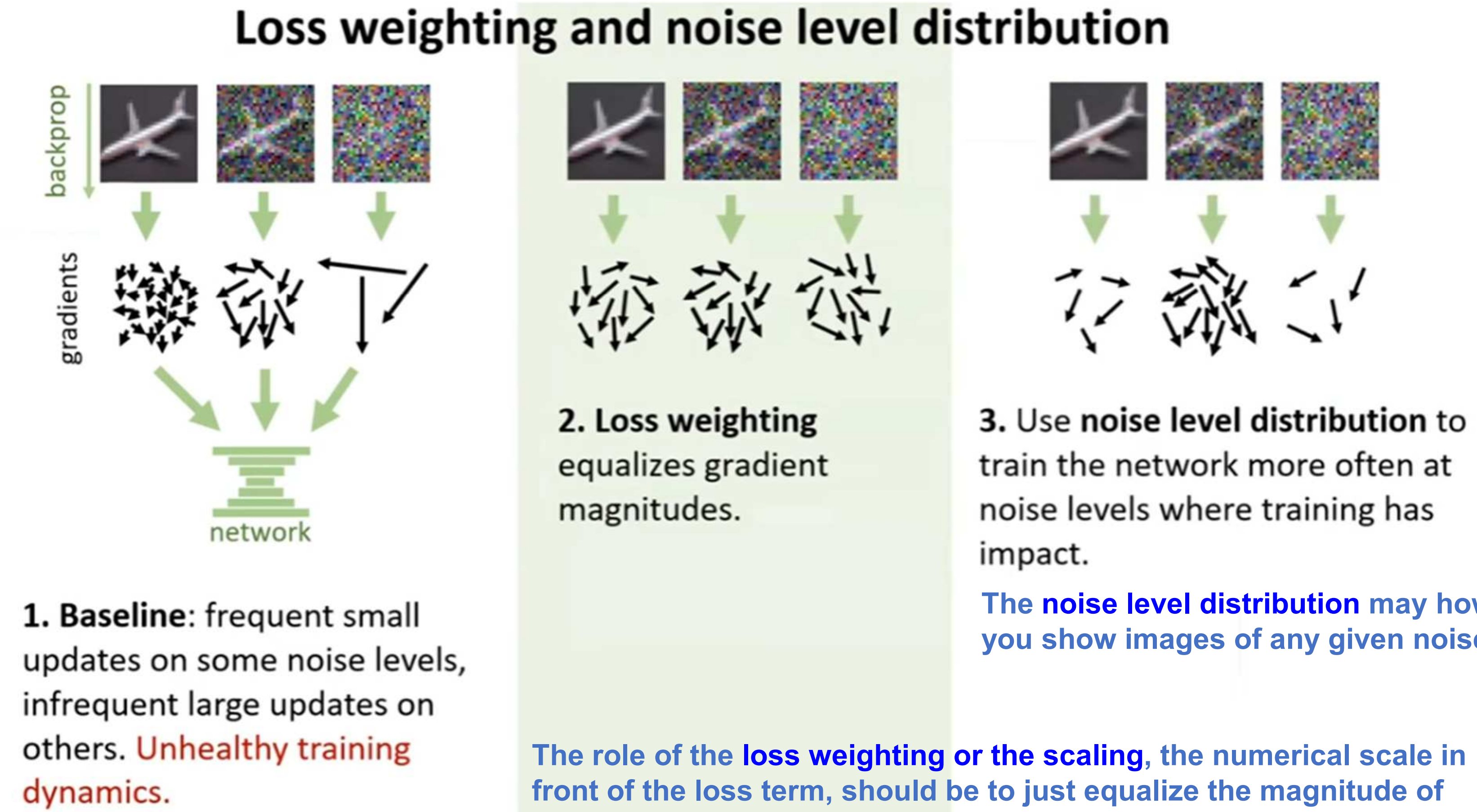
General problem

- **Might have a highly lopsided distribution of like gradient feedback.**
- **If not careful on most iteration, provide the weights gently to one direction or the other and have the massive gradient smash on the weights every few iterations.**

That's probably very bad for your training dynamics.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.



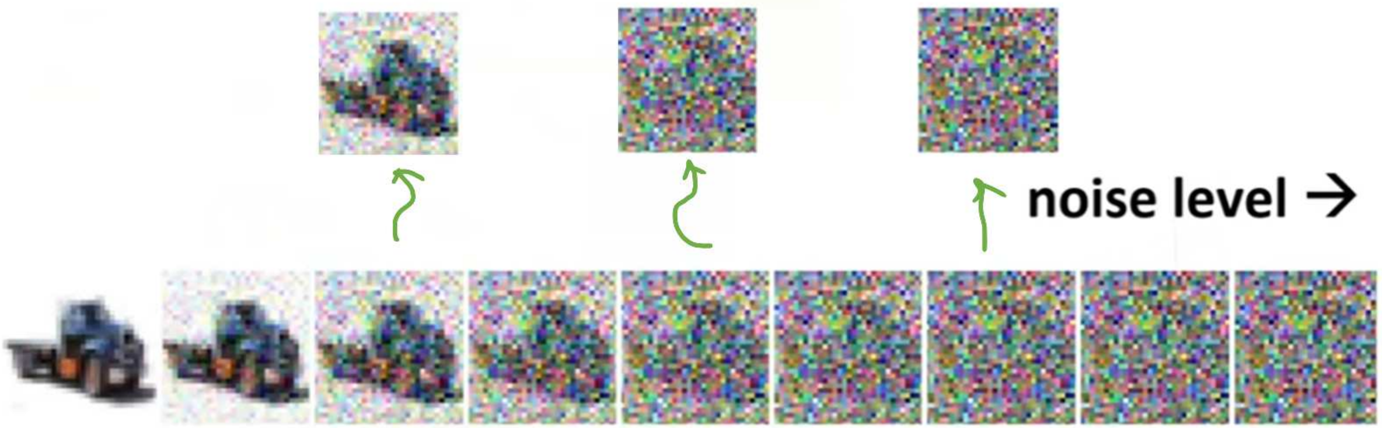
The **noise level distribution** may how often you show images of any given noise level.

The role of the **loss weighting or the scaling**, the numerical scale in front of the loss term, should be to just equalize the magnitude of the loss or equivalently equalize the magnitude of the gradient feedback it gives.

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

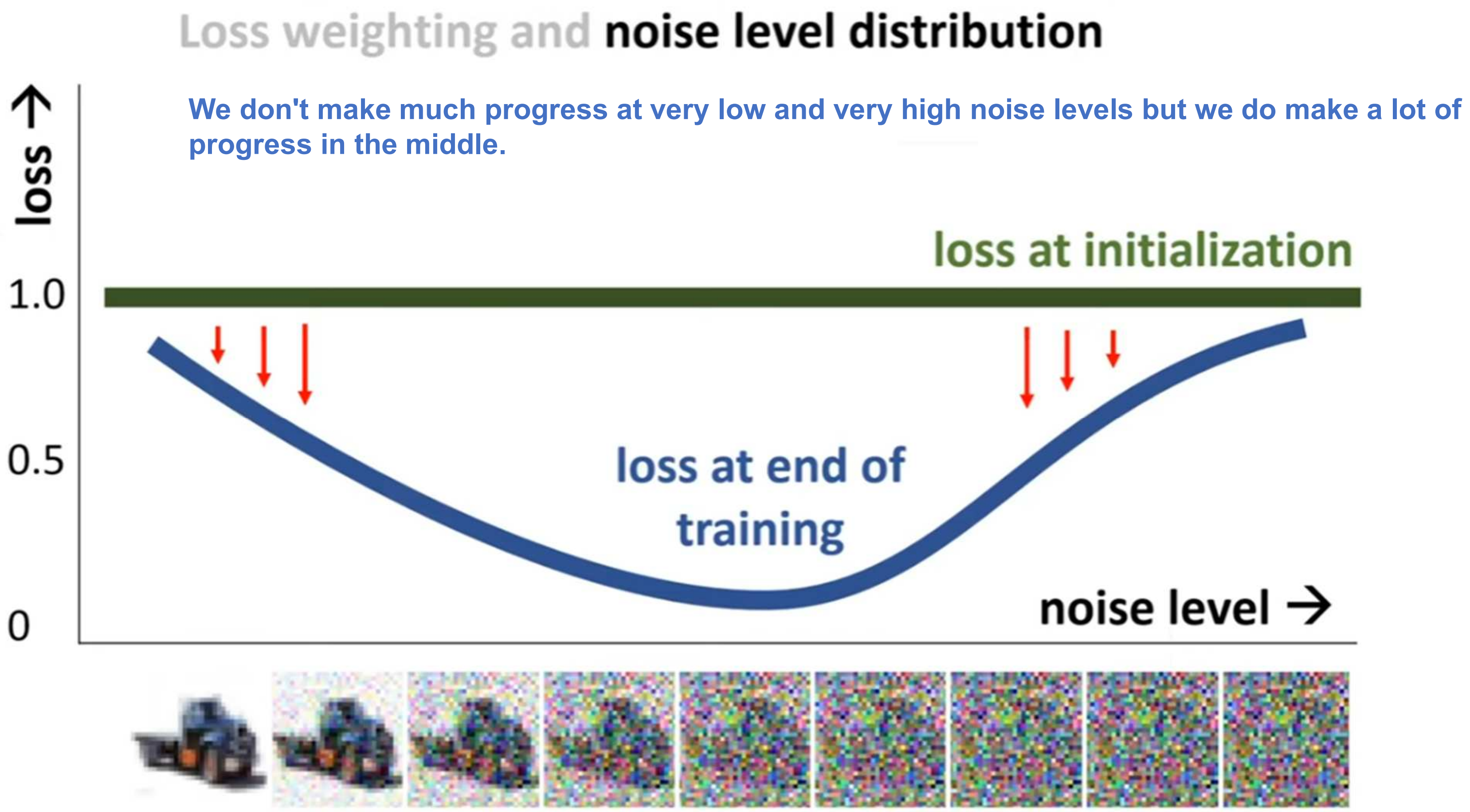
Loss weighting and noise level distribution

The role of noise level distribution is to direct your training efforts to the levels where you know it's relevant where you know you can make an impact.



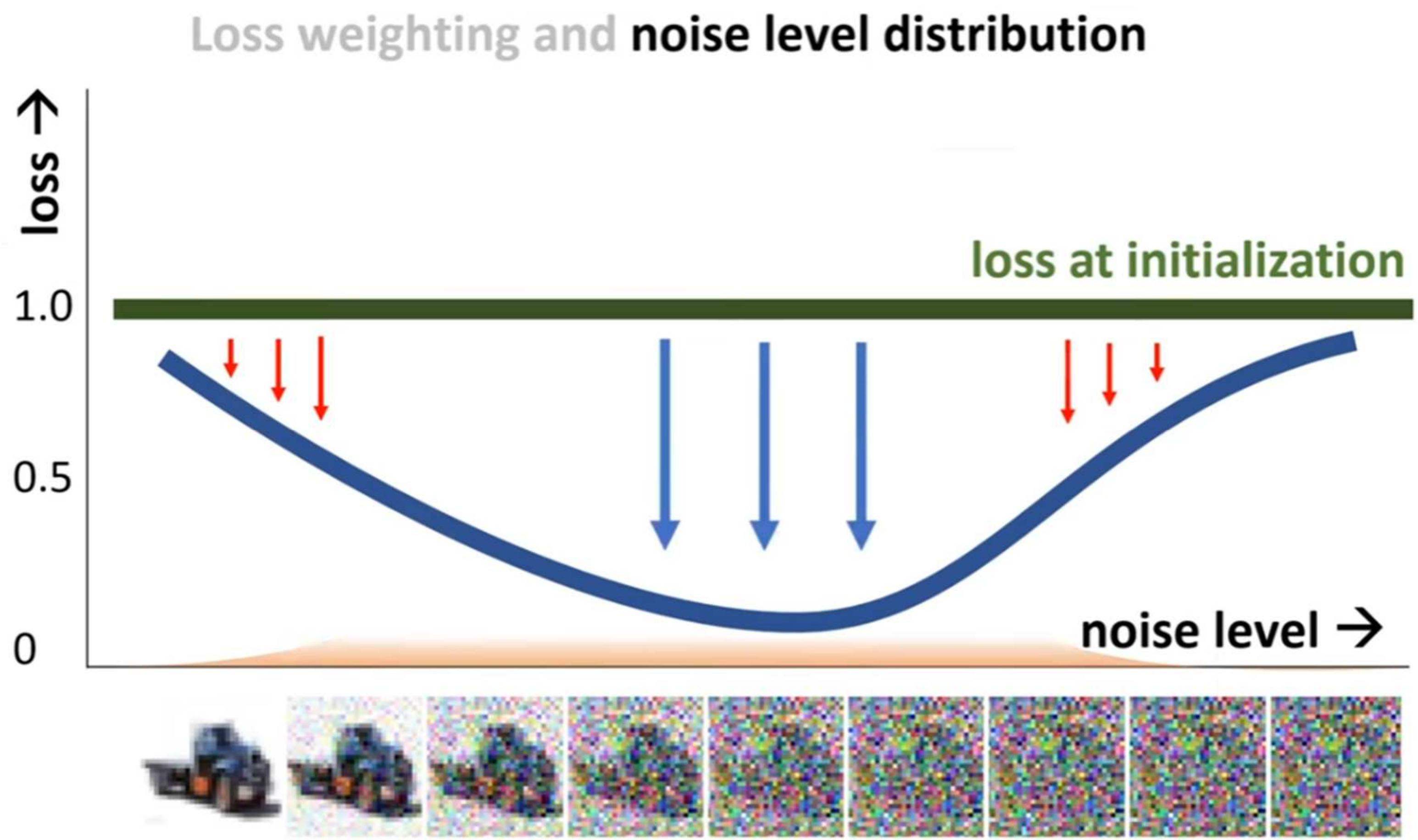
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

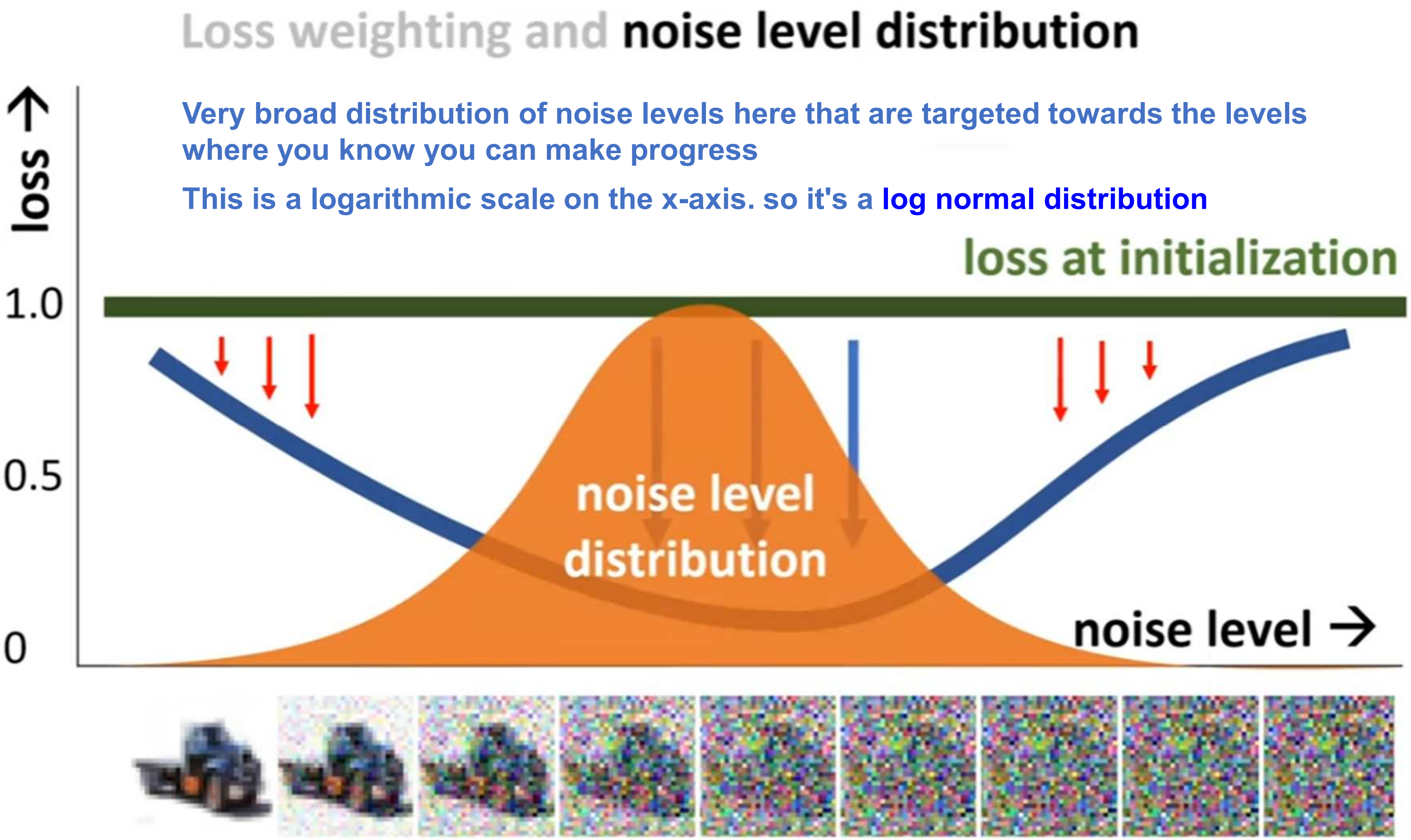


Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Very broad distribution of noise levels here that are targeted towards the levels where you know you can make progress

This is a logarithmic scale on the x-axis. so it's a **log normal distribution**



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

		VP	VE	iDDPM + DDIM	Ours
Sampling					
ODE solver		Euler	Euler	Euler	2nd order Heun
Time steps	$t_{i<N}$	$1+\frac{i}{N-1}(\epsilon_s-1)$	$\sigma_{\max}^2\left(\sigma_{\min}^2/\sigma_{\max}^2\right)^{\frac{i}{N-1}}$	$u_{\lfloor j_0+\frac{M-1-j_0}{N-1}i+\frac{1}{2}\rfloor}$, where $u_M=0$, $u_{j-1}=\sqrt{\frac{u_j^2+1}{\max(\bar{\alpha}_{j-1}/\bar{\alpha}_j,C_1)}-1}$	$\left(\sigma_{\max}{}^{\frac{1}{\rho}}+\frac{i}{N-1}\left(\sigma_{\min}{}^{\frac{1}{\rho}}-\sigma_{\max}{}^{\frac{1}{\rho}}\right)\right)^{\rho}$
Schedule	$\sigma(t)$	$\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}-1}$	$\sqrt{t}$	t	t
Scaling	$s(t)$	$1/\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}}$	1	1	1
Network and preconditioning					
Skip scaling	$c_{\text{skip}}(\sigma)$	1	1	1	$\sigma_{\text{data}}^2/\left(\sigma^2+\sigma_{\text{data}}^2\right)$
Output scaling	$c_{\text{out}}(\sigma)$	$-\sigma$	σ	$-\sigma$	$\sigma\cdot\sigma_{\text{data}}/\sqrt{\sigma_{\text{data}}^2+\sigma^2}$
Input scaling	$c_{\text{in}}(\sigma)$	$1/\sqrt{\sigma^2+1}$	1	$1/\sqrt{\sigma^2+1}$	$1/\sqrt{\sigma^2+\sigma_{\text{data}}^2}$
Noise cond.	$c_{\text{noise}}(\sigma)$	$(M-1)\,\sigma^{-1}(\sigma)$	$\ln(\frac{1}{2}\sigma)$	$M-1-\arg\min_j \lvert u_j-\sigma\rvert$	$\frac{1}{4}\ln(\sigma)$
Training					
Noise distribution		$\sigma^{-1}(\sigma)\sim\mathcal{U}(\epsilon_t,1)$	$\ln(\sigma)\sim\mathcal{U}(\ln(\sigma_{\min}),\ln(\sigma_{\max}))$	$\sigma=u_j,\ j\sim\mathcal{U}\{0,M-1\}$	$\ln(\sigma)\sim\mathcal{N}(P_{\text{mean}},P_{\text{std}}^2)$
Loss weighting	$\lambda(\sigma)$	$1/\sigma^2$	$1/\sigma^2$	$1/\sigma^2$ (note: *)	$\left(\sigma^2+\sigma_{\text{data}}^2\right)/(\sigma\cdot\sigma_{\text{data}})^2$

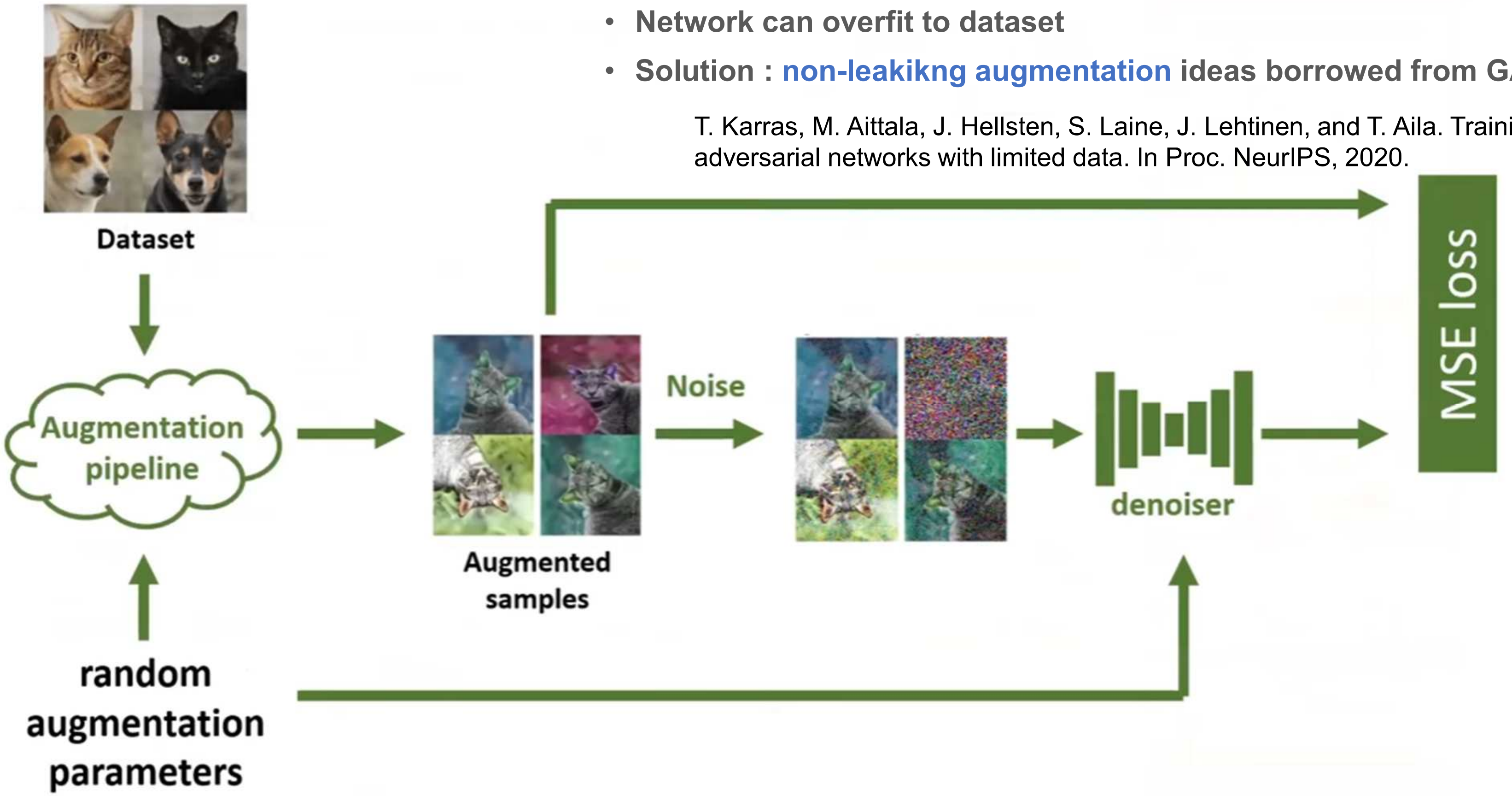
Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "Elucidating the Design Space of Diffusion-Based Generative Models," NeurIPS 2022.

Overfitting and Augmentations

- Network can overfit to dataset
- Solution : non-leakikng augmentation ideas borrowed from GAN literature

T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. In Proc. NeurIPS, 2020.



Youtube Presentaiton

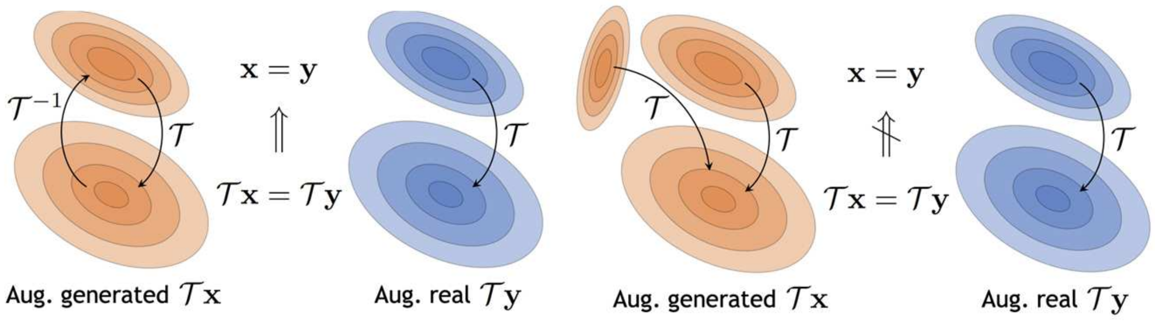
Training GANs With Limited Data, StyleGAN2 with Adaptive Discriminator Augmentation (ADA)

https://medium.com/swlh/training-gans-with-limited-data-22a7c8ffce78

https://yun905.tistory.com/56

Non-leakikng augmentation & ADA(Adaptive Discriminator Augmentation)

- 만약 training data에 90도 회전이라는 augmentation을 주면, generator는 90도 돌아간 이미지를 생성한다. 이것을 **leaking**이라고 한다. Discriminator가 augmentation된 이미지와 그렇지 않은 이미지를 구분하지 못하게 되는 것이다.
- **Non-leaking augmentation**을 위해서는 이미지에 transformation을 가하되, 그 transformation이 확률 분포의 관점에서 ***invertible***해야 한다. Invertible한 augmentation을 가했을 경우 training 과정에서 모델이 corruption을 걸러내고 원래의 분포를 잘 학습하게 된다고 한다.
- 가령 전체 이미지의 90%를 0으로 만드는 변환을 한다고 하자. 그러면 90%의 검은 이미지를 제외하고 10%의 진짜 이미지를 찾을 수 있고, 그 augmentation을 undo 할 수 있다. 이것은 확률 분포의 관점에서 invertible한 변환의 예이다. 이번에는 {0, 90, 180, 270}도 중 무작위로 골라 이미지를 회전하는 augmentation을 생각해보자. 그러면 원래의 방향을 가늠할 수 없으므로 undo할 수 없고, 따라서 invertible하지 않다. 이런 augmentation에서는 leaking이 발생하게 된다.
- 여기서, augmentation을 p의 확률로 적용한다고 하자. 그러면 회전되지 않은 이미지의 개수가 늘어나므로 진짜 이미지를 구분할 수 있고, invertible하게 된다. 다시 말해, 어떤 augmentation이 p 값에 따라 leaking 할 수도 있고 non-leaking 할 수도 있다는 뜻이다



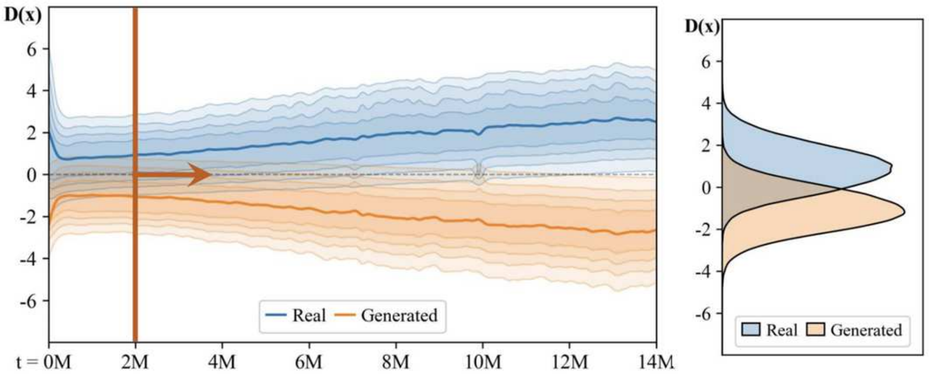
- The generator is forced to match the fake distribution x to the real distribution y in order to match the transformed distributions Tx and Ty.
- If we apply an invertible transformation T to the generated and real distributions x and y, then it is sufficient to match augmented distributions Tx and Ty in order to match the original distributions x and y.
- Theoretically, if the augmentation operator T is “invertible”, there exists one and only one x for the augmented distribution Tx, and there should be no “leaks” in x. However, in practice, due to limitations of finite sampling, finite representational power of the networks, inductive bias and training dynamics, very high values of p leads to leaking of augmentations in the generated images.

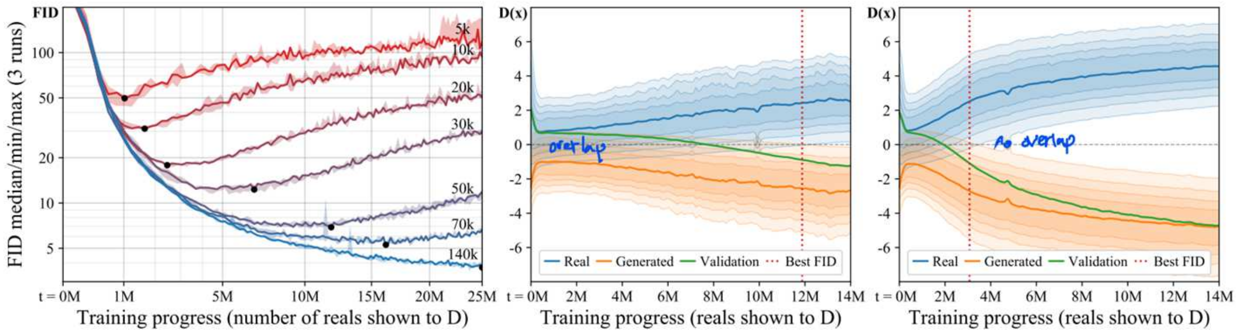
Training GANs With Limited Data, StyleGAN2 with Adaptive Discriminator Augmentation (ADA)

https://medium.com/swlh/training-gans-with-limited-data-22a7c8ffce78

https://yun905.tistory.com/56

Non-leakikng augmentation & ADA(Adaptive Discriminator Augmentation)





(a) Convergence of FFHQ (256 × 256) (b) Discriminator outputs, 50k (c) Discriminator outputs, 20k

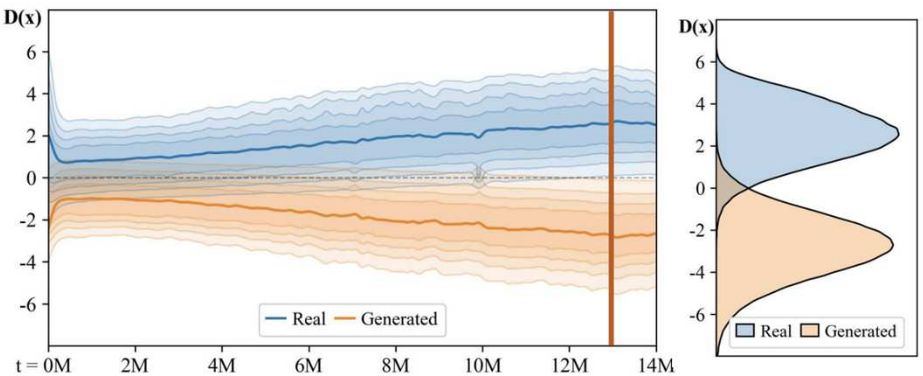


Figure. As the training progresses, the overlap between discriminator output distributions for real and generated images decreases

- The standard way of quantifying overfitting is to use a separate validation set and observe its behavior relative to the training set. When overfitting kicks in, the validation set starts to behave increasingly like the generated images, and the discriminator outputs for real and generated samples begin to diverge.
- Figure (a) : 데이터 수와 성능은 매우 밀접한 관계를 가지고 있음. 적은 데이터에서는 training이 diverge함.
- Figure (b)와 (c) : Discriminator output으로 discriminator overfitting이 일어나는 것을 확인 할 수 있음. Discriminator가 적은 수의 training data에 overfit되어서 overlap이 없어지면 FID 도 하락함을 알 수 있음.

Training GANs With Limited Data, StyleGAN2 with Adaptive Discriminator Augmentation (ADA)

https://medium.com/swlh/training-gans-with-limited-data-22a7c8ffce78

https://yun905.tistory.com/56

Non-leakikng augmentation & ADA(Adaptive Discriminator Augmentation)

Two plausible overfitting heuristics to measure overfitting

$$r_v = \frac{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{generated}}]} \qquad r_t = \mathbb{E}[\text{sign}(D_{\text{train}})]$$

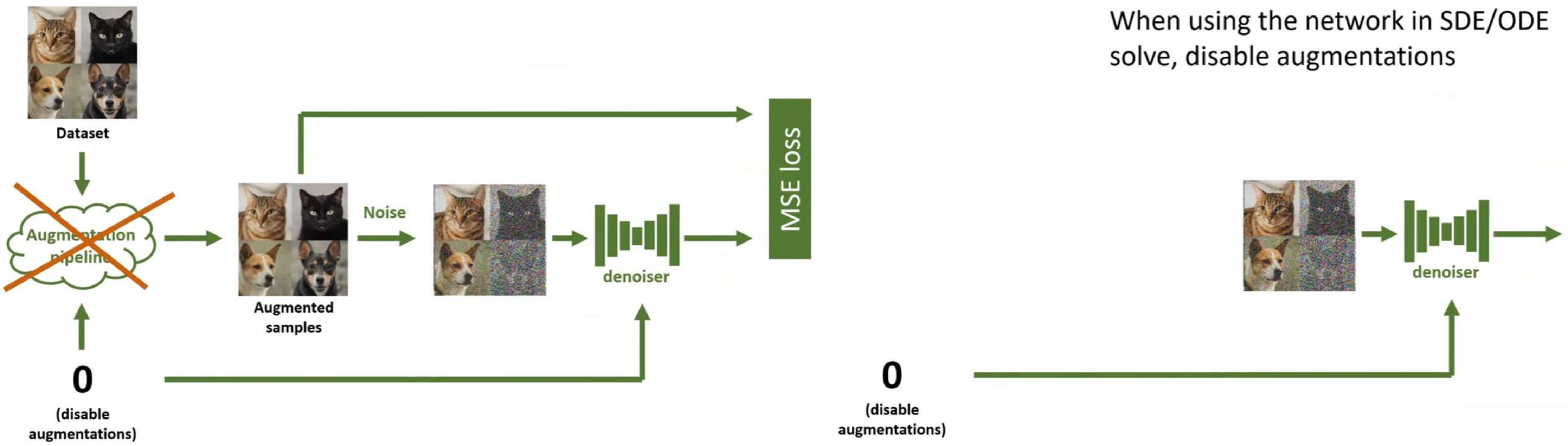
- The first heuristic rv, expresses the output for a validation set relative to the training set and generated images. The numerator is 0 when the training and validation set behave exactly the same, hence r=0 means no overfitting. The numerator and denominator are the same when the generated and validation set behave exactly the same, hence r=1 indicates complete overfitting.
- Since it assumes the existence of a separate validation set in an already small dataset, it is not feasible to calculate the rv heuristic. Hence, the authors turn to rt - which estimates the portion of the training set that gets positive discriminator outputs - to identify overfitting and dynamically adapt the augmentation probability p as the training progresses:
- rt too high → augment more (increase p)
- rt too low → augment less (decrease p)

왼쪽은 단지 비교를 위한 수식이고, 실제로 대부분의 실험에서 오른쪽을 사용하였다.
왼쪽의 수식은 validation set을 필요로 하기 때문에 limited dataset에서 적용하기 힘든 부분이 있다.
둘 다 0~1의 범위에서 1이면 discriminator overfitting이 매우 심한 것이고 0이면 전혀 없는 것이다. Overfitting이 심해질수록 discriminator가 validation set을 generated image라고 판단한다는 것을 위에서 확인했다. 이 경우 r_v는 1이 된다. r_t에서 sign이 붙은 이유는 단지 그렇게 하면 여러 세팅에서 덜 sensitive하기 때문이다.

이러한 heuristic의 target value를 0~1 사이의 임의의 값으로 정하고, 그 값을 기준으로 p값을 adaptive하게 조절하는 것을 ADA라고 한다.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Evaluation of training (deterministic sampling)

	CIFAR-10 [28] at 32×32				FFHQ [26] 64×64		AFHQv2 [7] 64×64	
	Conditional		Unconditional		Unconditional		Unconditional	
Training configuration	VP	VE	VP	VE	VP	VE	VP	VE
A Baseline [42] (*pre-trained)	2.48	3.11	3.01*	3.77*	3.39	25.95	2.58	18.52
B + Adjust hyperparameters	2.18	2.48	2.51	2.94	3.13	22.53	2.43	23.12
C + Redistribute capacity	2.08	2.52	2.31	2.83	2.78	41.62	2.54	15.04
D + Our preconditioning	2.09	2.64	2.29	3.10	2.94	3.39	2.79	3.81
E + Our loss function	1.88	1.86	2.05	1.99	2.60	2.81	2.29	2.28
F + Non-leaky augmentation	**1.79**	**1.79**	**1.97**	**1.98**	**2.39**	**2.53**	**1.96**	**2.16**
NFE	35	35	35	35	79	79	79	79

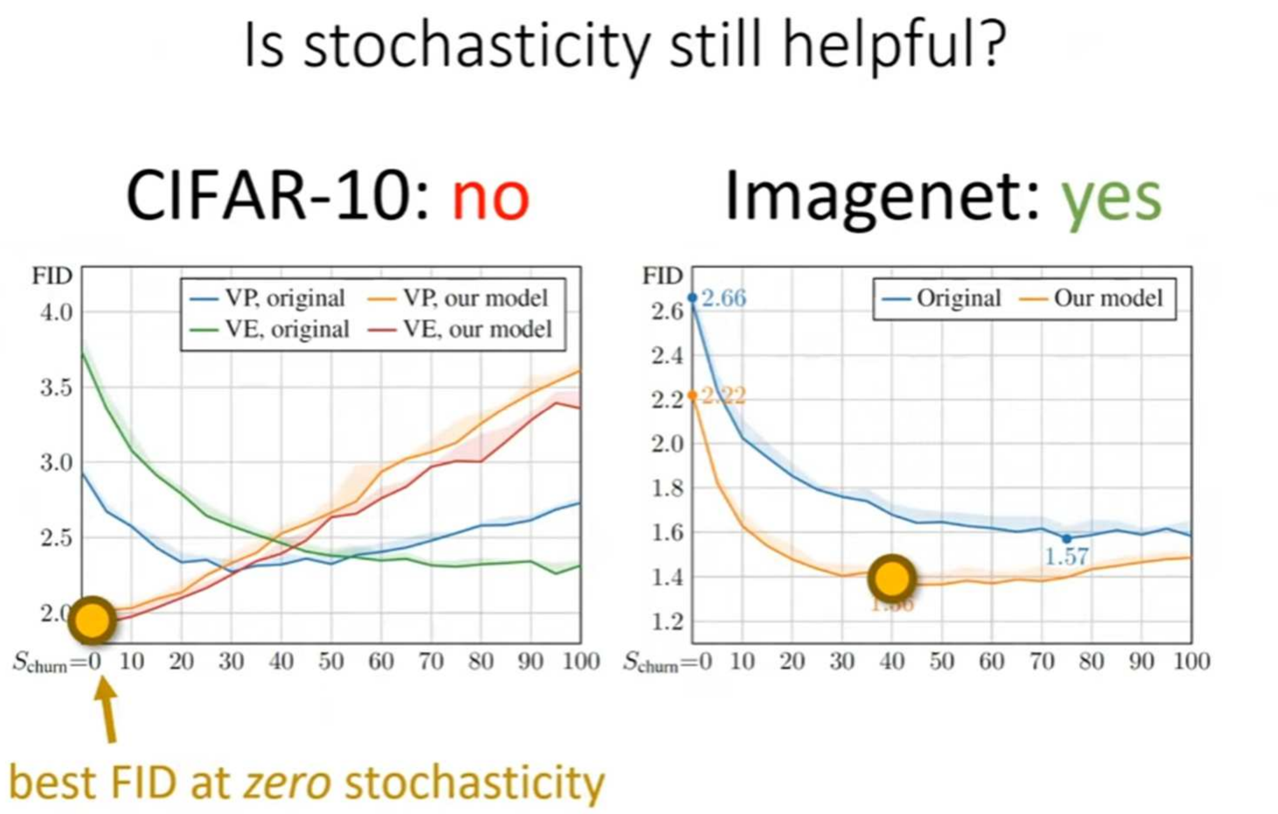
Imagenet 64 x 64 (stochastic sampling)

FID 1.36 **state of the art FID**

With deterministic sampling when we enabled the stochastic sampling and tailor it for these architectures for ImageNet and use this retrained these networks we trained ourselves using these principles, We get a FID of 1.36.

Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.



Youtube Presentaiton

EDM : Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, "**Elucidating the Design Space of Diffusion-Based Generative Models**," NeurIPS 2022.

Conclusions

- Modular design of diffusion models
  - Training, and sampling and network architectures are not tightly coupled
- Careful design of each “module” yields considerable improvements
- Stochasticity is a double-edged sword
- Higher resolutions, network architectures, conditioning/guidance, large scale datasets, … ?
  - Ripe for principled analysis of foundations