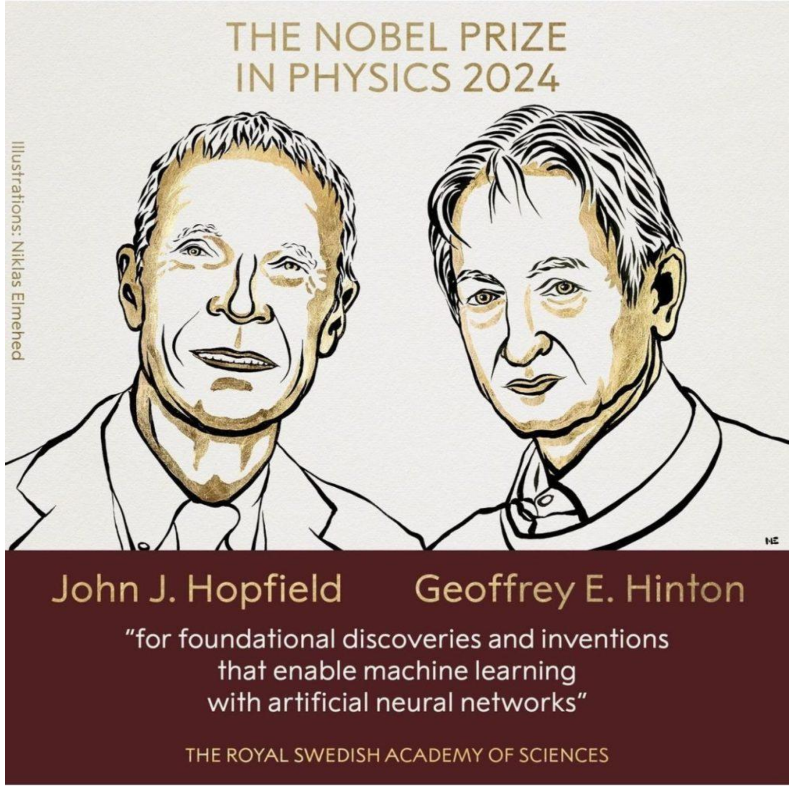
Probabilistic Foundation and Opportunities of Diffusion Models

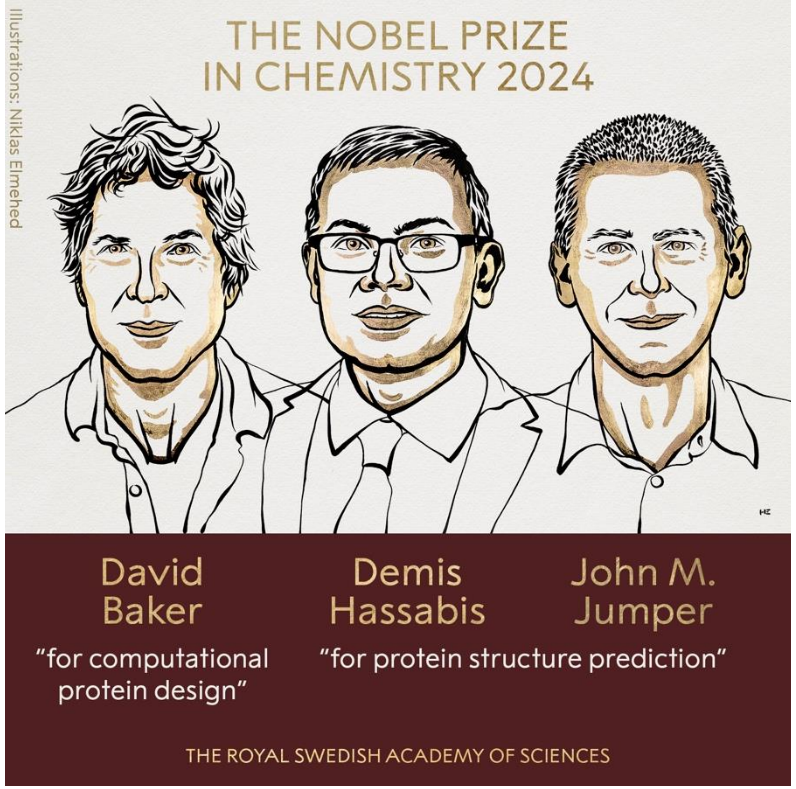
Mengdi Wang & Minshuo Chen

PRINCETON UNIVERSITY

Northwestern University

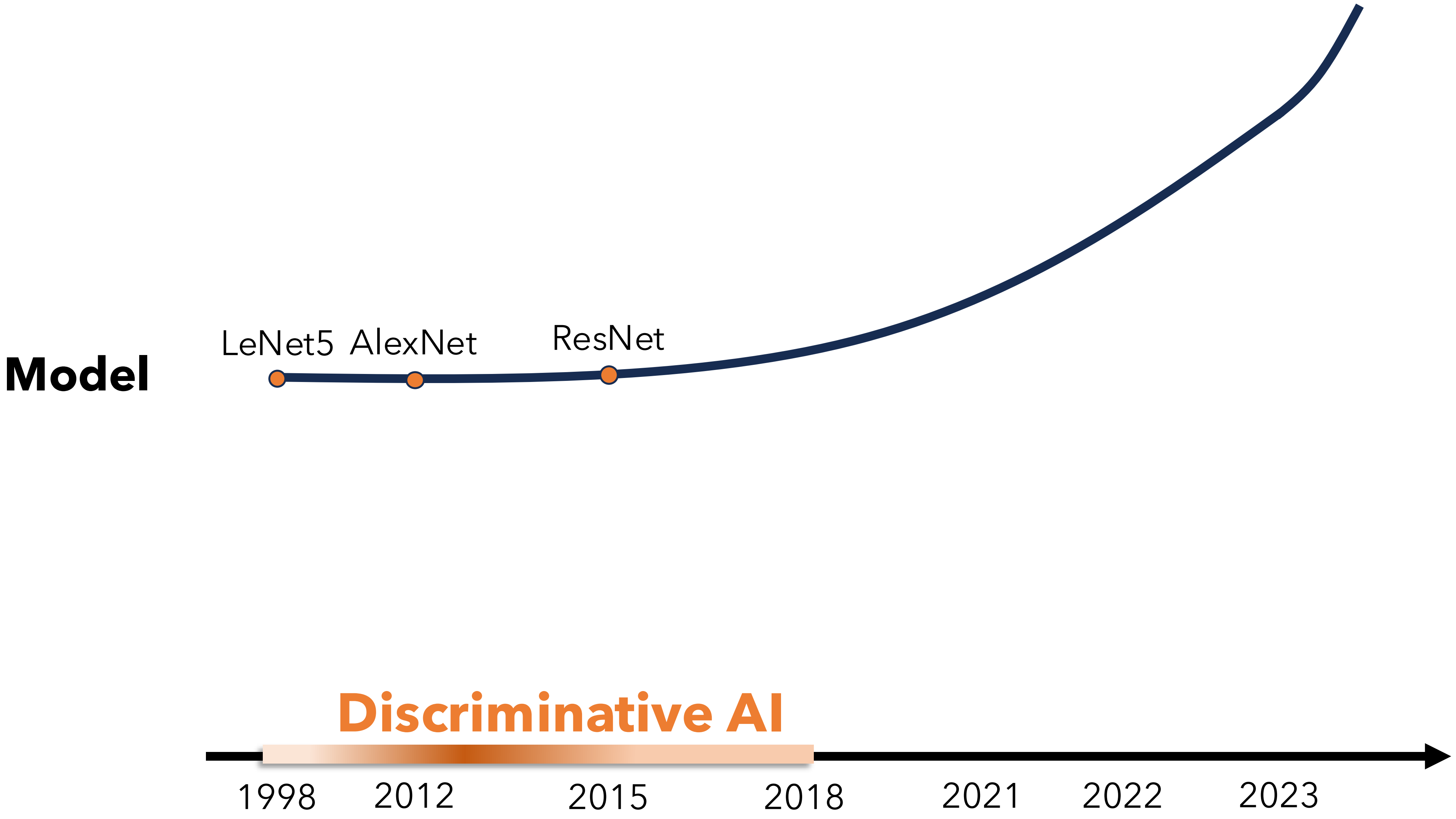
AI Comes to The Nobels





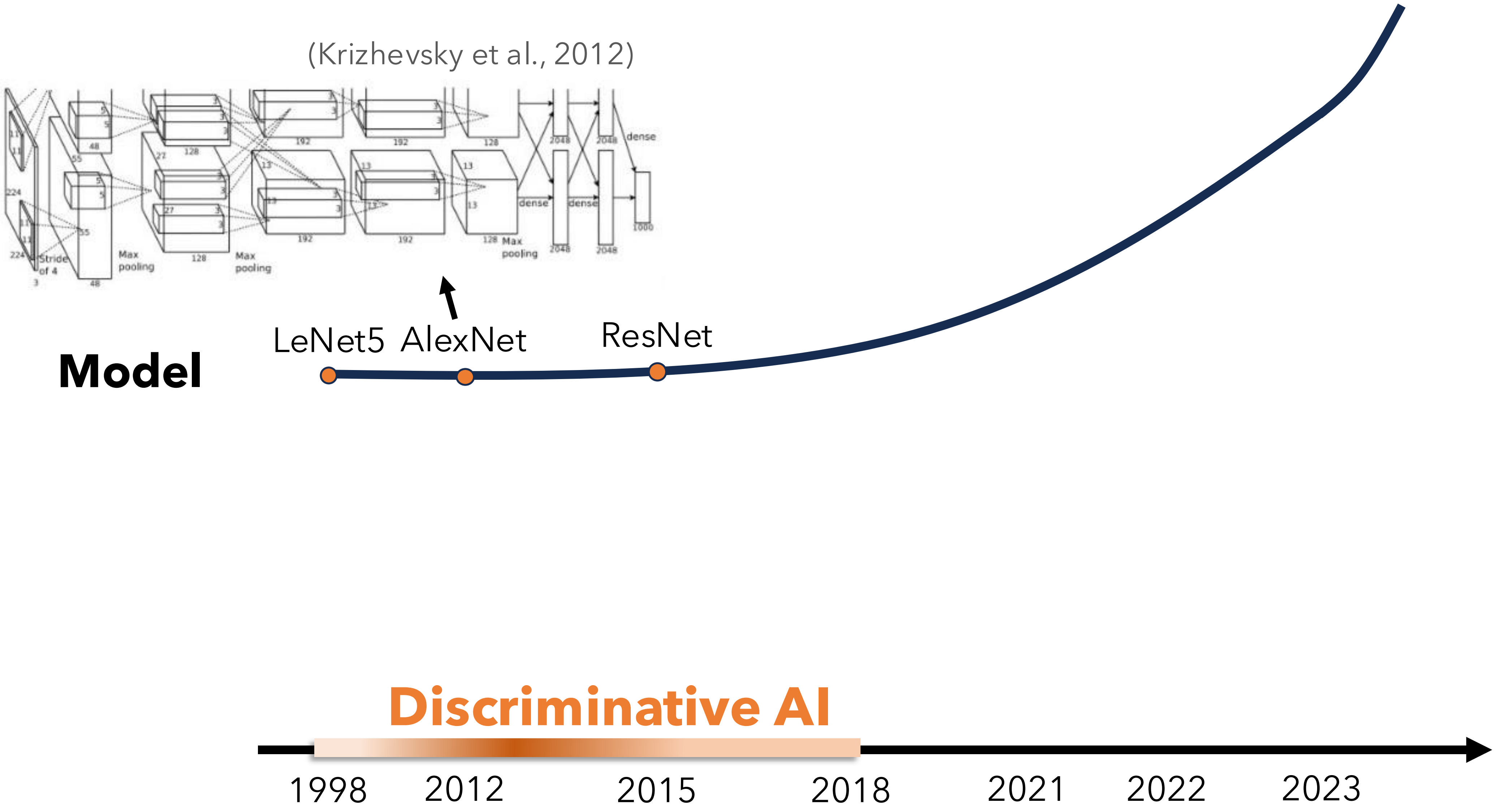
"For me, this story highlights how far **AI** has come and how much more potential there is to explore."

Millennium Growth of AI



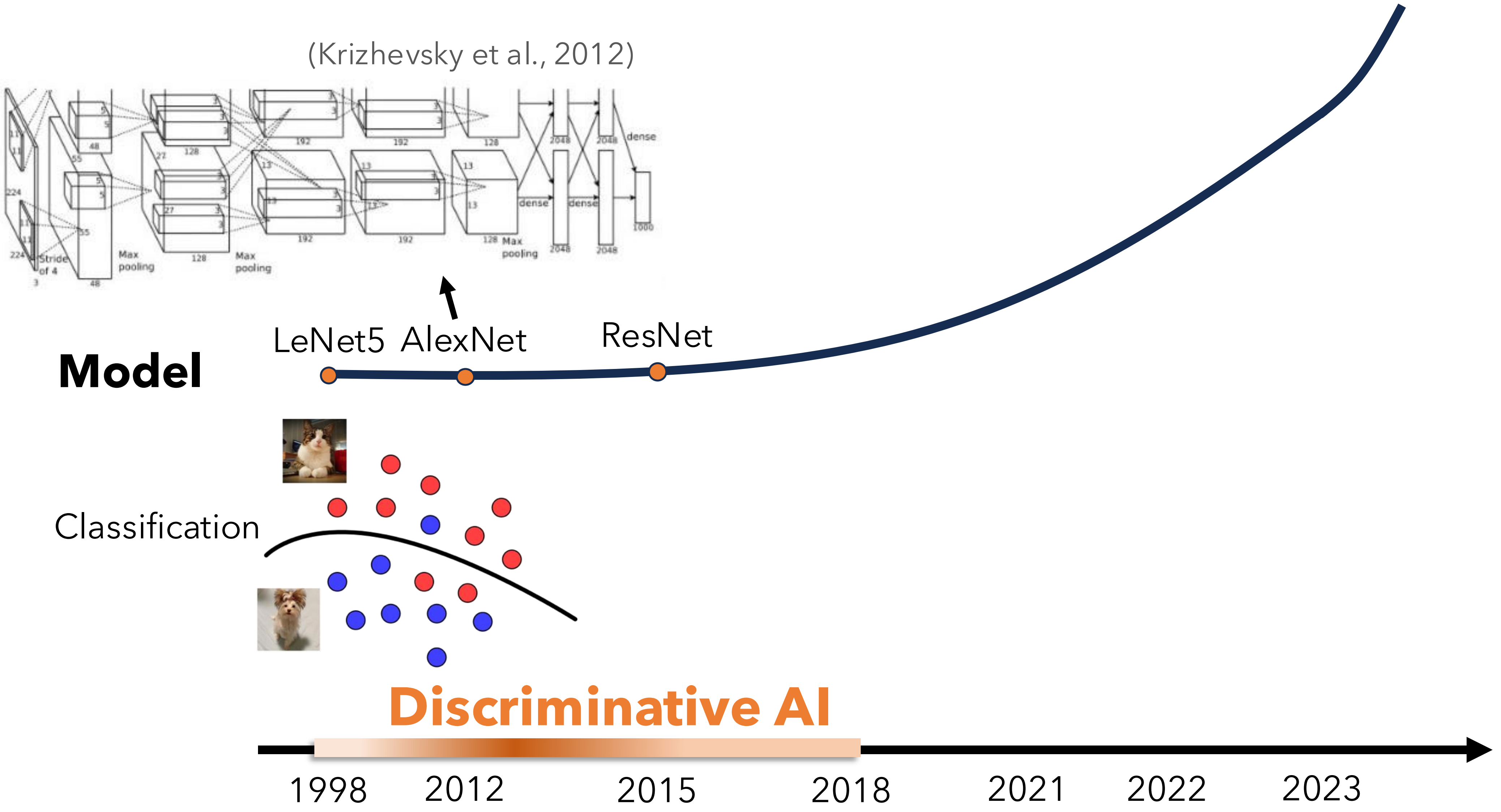
-- Thanks to blogs by Rockwell Anyoha , Toloka Team and Rick Merritt

Millennium Growth of AI



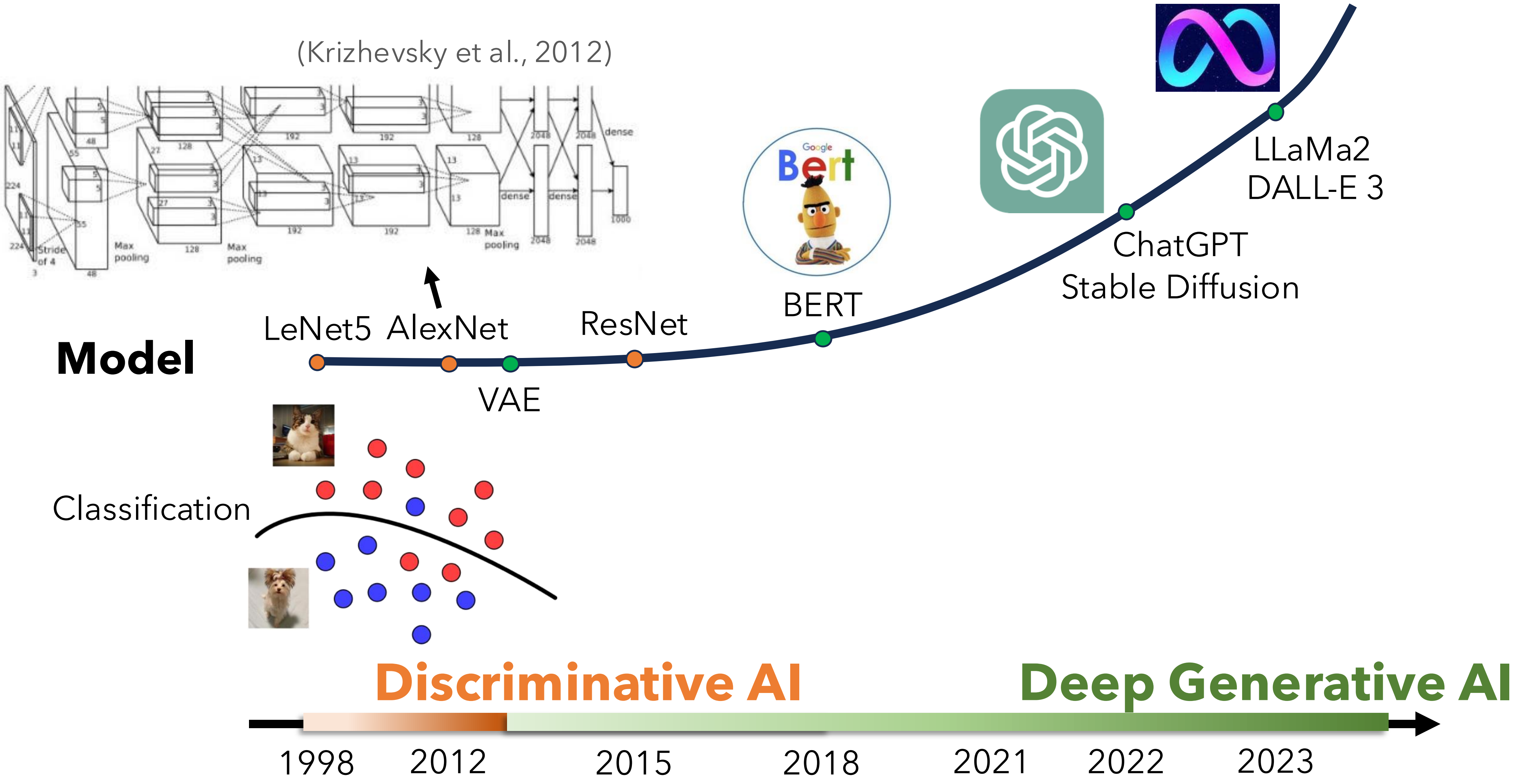
-- Thanks to blogs by Rockwell Anyoha , Toloka Team and Rick Merritt

Millennium Growth of AI

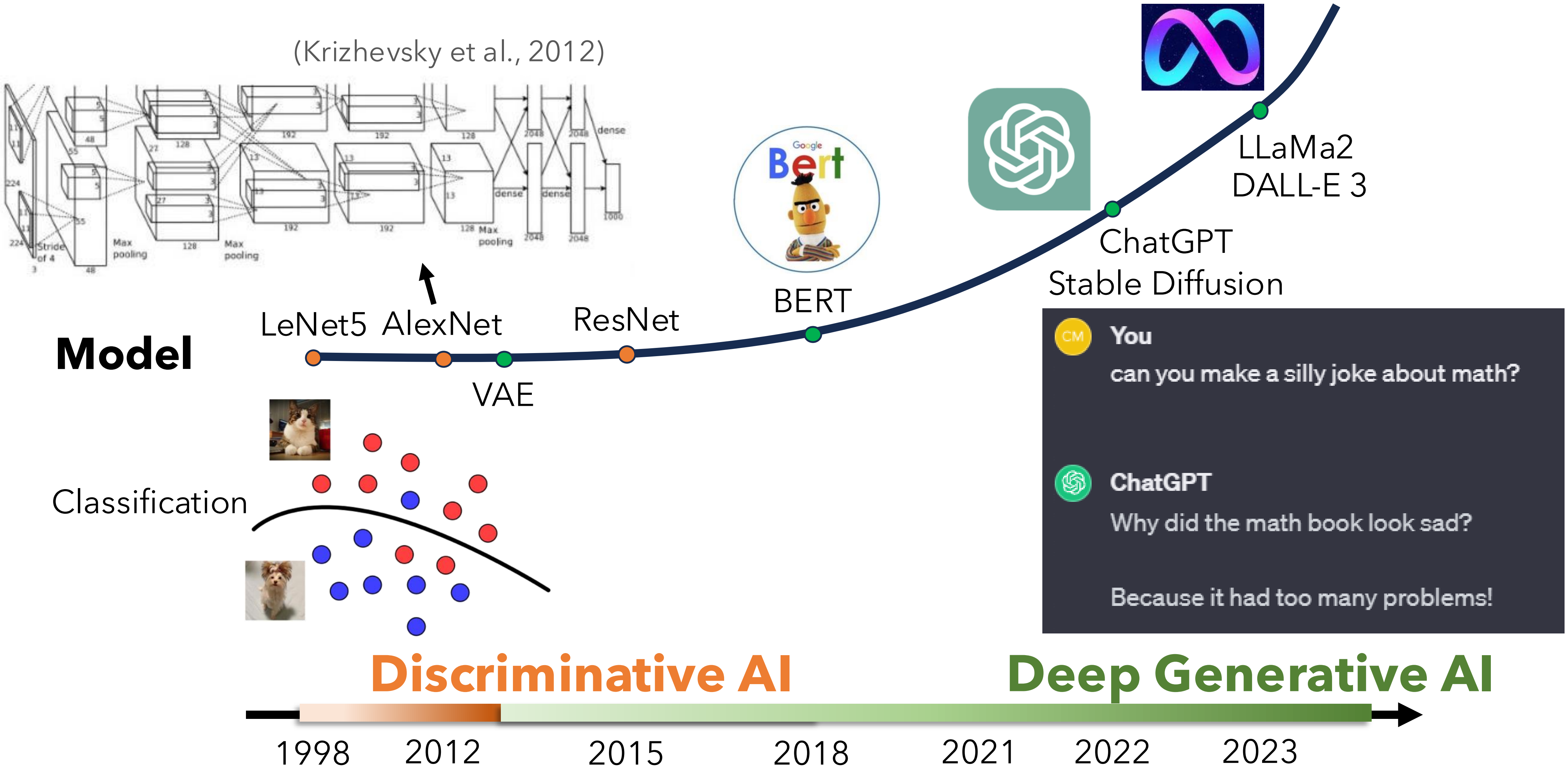


-- Thanks to blogs by Rockwell Anyoha , Toloka Team and Rick Merritt

Millennium Growth of AI

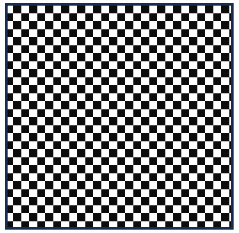


Millennium Growth of AI

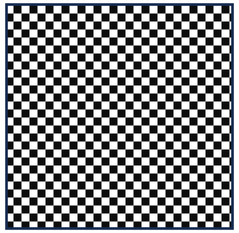


-- Thanks to blogs by Rockwell Anyoha , Toloka Team and Rick Merritt

Transformative Power of Deep Generative AI

Noise   **Generative AI**

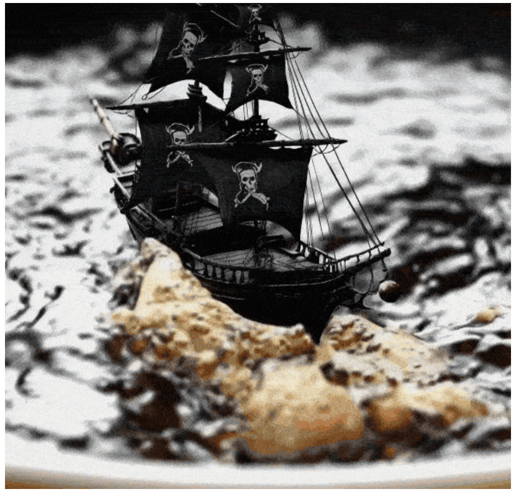
Transformative Power of Deep Generative AI

Noise   **Generative AI**

ChatGPT

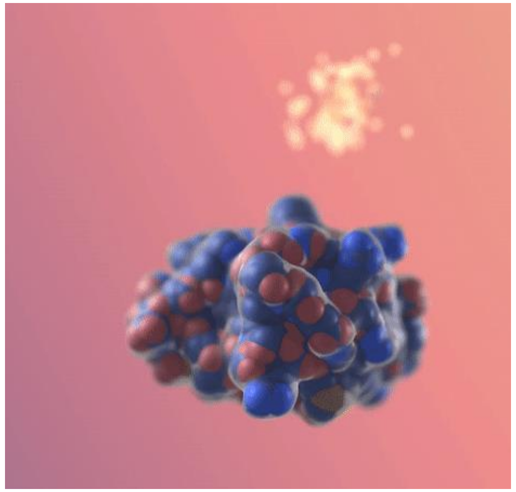
Language

Sora



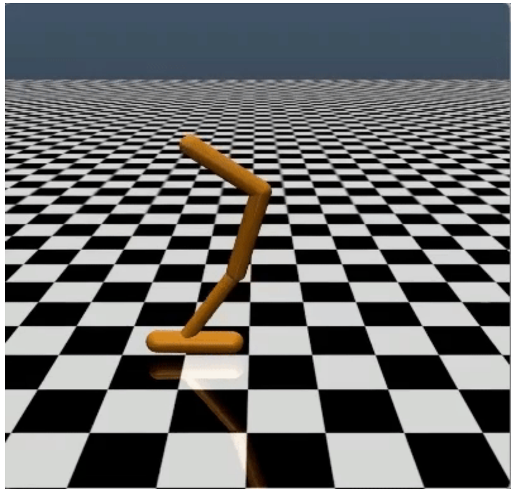
Video

RFDiffusion



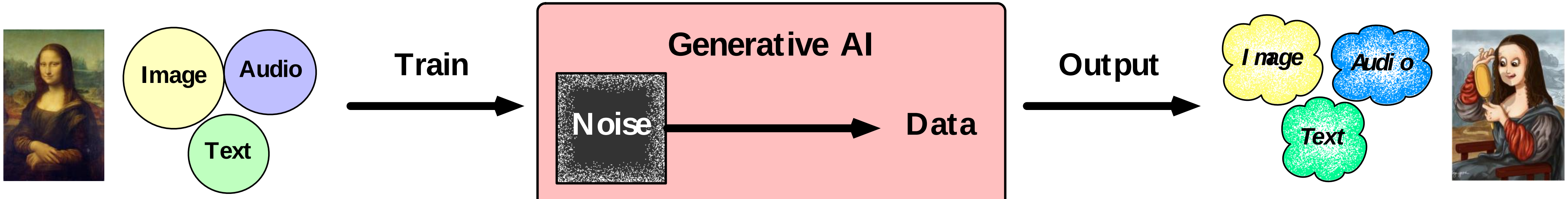
Biology

Decision Diffuser

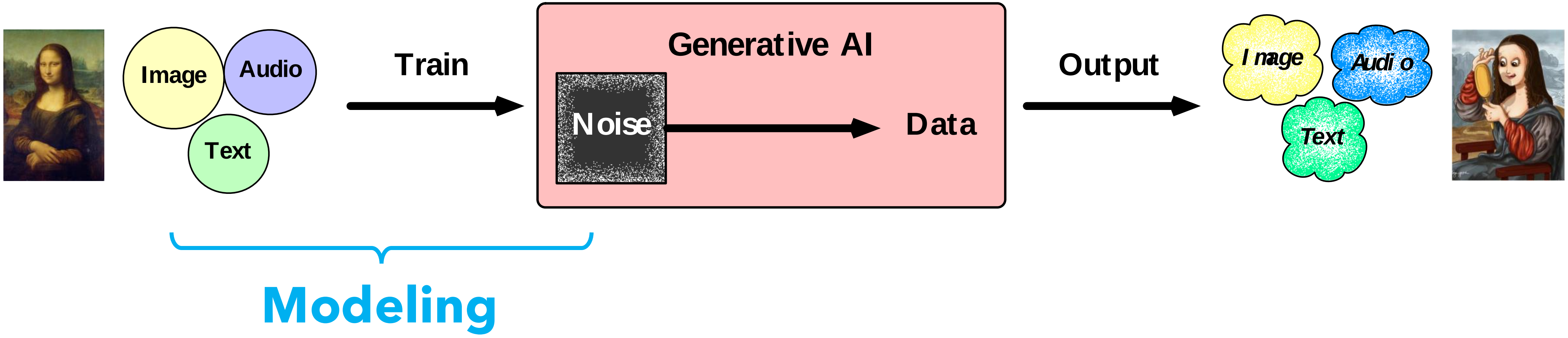


RL/control

Constitution of Deep Generative AI

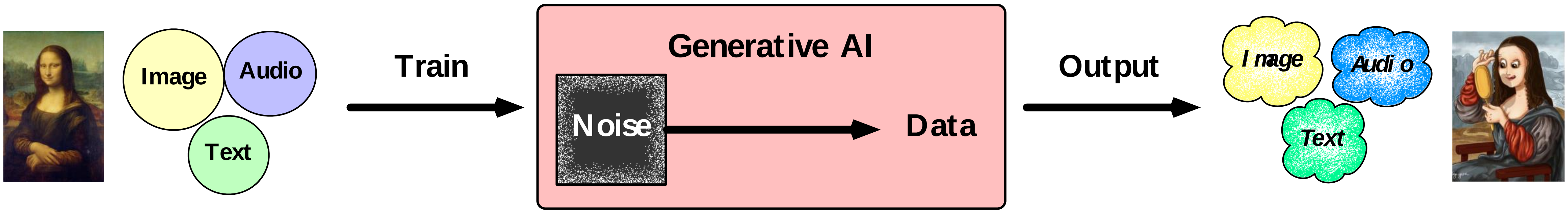


Constitution of Deep Generative AI



$$P(\cdot), P(\cdot|y)$$

Constitution of Deep Generative AI



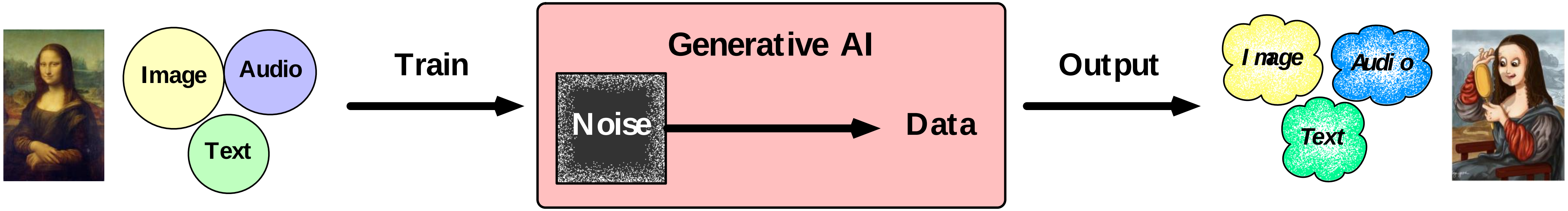
Modeling

$$P(\cdot), P(\cdot|y)$$

Sampling

$$x \sim P(\cdot), P(\cdot|y)$$

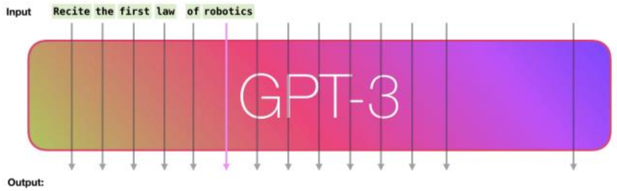
Constitution of Deep Generative AI



Modeling

$$P(\cdot), P(\cdot|y)$$

Autoregressive

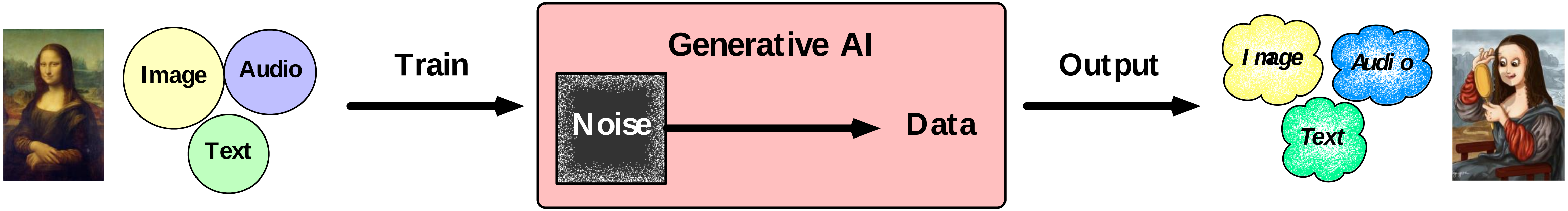


Sampling

$$x \sim P(\cdot), P(\cdot|y)$$

Autoregressive

Constitution of Deep Generative AI



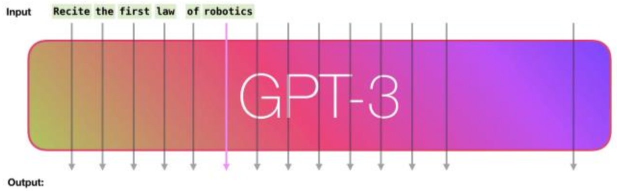
Modeling

$$P(\cdot), P(\cdot|y)$$

Sampling

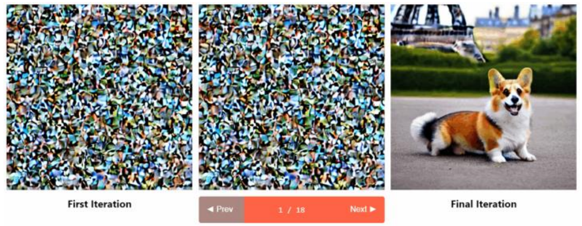
$$x \sim P(\cdot), P(\cdot|y)$$

Autoregressive



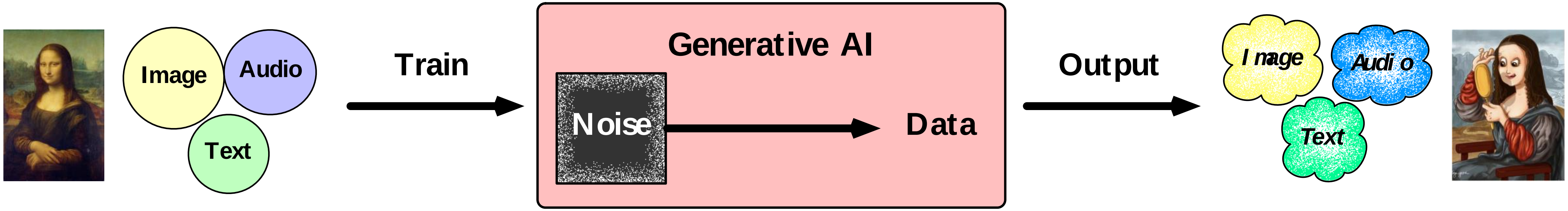
Autoregressive

Diffusion



Denoising

Constitution of Deep Generative AI



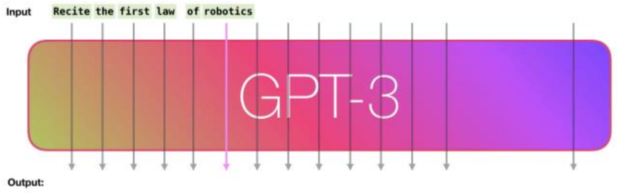
Modeling

$$P(\cdot), P(\cdot|y)$$

Sampling

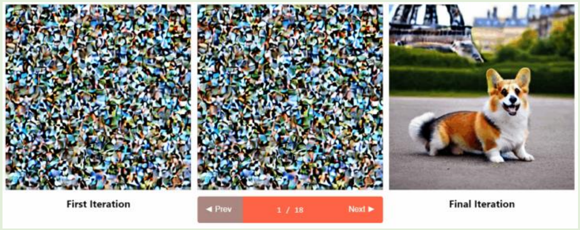
$$x \sim P(\cdot), P(\cdot|y)$$

Autoregressive



Autoregressive

Diffusion



Denoising

New Promises of Diffusion Models

New Promises of Diffusion Models

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

New Promises of Diffusion Models

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

Score-Based Generative Modeling through Stochastic Differential Equations

Yang Song*

durk@google.com

abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

colorization. Combined with multiple architectural improvements, we achieve $\boxed{\text{record-breaking performance}}$ 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

New Promises of Diffusion Models

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal* OpenAI prafulla@openai.com

Alex Nichol* OpenAI alex@openai.com

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song* Stanford Uni…

durk@google.com

abhishk@google.com

Stefano Ermon Stanford University ermon@cs.stanford.edu

Ben Poole Google Brain pooleb@google.com

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

stability.ai

Stable Diffusion XL

Get involved with the fastest growing open software project. Download and join other developers in creating incredible applications with Stable Diffusion XL as a foundation model.

Try Stable Diffusion XL | Download Code

Stable Audio

Generate music and sound effects in high quality using cutting-edge audio diffusion technology.

Learn more

Try Stable Audio

Stable LM

Experience cutting edge open access language models. Our Language researchers innovate rapidly and release open models that rank amongst the best in the industry.

Learn More

Stable Video

Our first open generative AI video model based on the image model Stable Diffusion.

Join App Waitlist

Download Weights

A thriving Stability AI Community

We've amassed a community of more than 300,000 creators, developers, and researchers around the world.

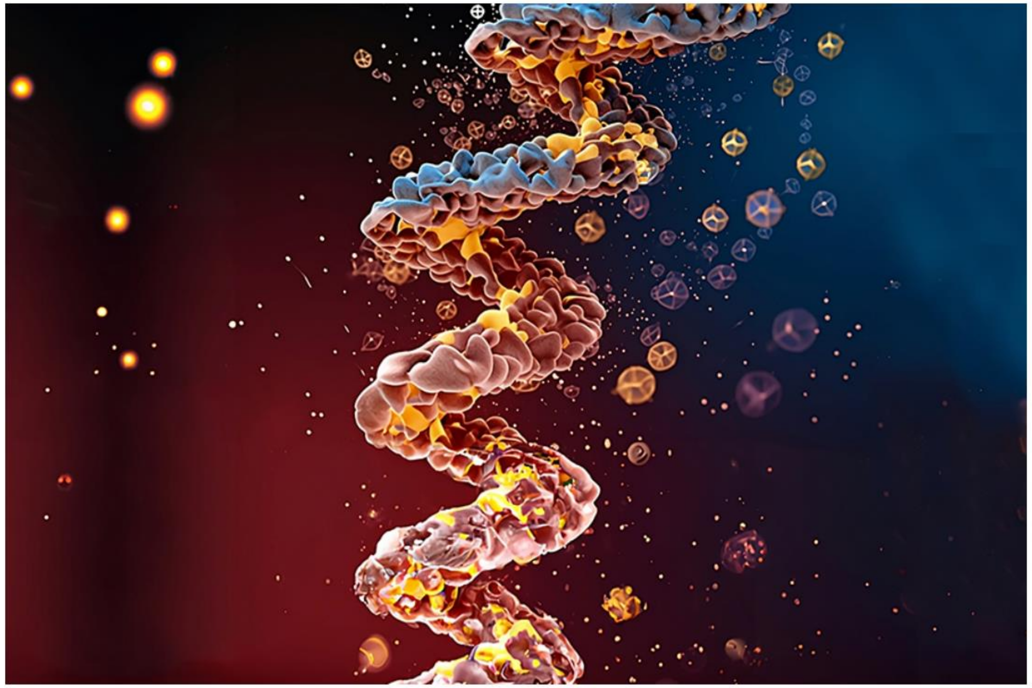
10M	270,000	+170M	400M
Global users just two months after its release	Stable Diffusion's Discord channel Members	Images generated with Clipdrop SDXL	Images generated using Stability AI's API

New Promises of Diffusion Models

Generative AI imagines new protein structures

"FrameDiff" is a computational tool that uses generative AI to craft new protein structures, with the aim of accelerating drug development and improving gene therapy.

Rachel Gordon | MIT CSAIL
July 12, 2023



Biology is a wondrous yet delicate tapestry. At the heart is DNA, the master weaver that encodes proteins, responsible for orchestrating the many biological functions that sustain life within the human body. However, our body is akin to a finely tuned instrument, susceptible to losing its harmony. After all, we're faced with an ever-changing and relentless natural world: pathogens, viruses, diseases, and cancer.

stability.ai

Stable Diffusion XL

Get involved with the fastest growing open software project. Download and join other developers in creating incredible applications with Stable Diffusion XL as a foundation model.

Try Stable Diffusion XL | Download Code

Stable Audio

Generate music and sound effects in high quality using cutting-edge audio diffusion technology.

Learn more

Try Stable Audio

Stable LM

Experience cutting edge open access language models. Our Language researchers innovate rapidly and release open models that rank amongst the best in the industry.

Learn More

Stable Video

Our first open generative AI video model based on the image model Stable Diffusion.

Join App Waitlist

Download Weights

A thriving Stability AI Community

We've amassed a community of more than 300,000 creators, developers, and researchers around the world.

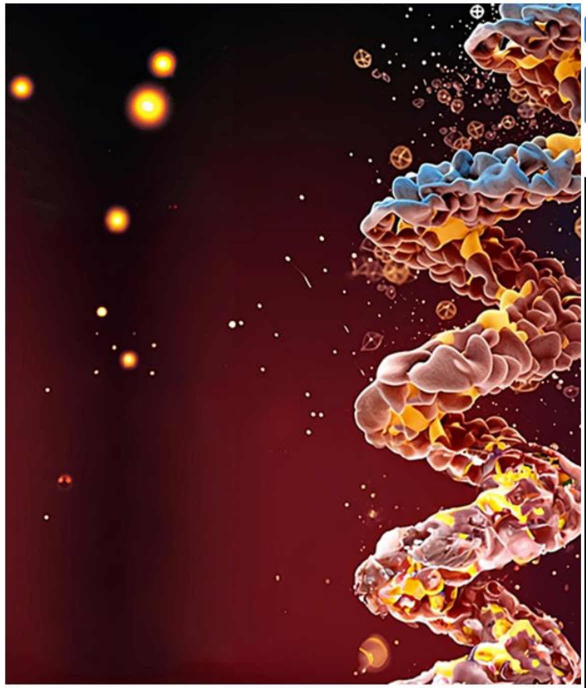
10M	270,000	+170M	400M
Global users just two months after its release	Stable Diffusion's Discord channel Members	Images generated with Clipdrop SDXL	Images generated using Stability AI's API

New Promises of Diffusion Models

Generative AI imagines new

"FrameDiff" is a computational tool that use
protein structures, with the aim of accelerat
improving gene therapy.

Rachel Gordon | MIT CSAIL
July 12, 2023



color
reco
with

Biology is a wondrous yet delicate tapestry. At the hea
encodes proteins, responsible for orchestrating the m
within the human body. However, our body is akin to a
losing its harmony. After all, we're faced with an ever-c
pathogens, viruses, diseases, and cancer.

Diffusion models are now turbocharging reinforcement learning systems

By **Ben Dickson** - March 4, 2024

Like 75

Facebook | Twitter | ReddIt | Linkedin

Image generated with Bing Image Creator

This article is part of our coverage of the latest in AI research.

Stable Diffusion XL

Get involved with the fastest growing open software project. Download and join other developers in creating incredible applications with Stable Diffusion XL as a foundation model.

Try Stable Diffusion XL | Download Code

le LM

e cutting edge open access language
)ur Language researchers innovate
d release open models that rank
the best in the industry.

Learn More

Stable Video

Our first open generative AI video model based on the image model Stable Diffusion.

Join App Waitlist

Download Weights

Stability AI Community

ommunity of more than 300,000 creators,
and researchers around the world.

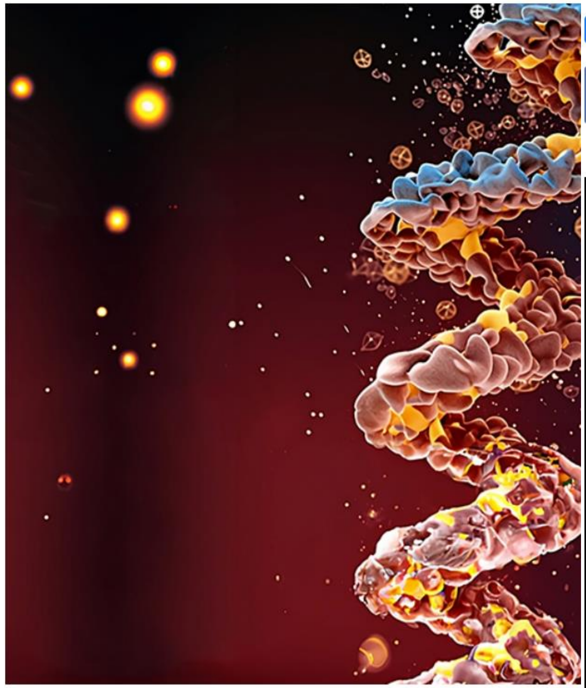
)0	+170M	400M
Discord	Images generated with Clipdrop SDXL	Images generated using Stability AI's API

New Promises of Diffusion Models

Generative AI imagines new

"FrameDiff" is a computational tool that use
protein structures, with the aim of accelerat
improving gene therapy.

Rachel Gordon | MIT CSAIL
July 12, 2023



color
reco
with

Biology is a wondrous yet delicate tapestry. At the hea
encodes proteins, responsible for orchestrating the m
within the human body. However, our body is akin to a
losing its harmony. After all, we're faced with an ever-c
pathogens, viruses, diseases, and cancer.

Diffusion models are turbocharging reinfo learning systems

By **Ben Dickson** - March 4, 2024

Like 75

Facebook | Twitter | ReddIt

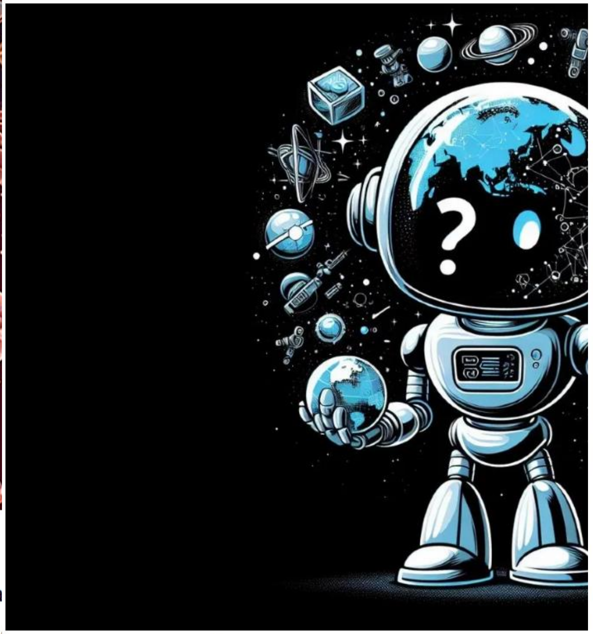


Image generated with Bing Image Creator

This article is part of our coverage of the late

ARTIFICIAL INTELLIGENCE

AniPortrait: Audio-Driven Synthesis of Photorealistic Portrait Animation

Published 3 days ago on May 3, 2024
By **Kunal Kejriwal**

Over the years, the creation of realistic and expressive portraits animations from static images and audio has found a range of applications including gaming, digital media, virtual reality, and a lot more. Despite its potential application, it is still difficult for developers to create frameworks capable of generating high-quality animations that maintain temporal consistency and are visually captivating. A major cause for the complexity is the need for intricate coordination of lip movements, head positions, and facial expressions to craft a visually compelling effect.

New Promises of Diffusion Models

Generative AI imagines new

“FrameDiff” is a computational tool that use
protein structures, with the aim of accelerat
improving gene therapy.

Rachel Gordon | MIT CSAIL
July 12, 2023

Biology is a wondrous yet delicate tapestry. At the hea
encodes proteins, responsible for orchestrating the m
within the human body. However, our body is akin to a
losing its harmony. After all, we’re faced with an ever-c
pathogens, viruses, diseases, and cancer.

color
reco
with

Diffusion models are turbocharging reinfo learning systems

By **Ben Dickson** - March 4, 2024

Like 75

Facebook Twitter ReddIt

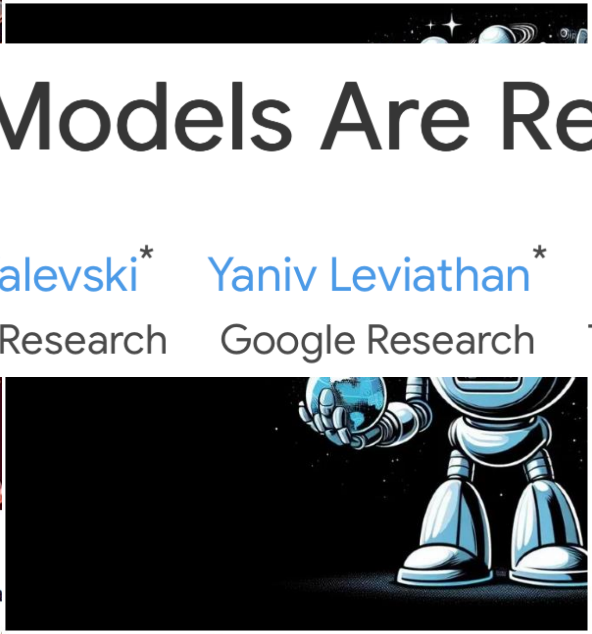


Image generated with Bing Image Creator

This article is part of our coverage of the late

ARTIFICIAL INTELLIGENCE

AniPortrait: Audio-Driven Synthesis of Photorealistic Portrait Animation

Published 3 days ago on May 3, 2024
By Kunal Kejriwal

Over the years, the creation of realistic and expressive portraits animations from static images and audio has found a range of applications including gaming, digital media, virtual reality, and a lot more. Despite its potential application, it is still difficult for developers to create frameworks capable of generating high-quality animations that maintain temporal consistency and are visually captivating. A major cause for the complexity is the need for intricate coordination of lip movements, head positions, and facial expressions to craft a visually compelling effect.

Diffusion Models Are Real-Time Game Engines

Dani Valevski* Yaniv Leviathan* Moab Arar*† Shlomi Fruchter*
Google Research Google Research Tel Aviv University Google DeepMind

Outline

- A probabilistic foundation for diffusion models
- How diffusion models capture diverse data
- How to leverage diffusion models
- Inspirations and future directions

Outline

Chen et al., “Challenges and Opportunities of Diffusion Models for Generative AI”, NSR 2024

- A probabilistic foundation for diffusion models
- How diffusion models capture diverse data
- How to leverage diffusion models
- Inspirations and future directions

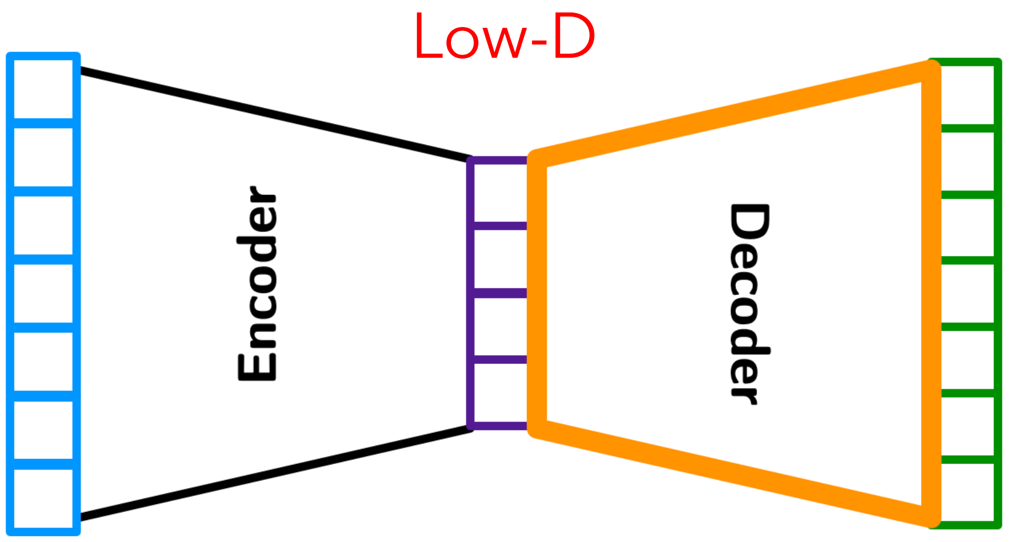
Outline

Chen et al., "Challenges and Opportunities of Diffusion Models for Generative AI", NSR 2024

- A probabilistic foundation for diffusion models
- How diffusion models capture diverse data
- How to leverage diffusion models
- Inspirations and future directions

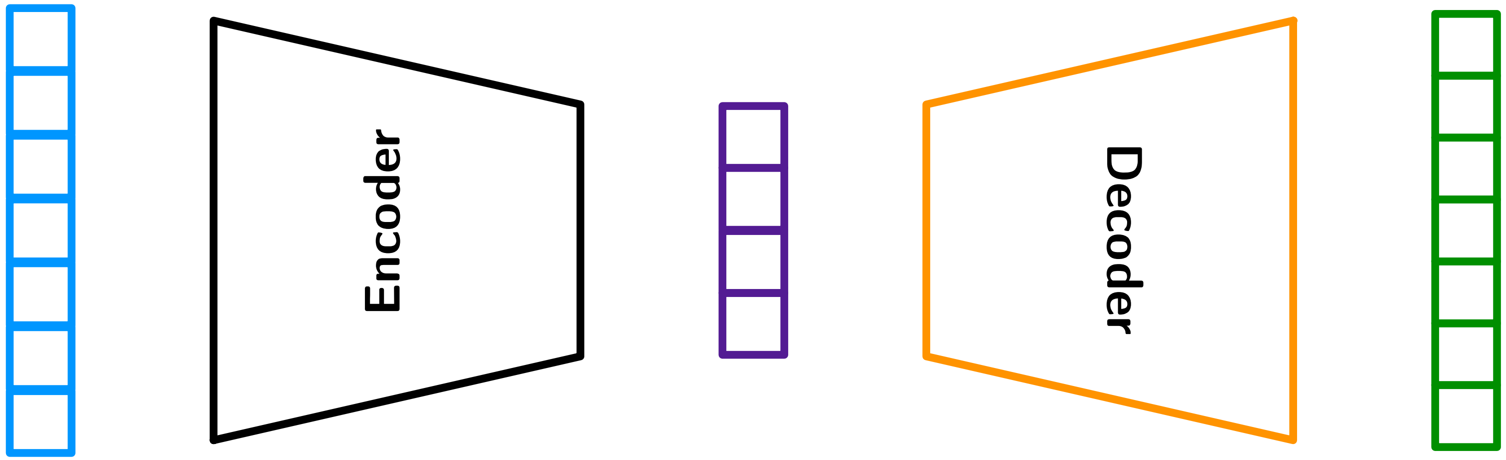
Foundation of Diffusion Models

Early Model of Deep Generative AI

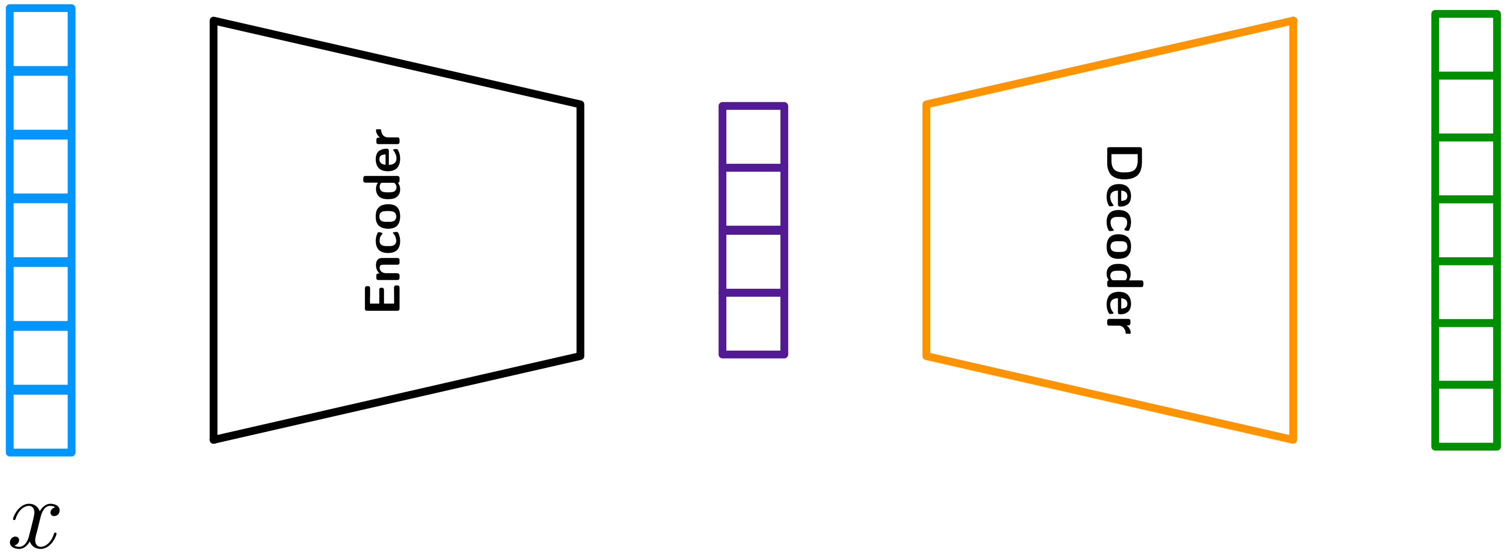


VAE (Kingma & Welling, 2013)

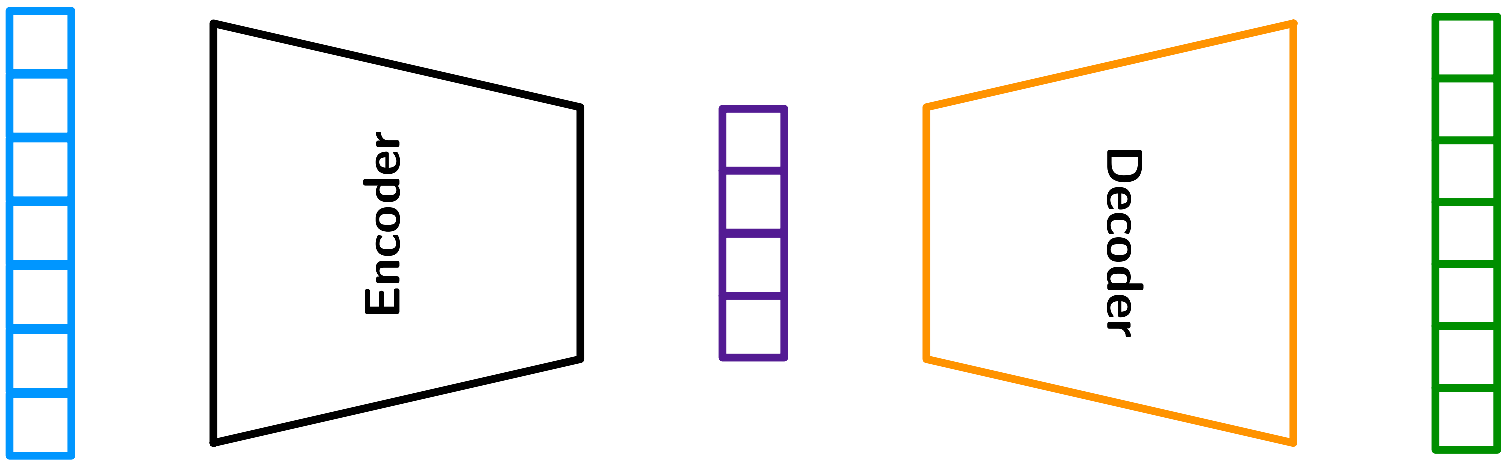
Dissemble VAE



Dissemble VAE

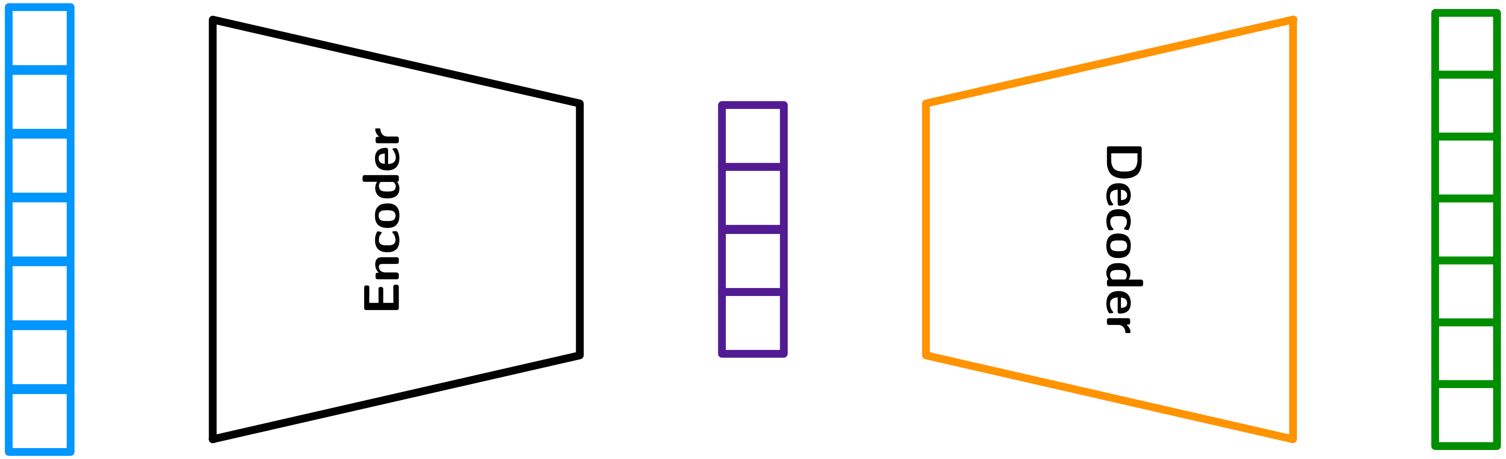


Dissemble VAE



$$x \longrightarrow p(z|x) \longrightarrow z$$

Dissemble VAE

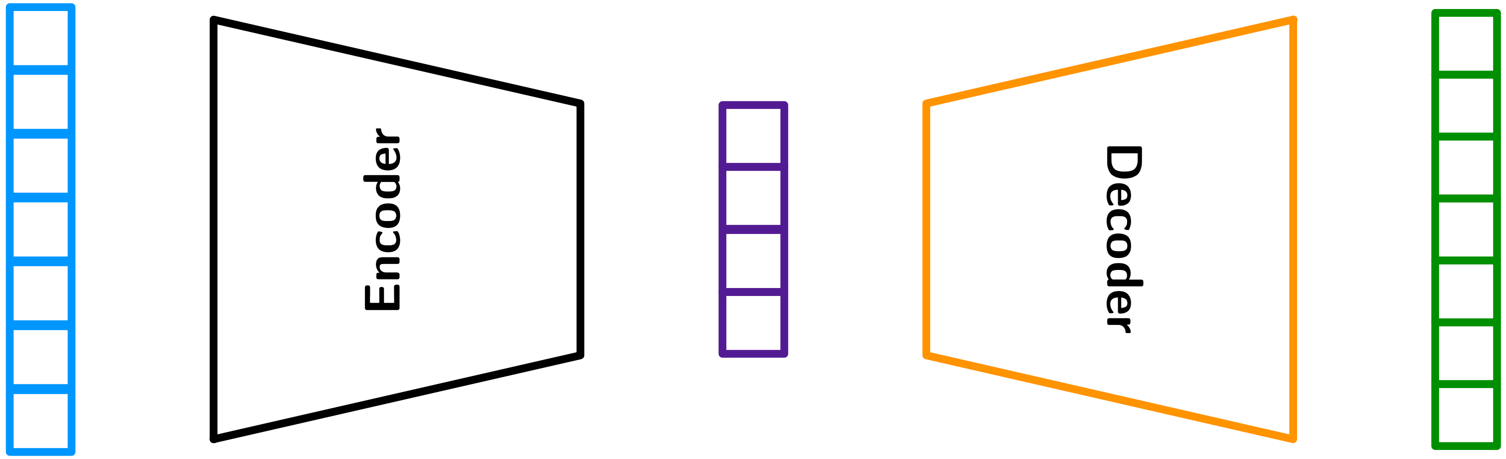


$x \longrightarrow p(z|x) \longrightarrow z$

$\wr$

$\mathrm{N}(0, I)$

Dissemble VAE

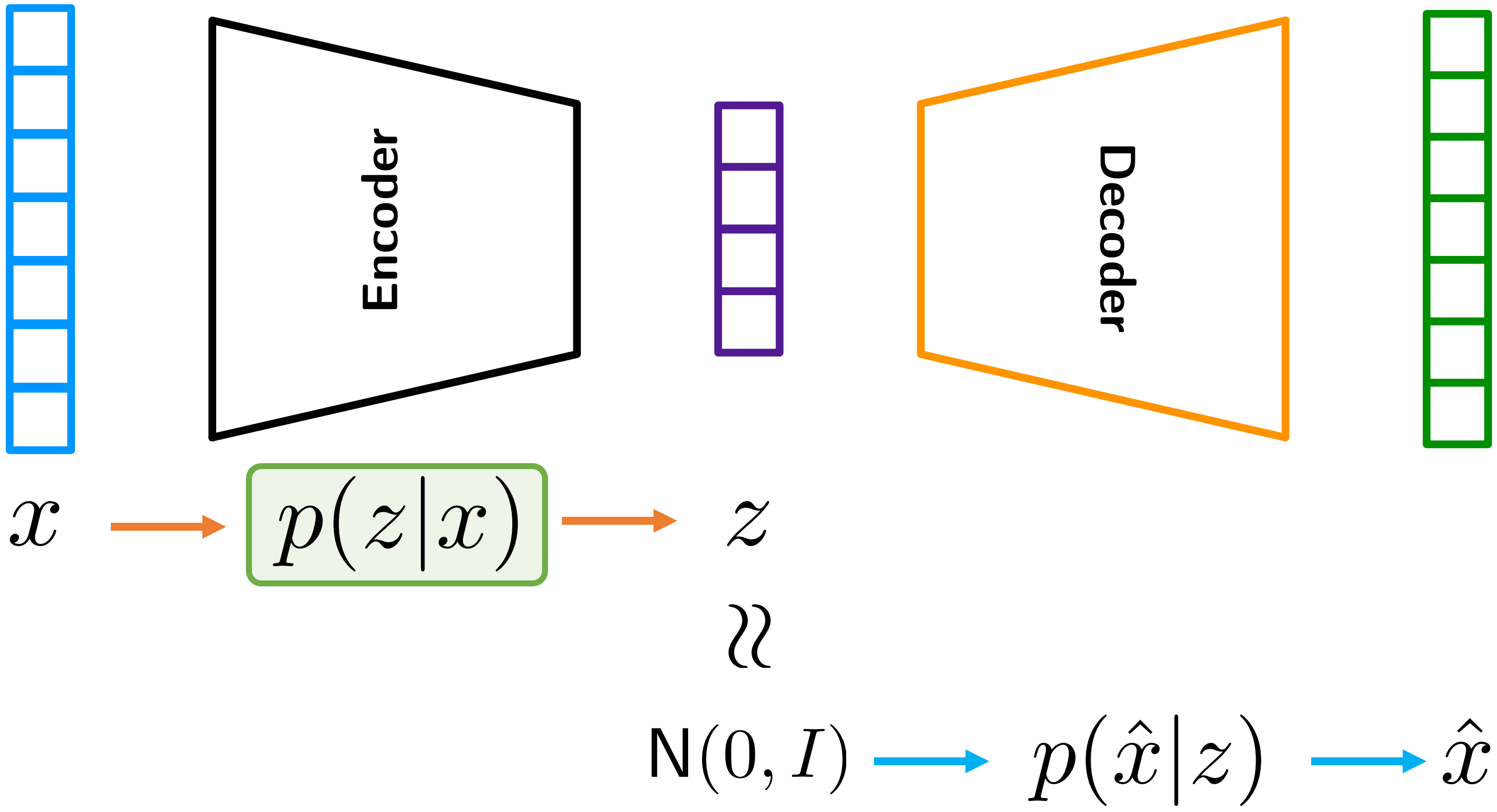


$$x \longrightarrow p(z|x) \longrightarrow z$$

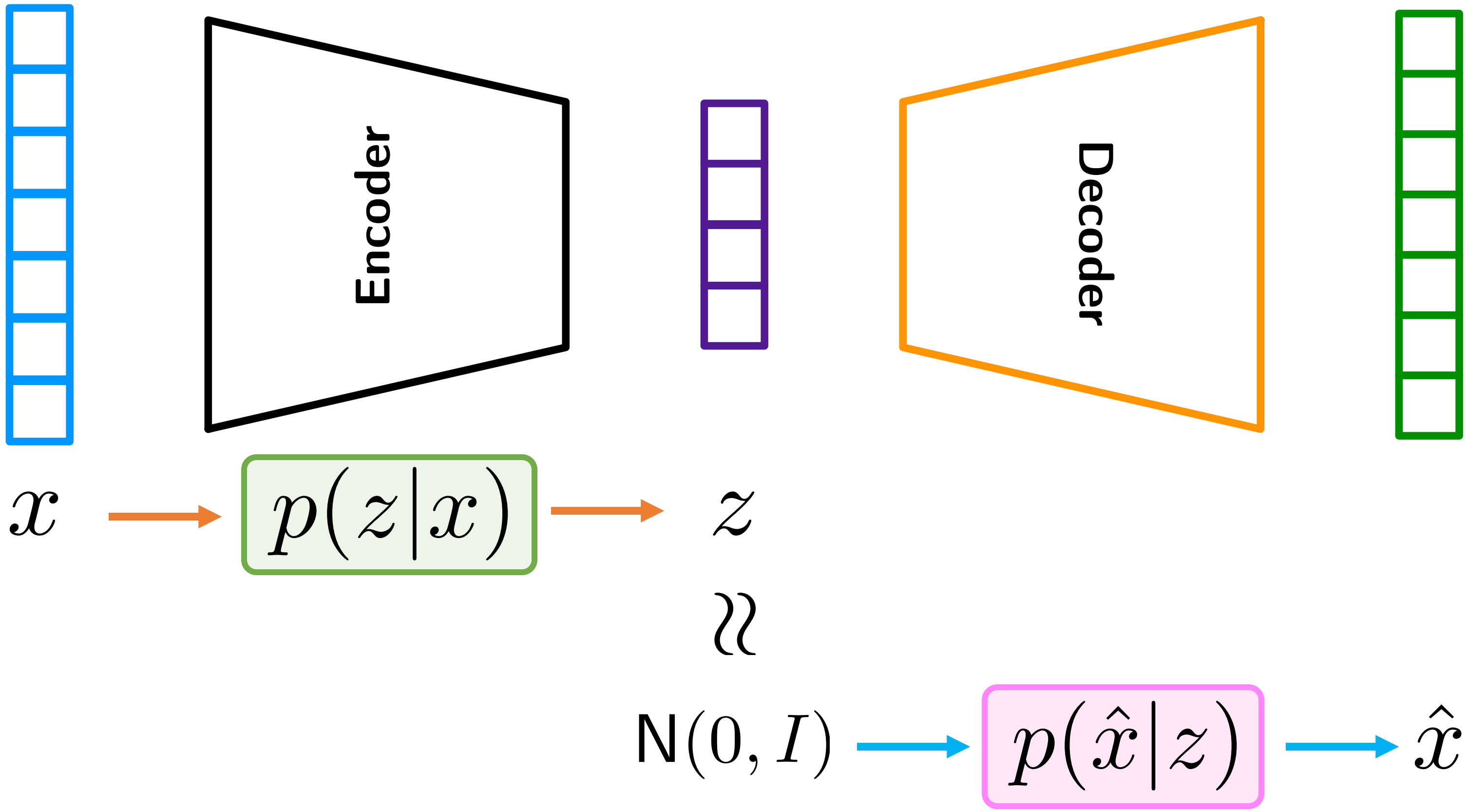
$$\wr\wr$$

$$\mathrm{N}(0, I) \longrightarrow p(\hat{x}|z) \longrightarrow \hat{x}$$

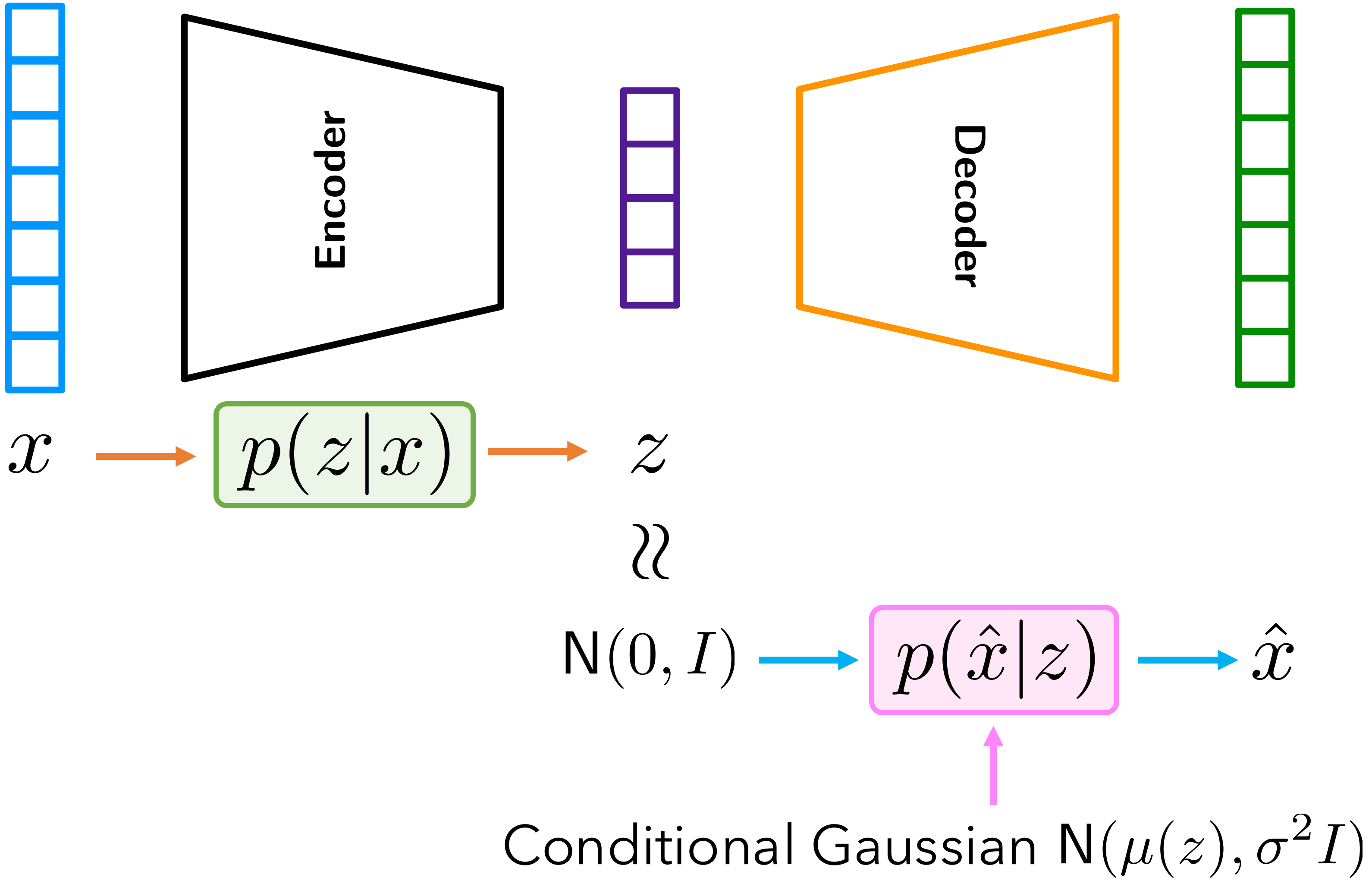
Dissemble VAE



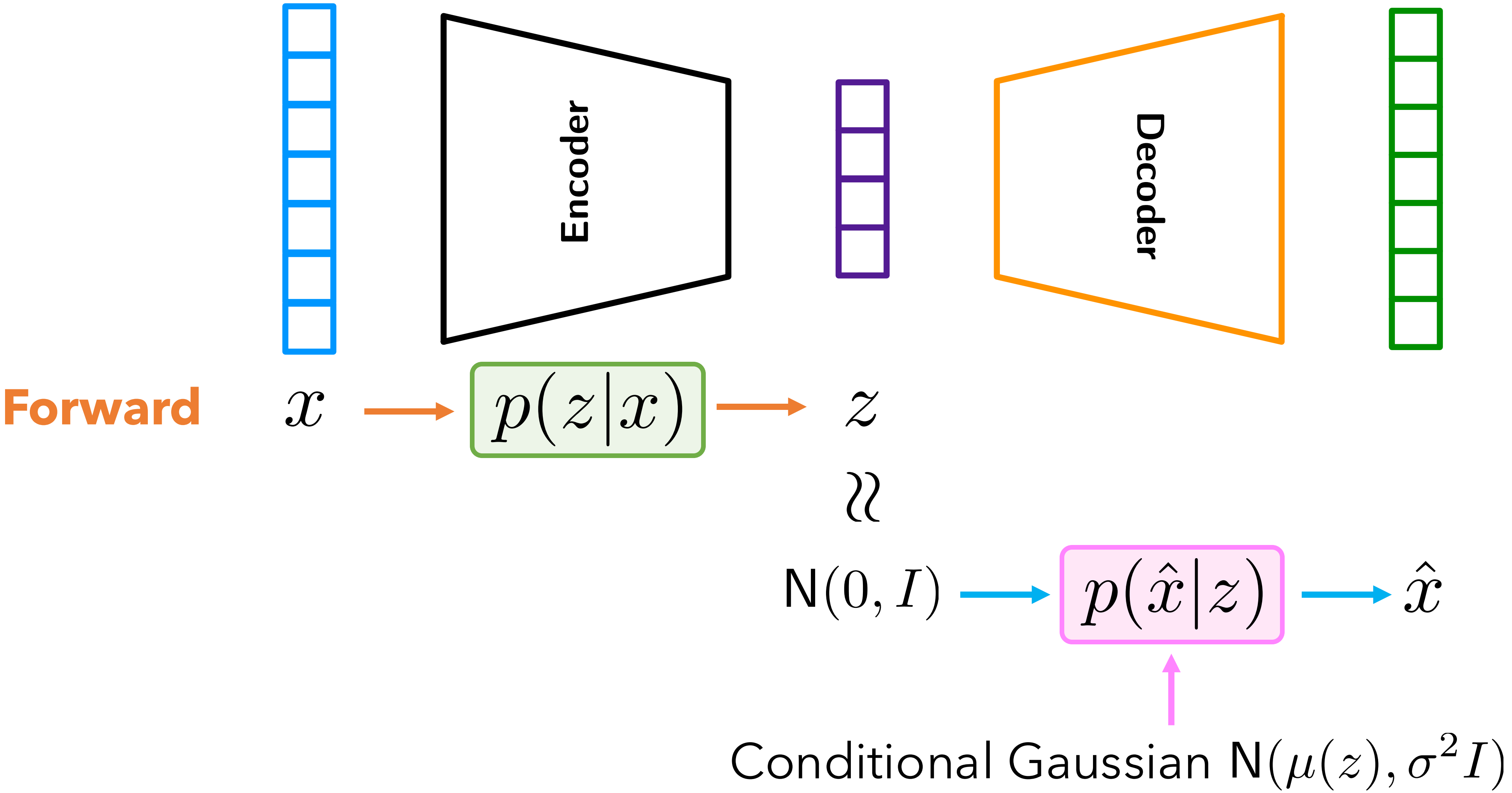
Dissemble VAE



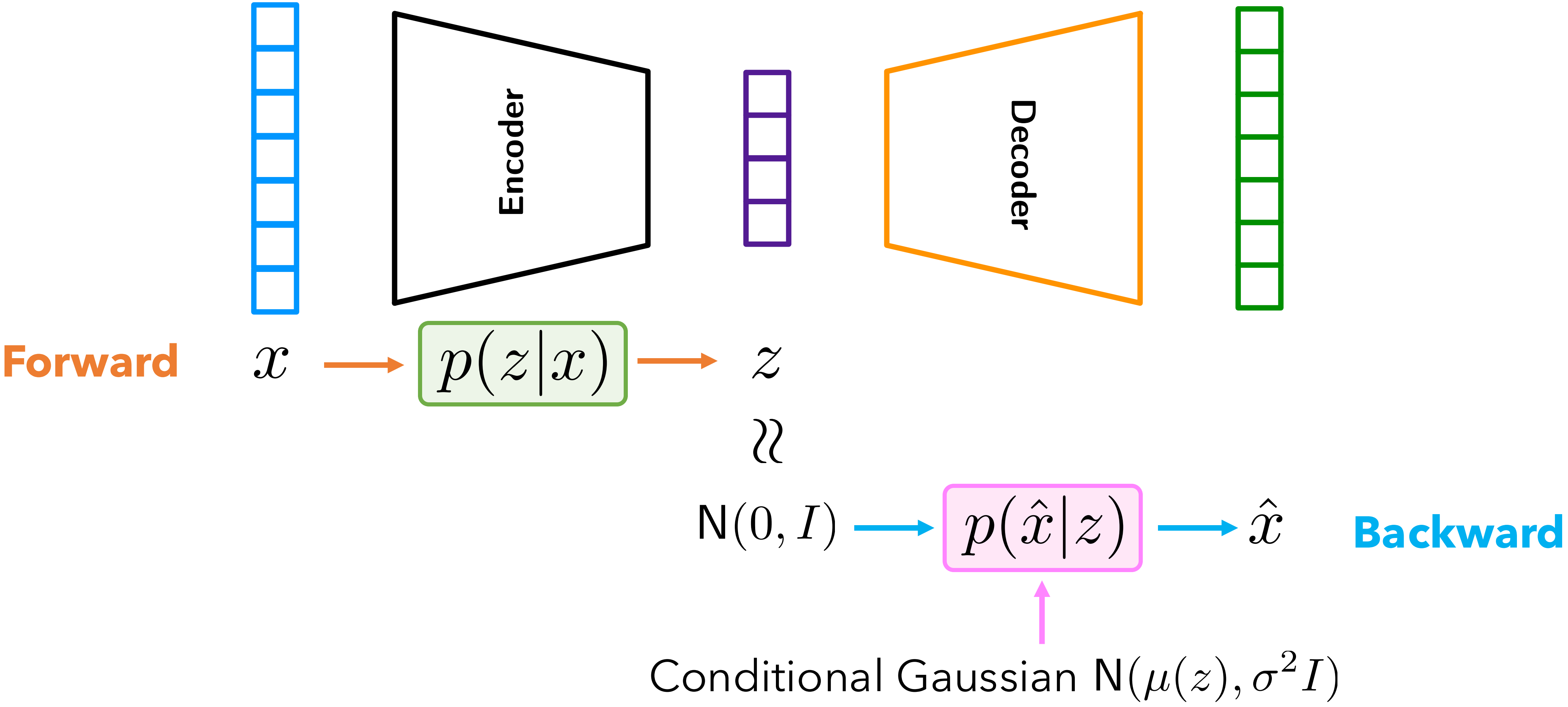
Dissemble VAE



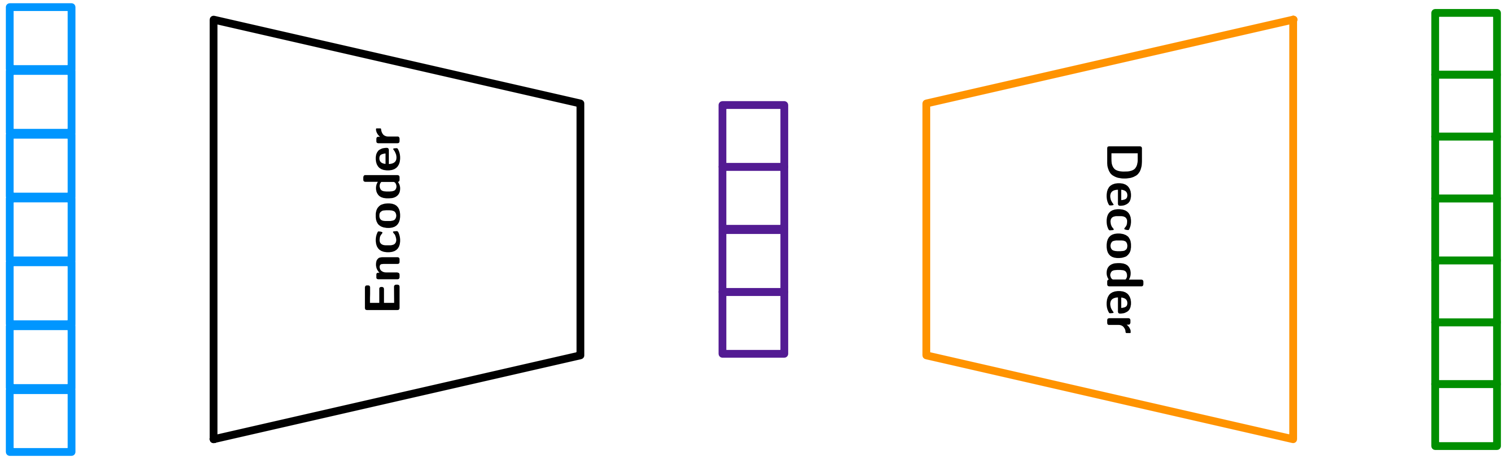
Dissemble VAE



Dissemble VAE



Dissemble VAE



Forward $x \rightarrow \boxed{p(z|x)} \rightarrow z$

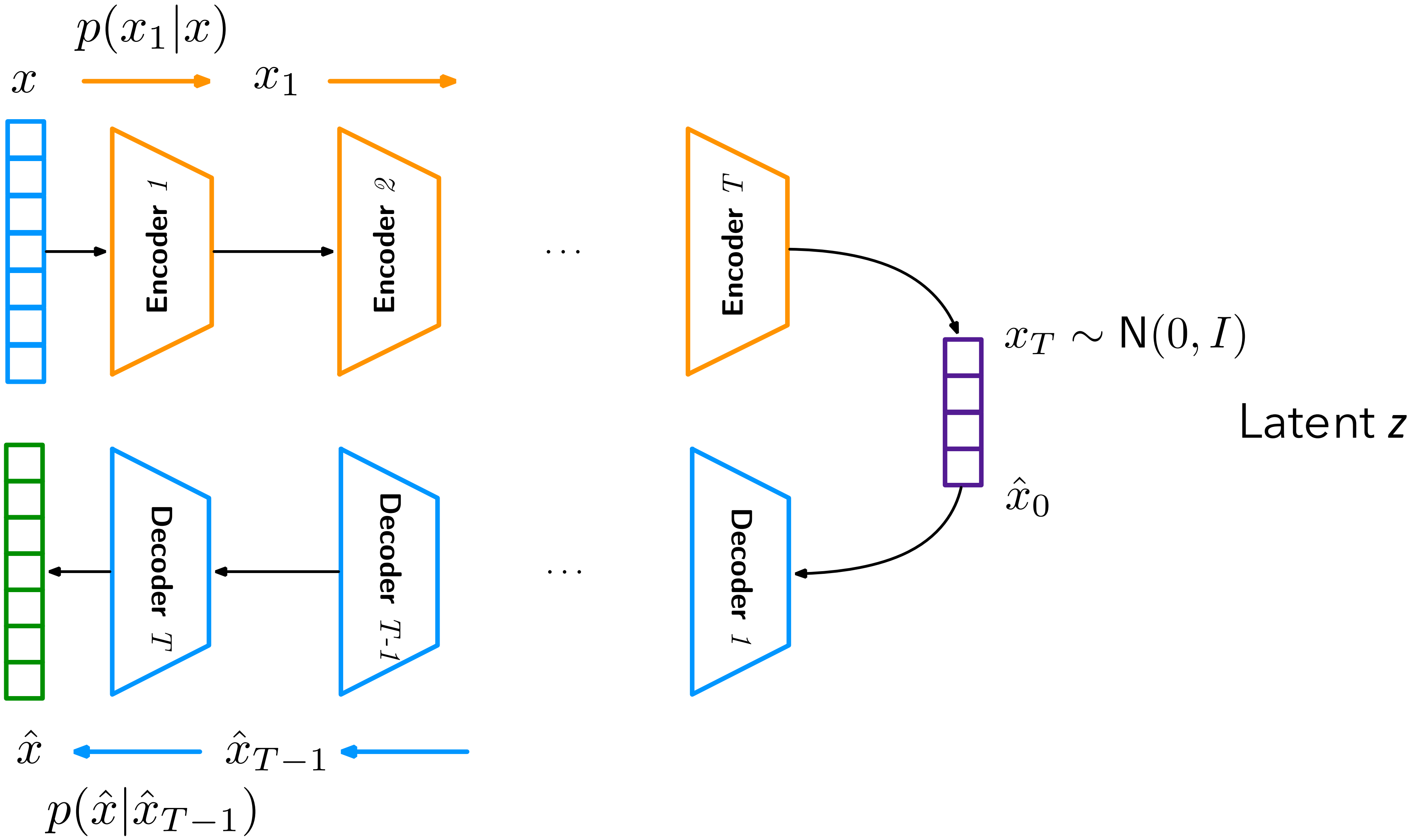
$\wr$

$\mathrm{N}(0, I) \rightarrow \boxed{p(\hat{x}|z)} \rightarrow \hat{x}$ **Backward**

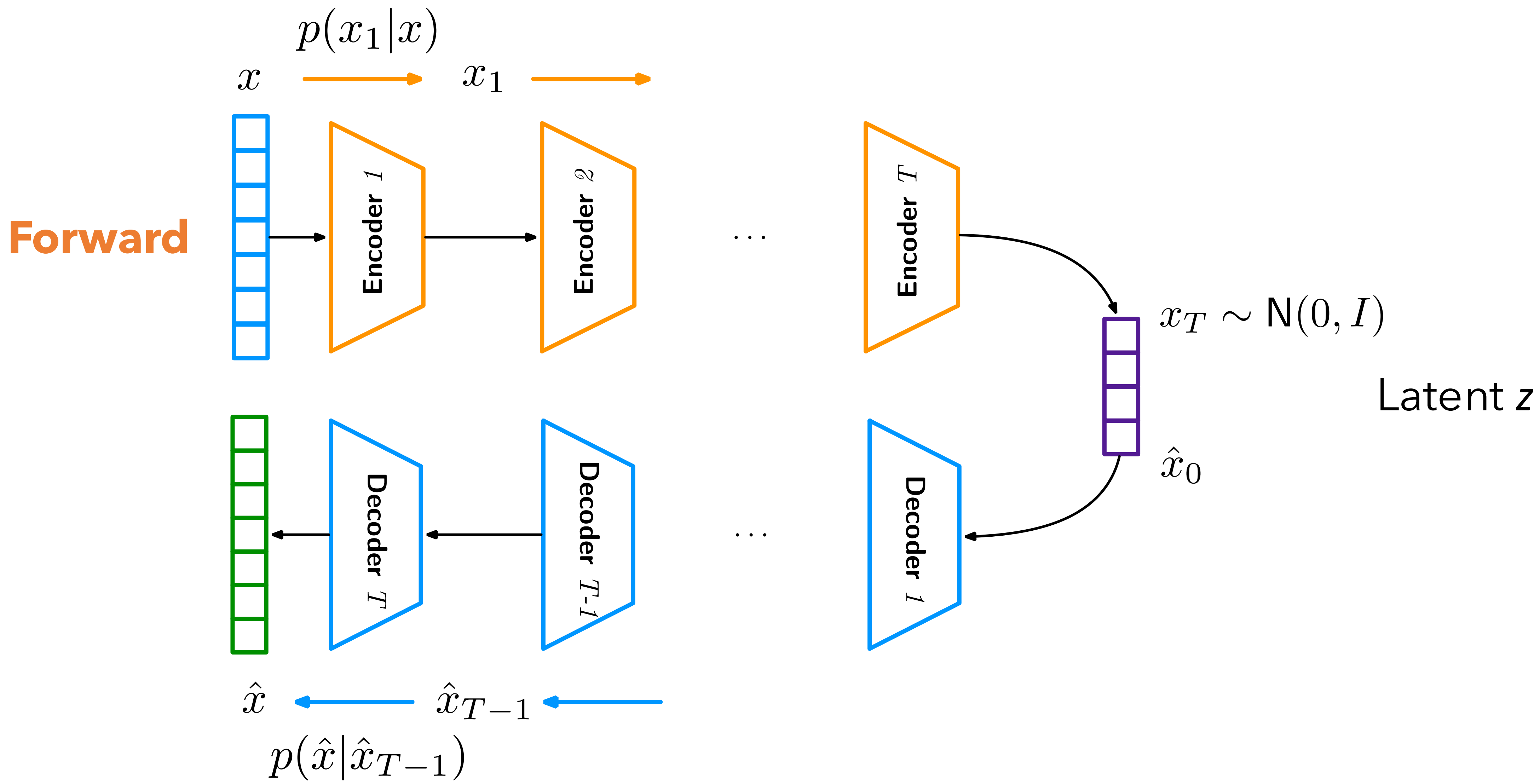
One-step transformation?

Conditional Gaussian $\mathrm{N}(\mu(z), \sigma^2 I)$

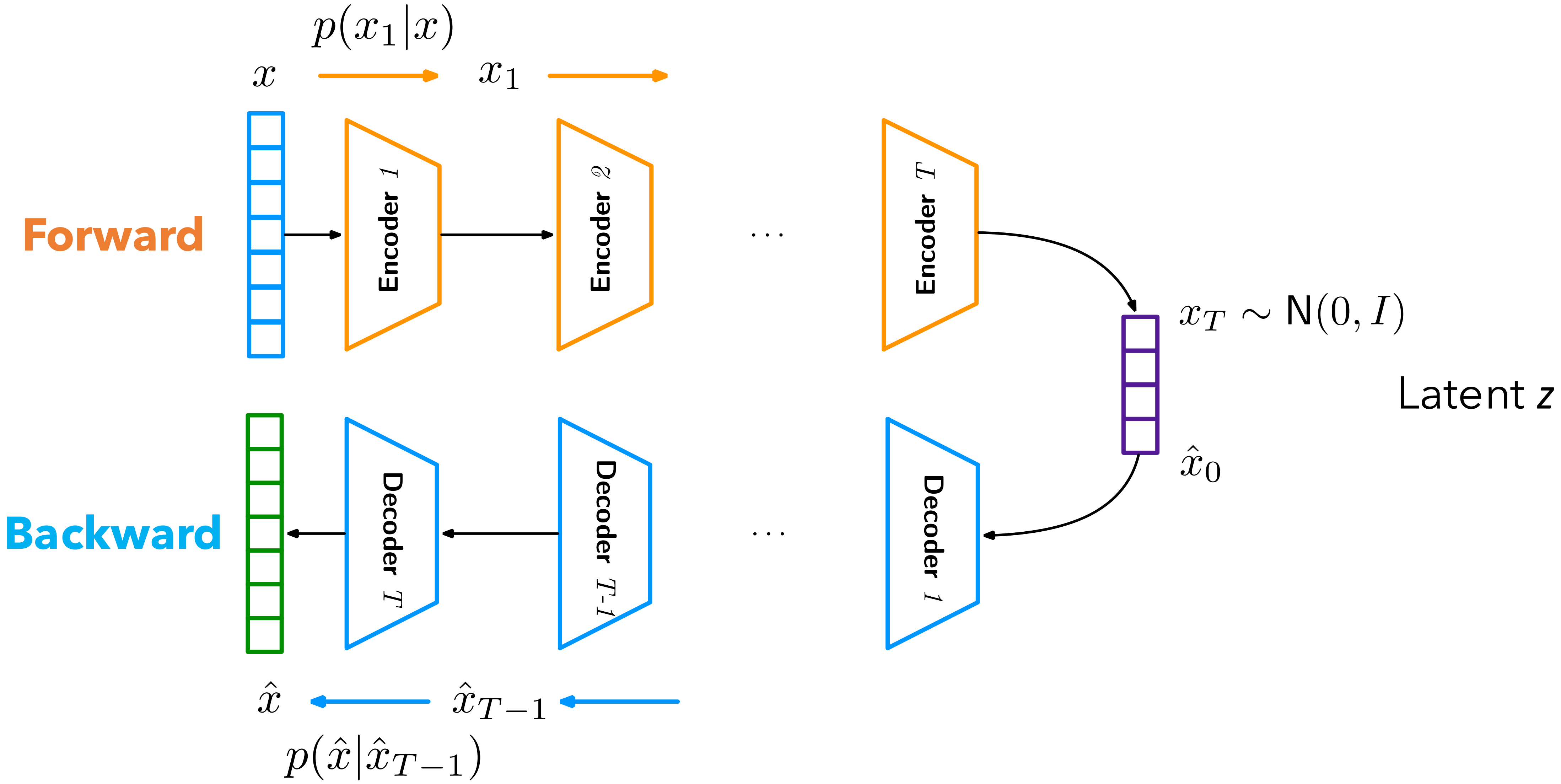
Let's Insert Some Intermediate Layers



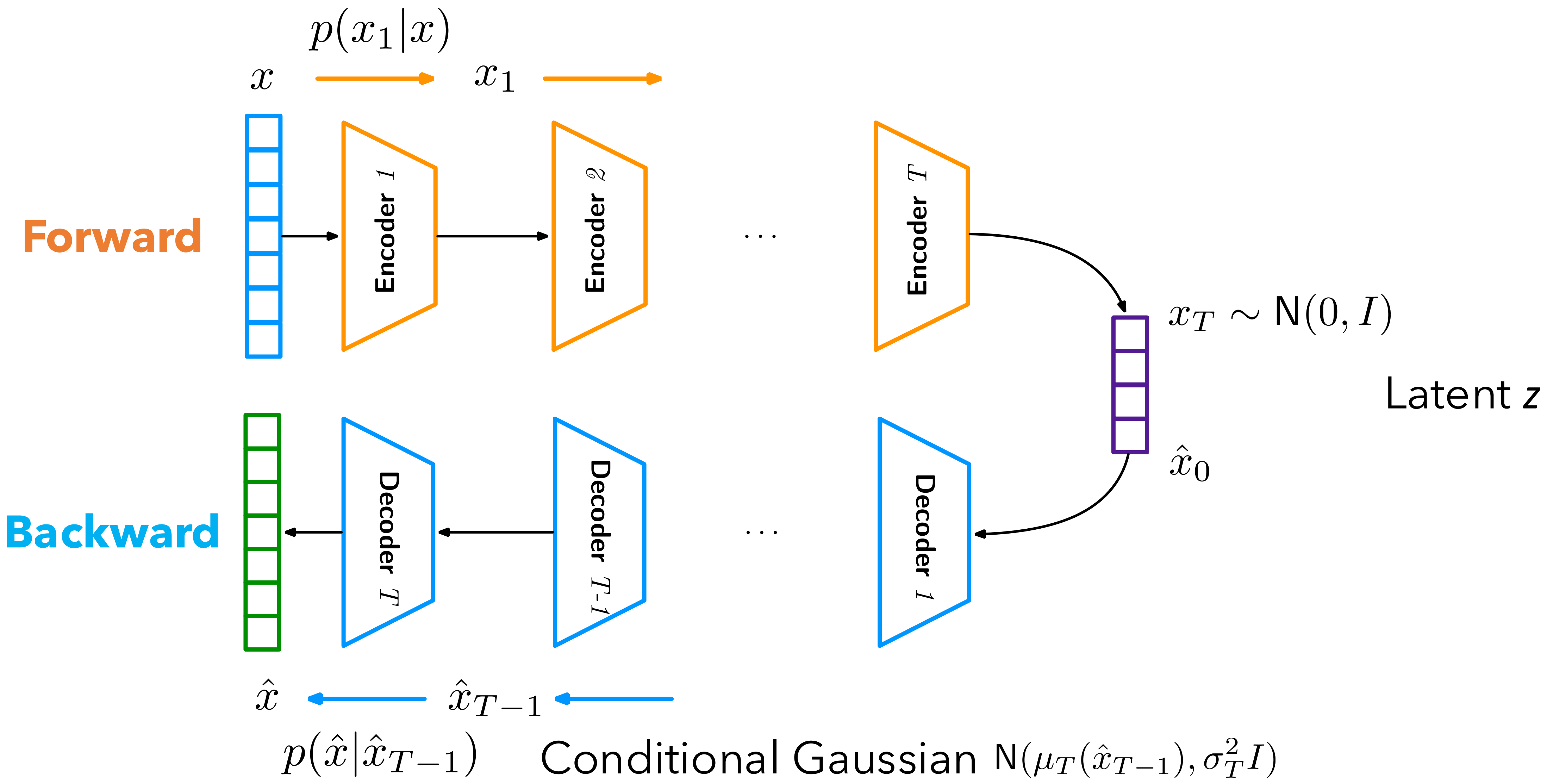
Let's Insert Some Intermediate Layers



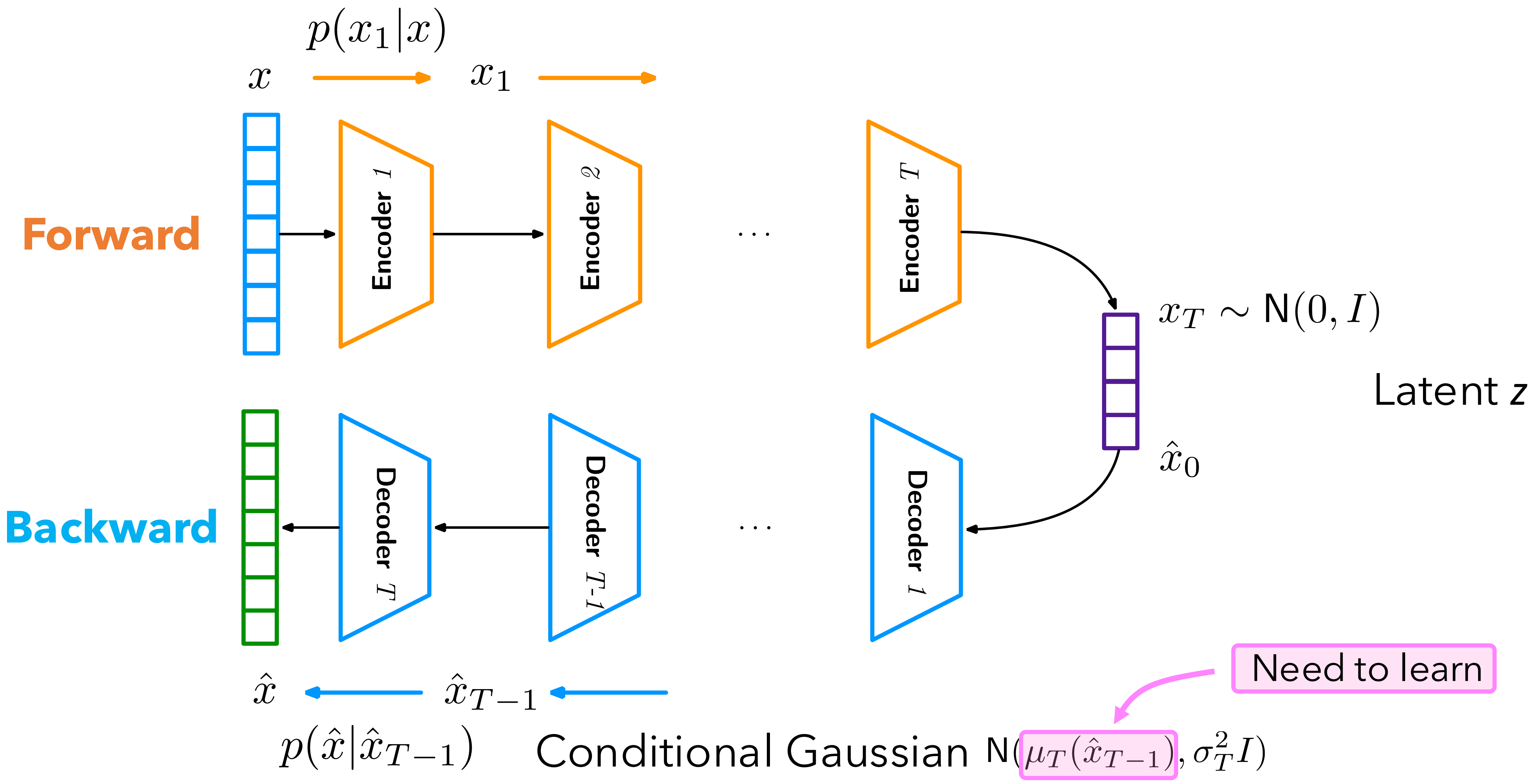
Let's Insert Some Intermediate Layers



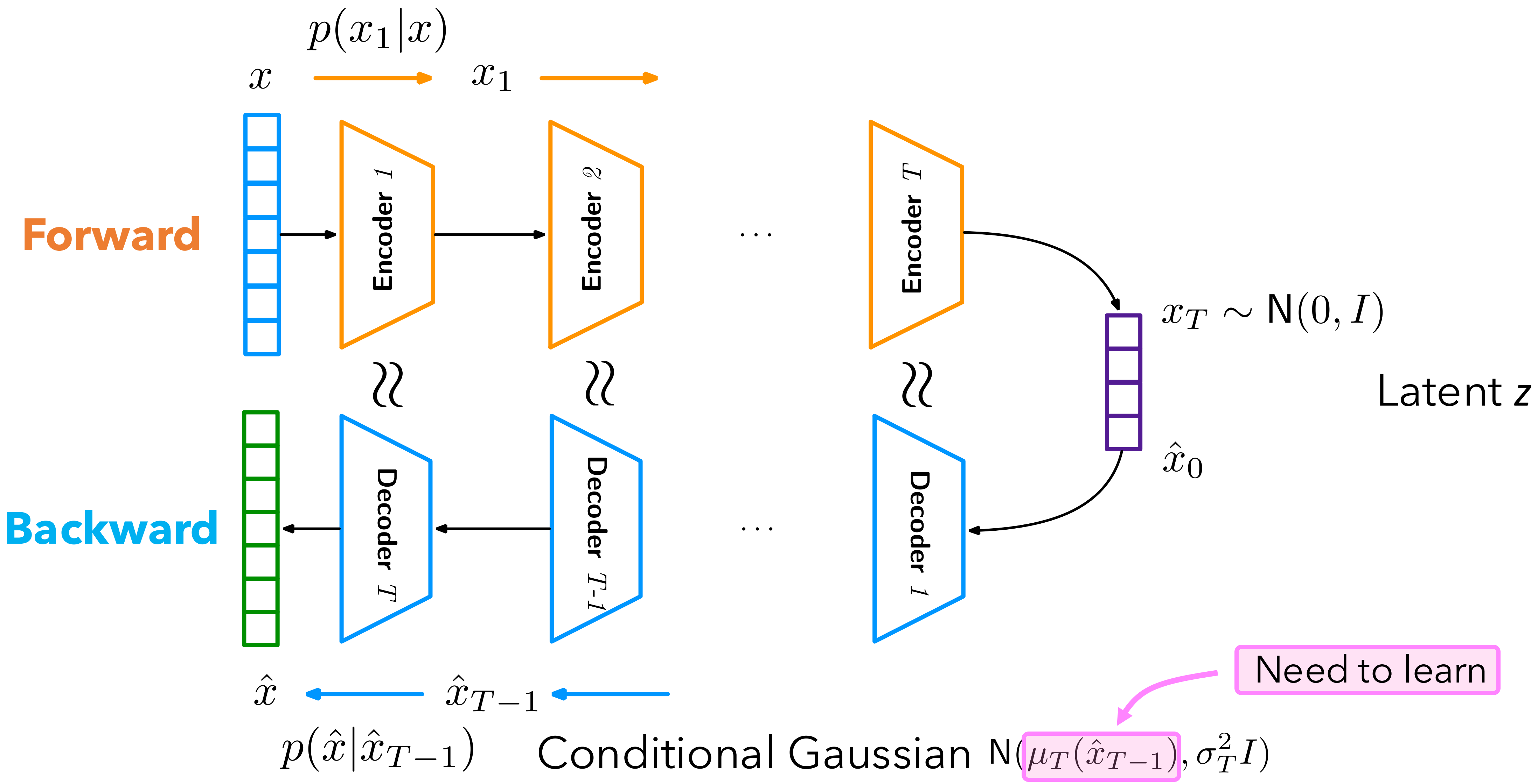
Let's Insert Some Intermediate Layers



Let's Insert Some Intermediate Layers



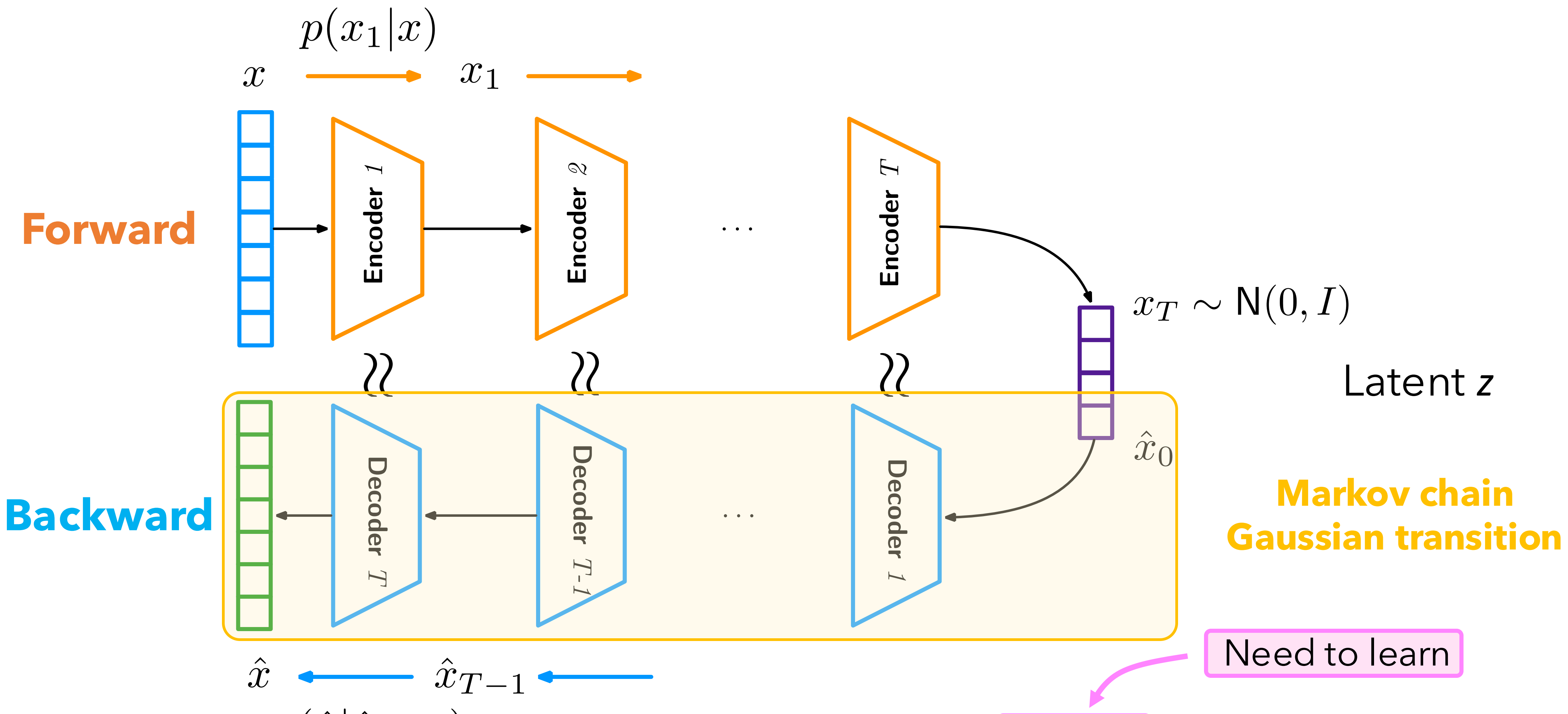
Let's Insert Some Intermediate Layers



Need to learn

Conditional Gaussian $\mathrm{N}(\boxed{\mu_T(\hat{x}_{T-1})}, \sigma_T^2 I)$

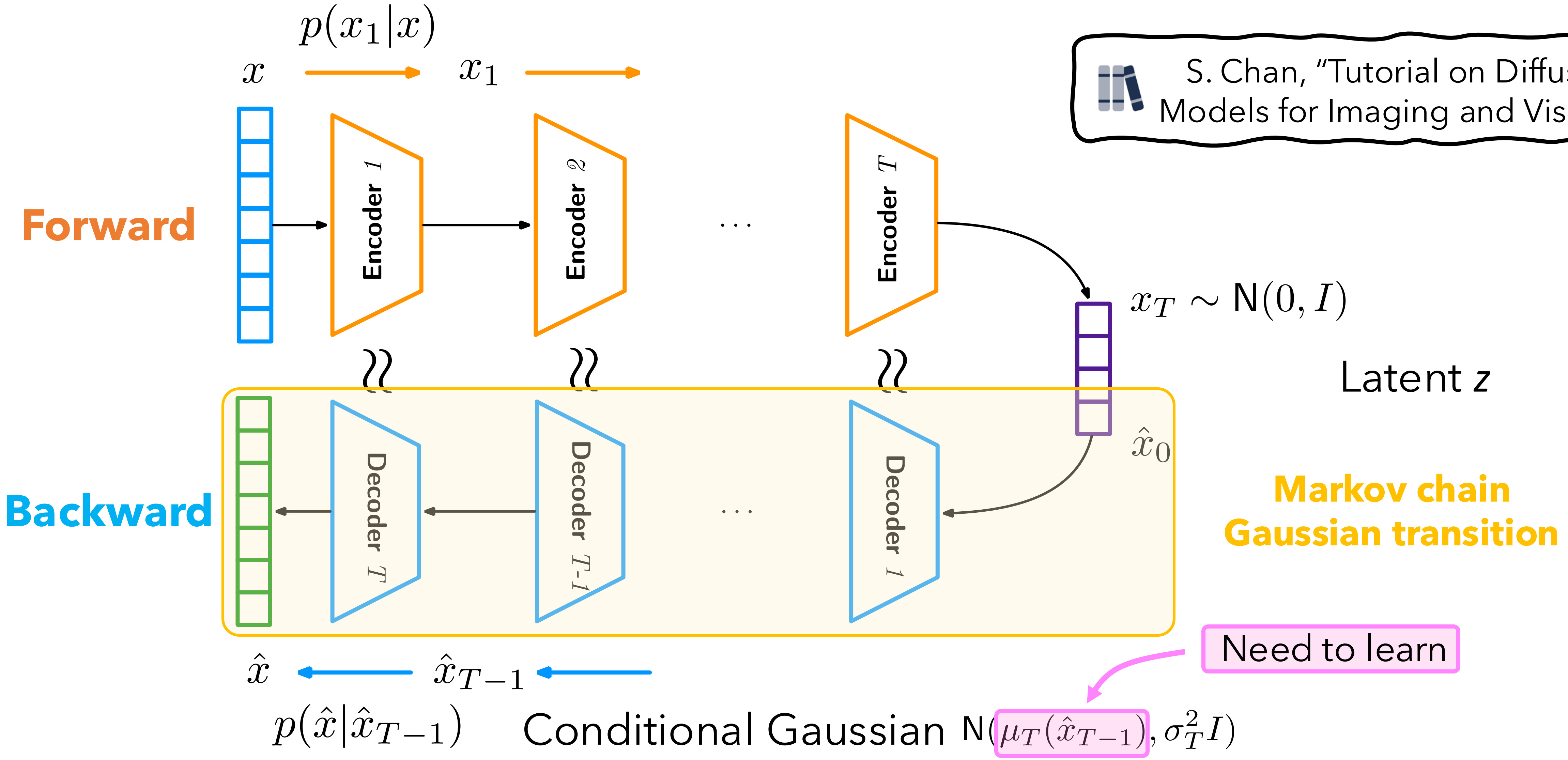
Let's Insert Some Intermediate Layers



$p(\hat{x}|\hat{x}_{T-1})$ Conditional Gaussian $\mathrm{N}(\boxed{\mu_T(\hat{x}_{T-1})}, \sigma_T^2 I)$

Let's Insert Some Intermediate Layers

S. Chan, "Tutorial on Diffusion Models for Imaging and Vision"



Conditional Gaussian $\mathrm{N}(\boxed{\mu_T(\hat{x}_{T-1})}, \sigma_T^2 I)$

Need to learn

A Revolution – Diffusion Model

- Sequential transformation in high-D

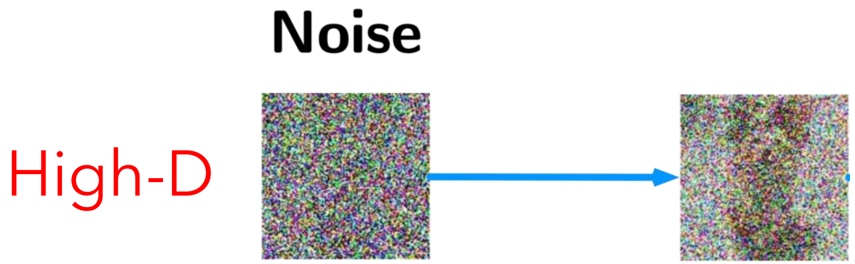
Noise

High-D

(Sohl-Dickstein et al., 2015)
(Song and Ermon, 2019)
(Ho et al., 2020)

A Revolution – Diffusion Model

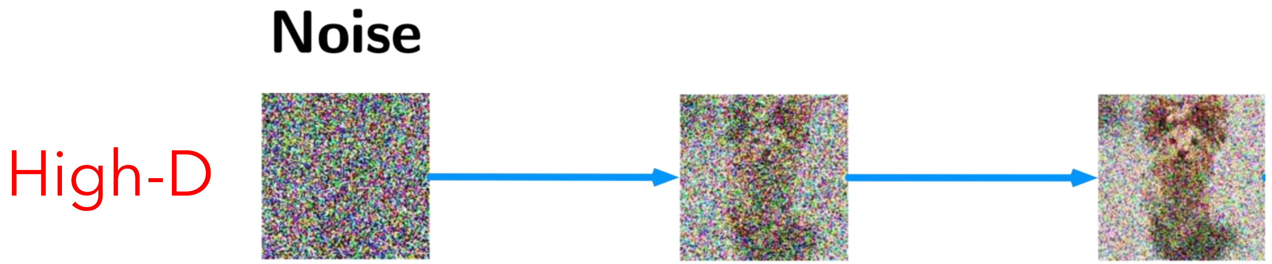
- Sequential transformation in high-D



(Sohl-Dickstein et al., 2015)
(Song and Ermon, 2019)
(Ho et al., 2020)

A Revolution – Diffusion Model

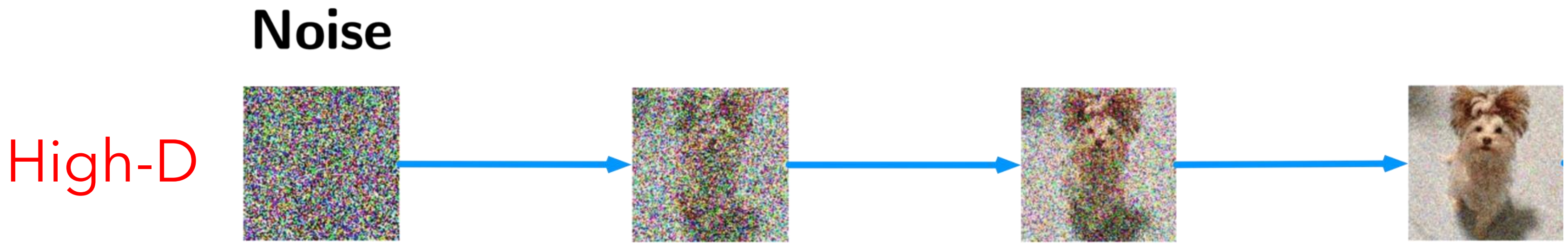
- Sequential transformation in high-D



(Sohl-Dickstein et al., 2015)
(Song and Ermon, 2019)
(Ho et al., 2020)

A Revolution – Diffusion Model

- Sequential transformation in high-D



(Sohl-Dickstein et al., 2015)
(Song and Ermon, 2019)
(Ho et al., 2020)

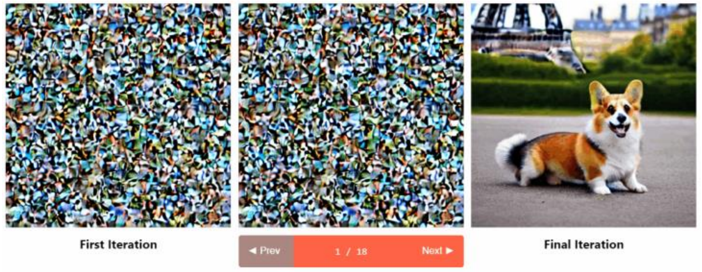
A Revolution – Diffusion Model

- Sequential transformation in high-D

(Sohl-Dickstein et al., 2015)
(Song and Ermon, 2019)
(Ho et al., 2020)

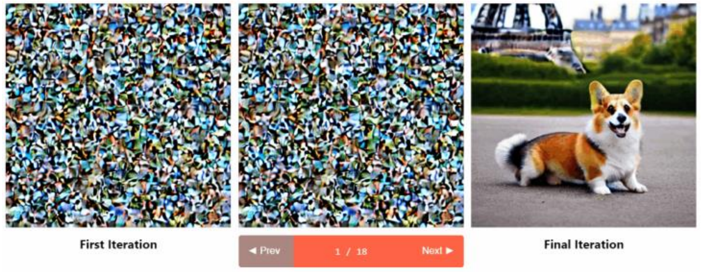
A Revolution – Diffusion Model

- Sequential transformation in high-D



(Sohl-Dickstein et al., 2015)
(Song and Ermon, 2019)
(Ho et al., 2020)

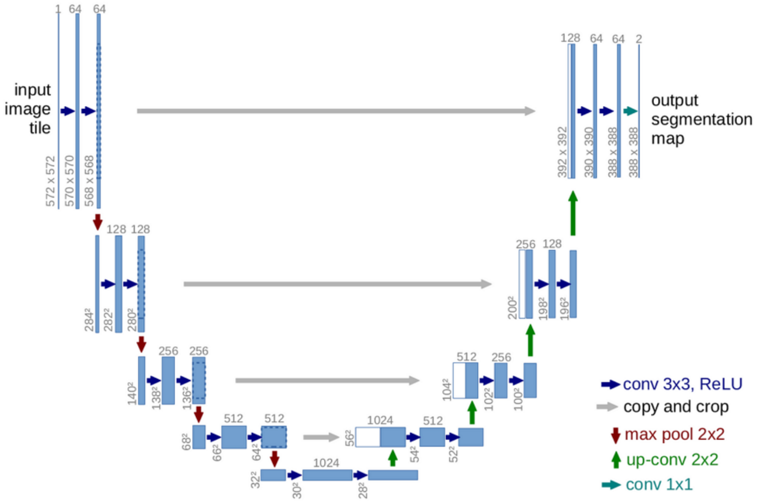
A Revolution – Diffusion Model



- Sequential transformation in high-D

Noise **Data**

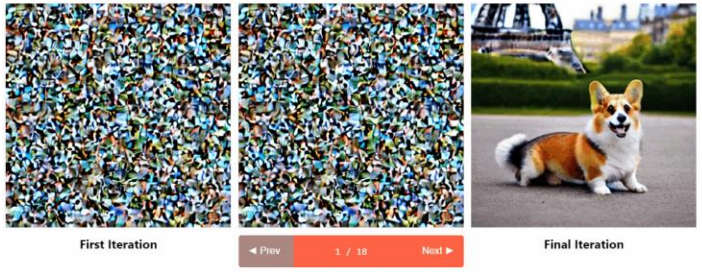
High-D



> 890M parameters
(Stable Diffusion)

(Sohl-Dickstein et al., 2015)
(Song and Ermon, 2019)
(Ho et al., 2020)

A Revolution – Diffusion Model

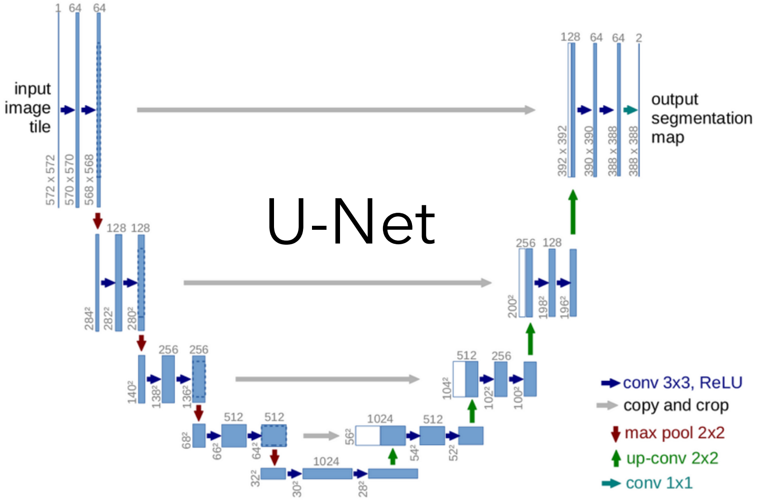


- Sequential transformation in high-D

Noise

Data

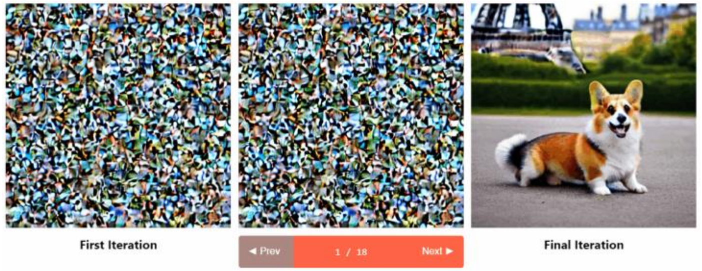
High-D



> 890M parameters
(Stable Diffusion)

(Sohl-Dickstein et al., 2015)
(Song and Ermon, 2019)
(Ho et al., 2020)

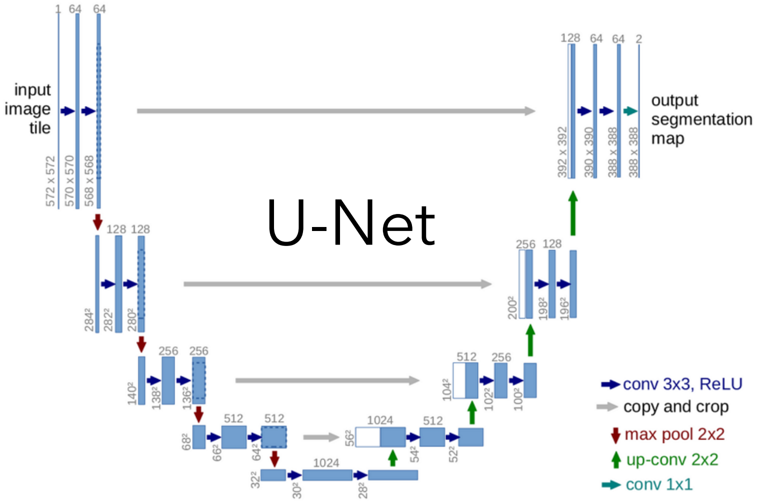
A Revolution – Diffusion Model



- Sequential transformation in high-D

Noise **Data**

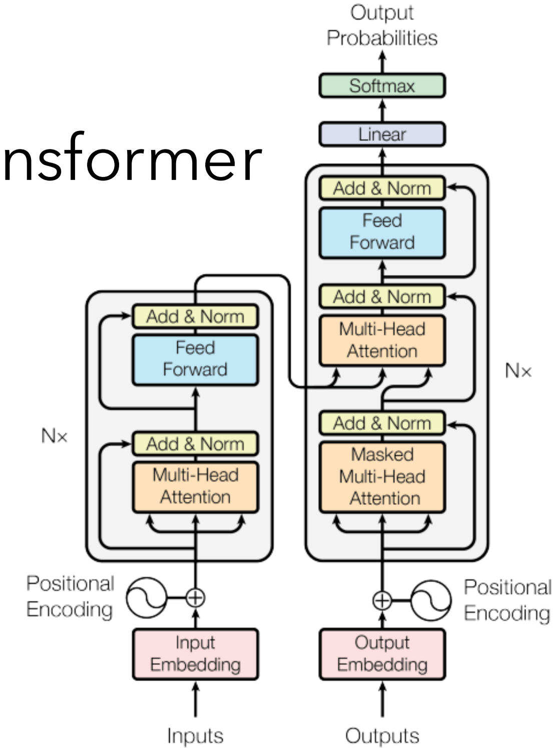
High-D



U-Net

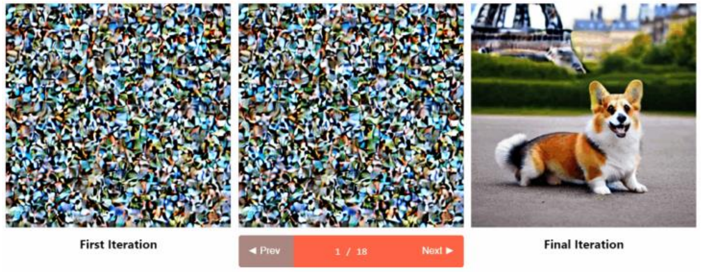
> 890M parameters
(Stable Diffusion)

Transformer



2015)
2019)
2020)

A Revolution – Diffusion Model

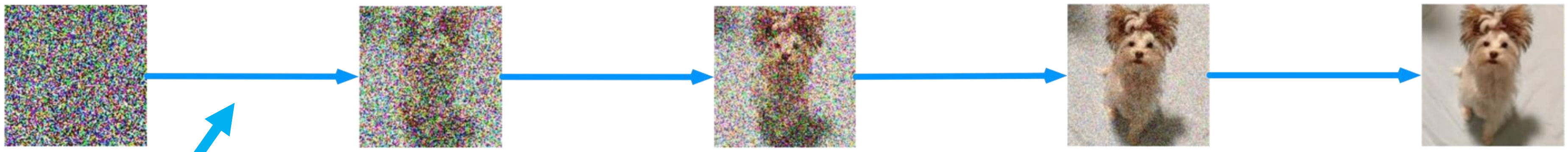


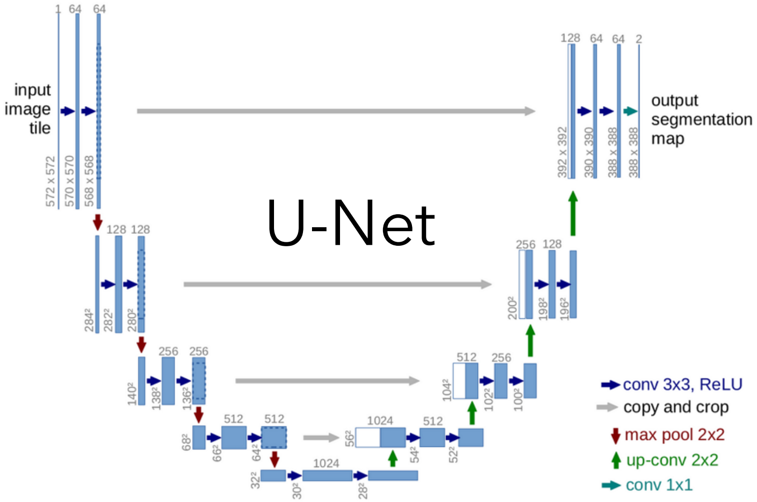
- Sequential transformation in high-D

Noise

Data

High-D

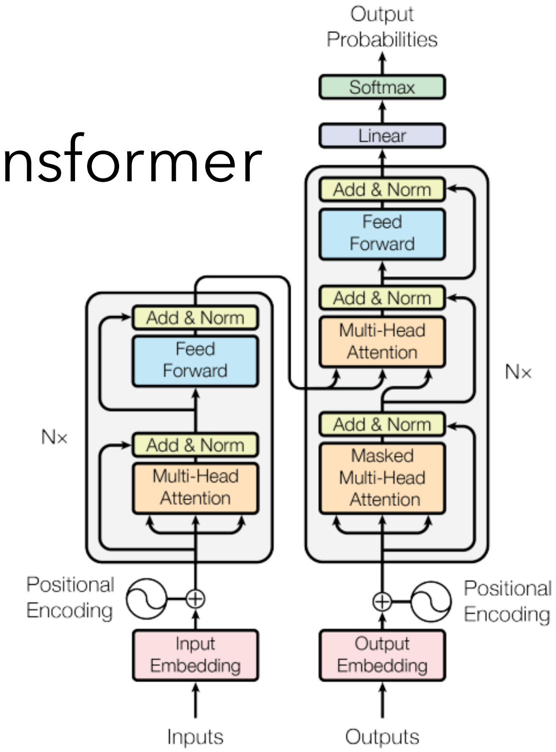




U-Net

$\mu_t(\cdot)$

Transformer



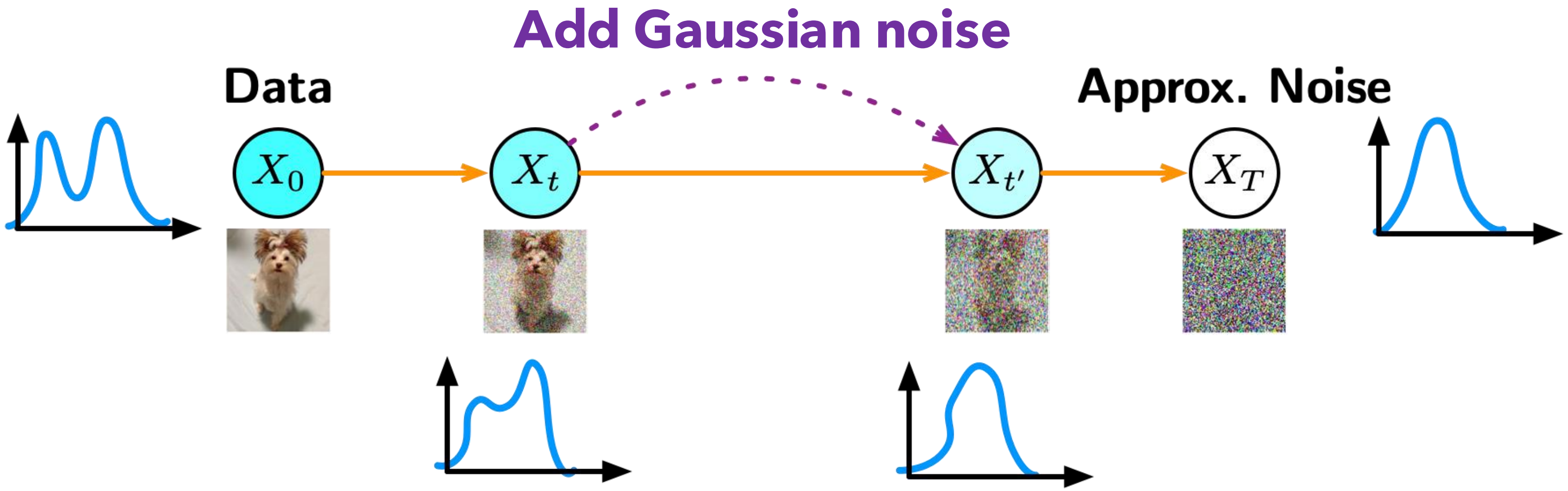
2015)
2019)
2020)

> 890M parameters
(Stable Diffusion)

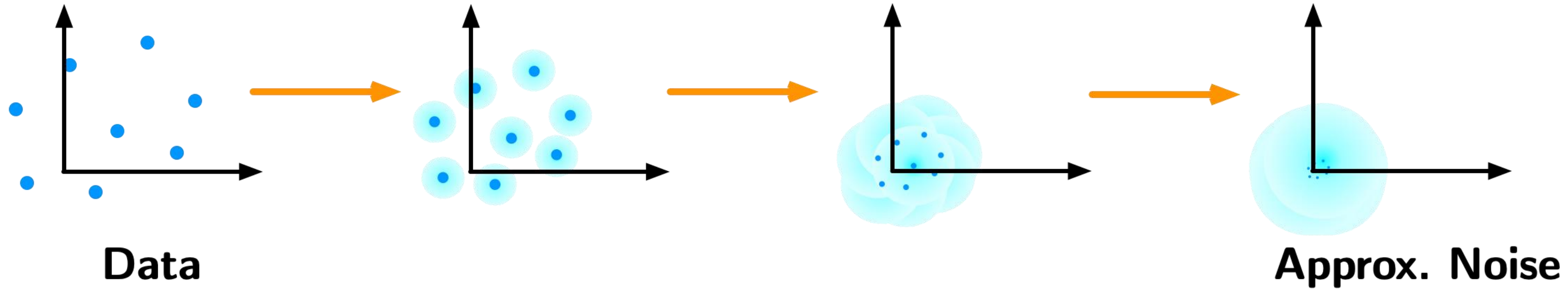
Forward Process – Noise Corruption

Insert infinitely many intermediate layers!

- Noise corruption process $\mathrm{d}X_t = -\frac{1}{2}X_t\mathrm{d}t + \mathrm{d}W_t$

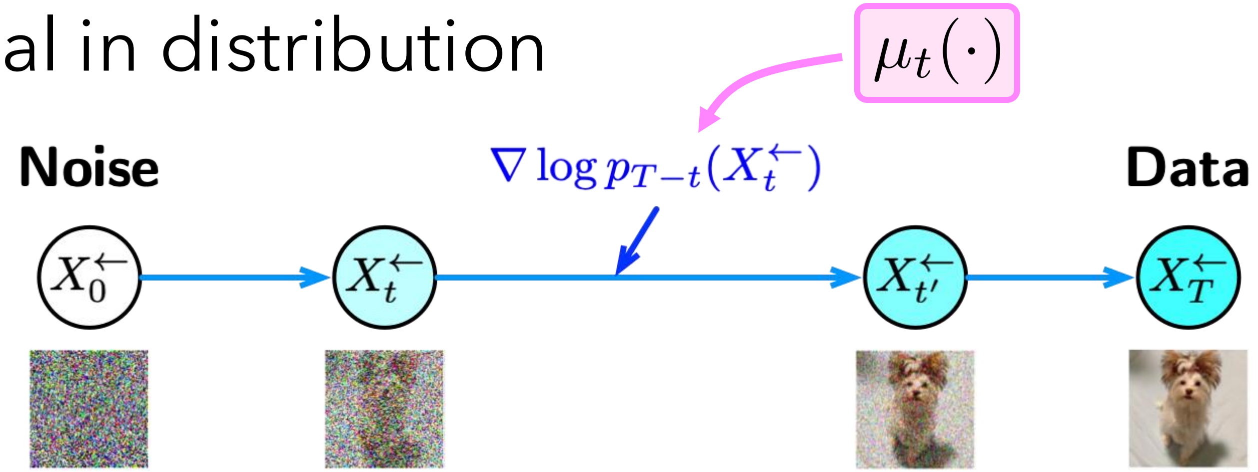


- The noise corruption



Backward Process – Sample Generation

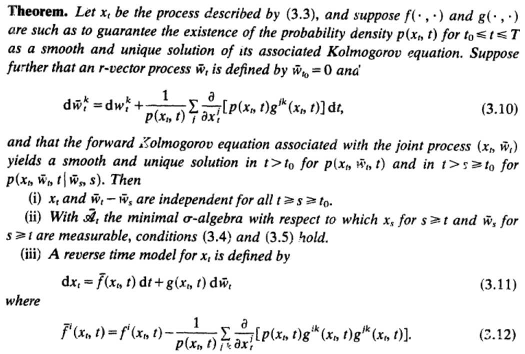
- Time reversal in distribution



- The math (Anderson, 1982; Haussmann and Pardoux, 1986)

Forward $$\mathrm{d}X_t = -\frac{1}{2}X_t \mathrm{d}t + \mathrm{d}W_t$$

Backward $$\mathrm{d}X_t^{\leftarrow} = \Big[\frac{1}{2}X_t^{\leftarrow} + \underbrace{\nabla \log p_{T-t}(X_t^{\leftarrow})}_{\text{Score Function}}\Big]\mathrm{d}t + \underbrace{\mathrm{d}\overline{W}_t}_{\text{Brownian}}$$



Theorem. *Let x_t be the process described by (3.3), and suppose $f(\cdot,\cdot)$ and $g(\cdot,\cdot)$ are such as to guarantee the existence of the probability density $p(x_t, t)$ for $t_0 \leqslant t \leqslant T$ as a smooth and unique solution of its associated Kolmogorov equation. Suppose further that an r-vector process $\bar{w}_t$ is defined by $\bar{w}_{t_0} = 0$ and*

$$\mathrm{d}\bar{w}_t^k = \mathrm{d}w_t^k + \frac{1}{p(x_t, t)} \sum_j \frac{\partial}{\partial x_t^j}[p(x_t, t) g^{jk}(x_t, t)]\,\mathrm{d}t, \tag{3.10}$$

and that the forward Kolmogorov equation associated with the joint process $(x_t, \bar{w}_t)$ yields a smooth and unique solution in $t > t_0$ for $p(x_t, \bar{w}_t, t)$ and in $t > s \geqslant t_0$ for $p(x_t, \bar{w}_t, t \mid \bar{w}_s, s)$. Then

(i) *x_t and $\bar{w}_t - \bar{w}_s$ are independent for all $t \geqslant s \geqslant t_0$.*

(ii) *With $\bar{\mathscr{A}}_t$ the minimal σ-algebra with respect to which x_s for $s \geqslant t$ and $\bar{w}_s$ for $s \geqslant t$ are measurable, conditions (3.4) and (3.5) hold.*

(iii) *A reverse time model for x_t is defined by*

$$\mathrm{d}x_t = \bar{f}(x_t, t)\,\mathrm{d}t + g(x_t, t)\,\mathrm{d}\bar{w}_t \tag{3.11}$$

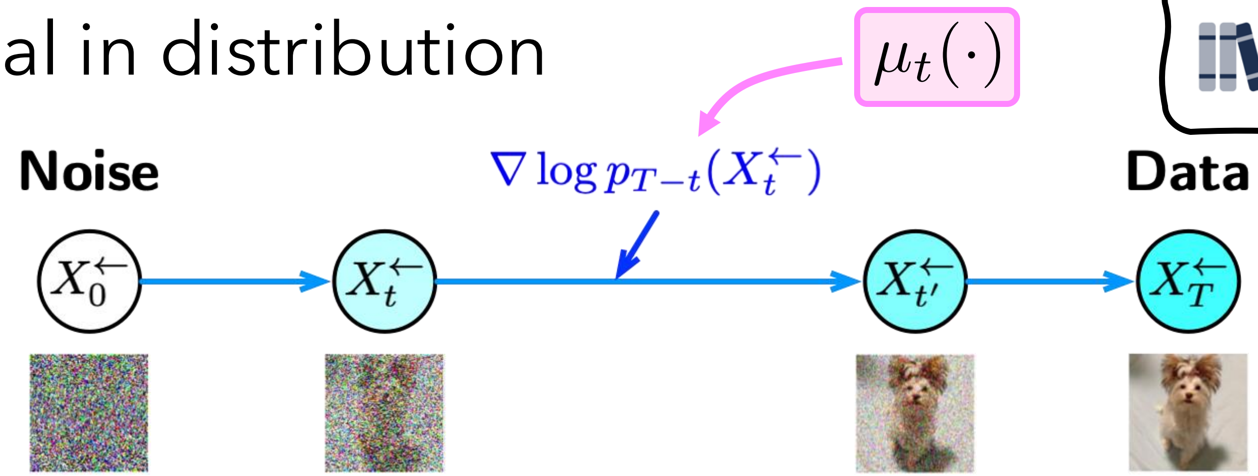
where

$$\bar{f}^i(x_t, t) = f^i(x_t, t) - \frac{1}{p(x_t, t)} \sum_{j,k} \frac{\partial}{\partial x_t^j}[p(x_t, t) g^{ik}(x_t, t) g^{jk}(x_t, t)]. \tag{3.12}$$

Backward Process – Sample Generation

> Tang & Zhao, “Score-based Diffusion Models via SDE”

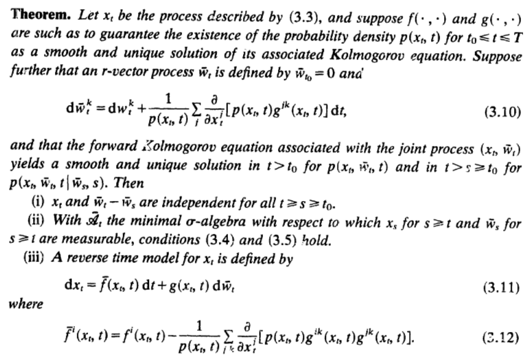
- Time reversal in distribution



- The math (Anderson, 1982; Haussmann and Pardoux, 1986)

Forward $$\mathrm{d}X_t = -\frac{1}{2}X_t \mathrm{d}t + \mathrm{d}W_t$$

Backward $$\mathrm{d}X_t^{\leftarrow} = \left[\frac{1}{2}X_t^{\leftarrow} + \underbrace{\boxed{\nabla \log p_{T-t}(X_t^{\leftarrow})}}_{\text{Score Function}}\right]\mathrm{d}t + \underbrace{\mathrm{d}\overline{W}_t}_{\text{Brownian}}$$



Theorem. *Let x_t be the process described by (3.3), and suppose $f(\cdot,\cdot)$ and $g(\cdot,\cdot)$ are such as to guarantee the existence of the probability density $p(x_t, t)$ for $t_0 \leqslant t \leqslant T$ as a smooth and unique solution of its associated Kolmogorov equation. Suppose further that an r-vector process $\bar{w}_t$ is defined by $\bar{w}_{t_0} = 0$ and*

$$\mathrm{d}\bar{w}_t^k = \mathrm{d}w_t^k + \frac{1}{p(x_t, t)}\sum_j \frac{\partial}{\partial x_t^j}[p(x_t, t)g^{jk}(x_t, t)]\,\mathrm{d}t, \qquad (3.10)$$

and that the forward Kolmogorov equation associated with the joint process $(x_t, \bar{w}_t)$ yields a smooth and unique solution in $t > t_0$ for $p(x_t, \bar{w}_t, t)$ and in $t > s \geqslant t_0$ for $p(x_t, \bar{w}_t, t \mid \bar{w}_s, s)$. Then

(i) *x_t and $\bar{w}_t - \bar{w}_s$ are independent for all $t \geqslant s \geqslant t_0$.*

(ii) *With $\bar{\mathscr{A}}_t$ the minimal σ-algebra with respect to which x_s for $s \geqslant t$ and $\bar{w}_s$ for $s \geqslant t$ are measurable, conditions (3.4) and (3.5) hold.*

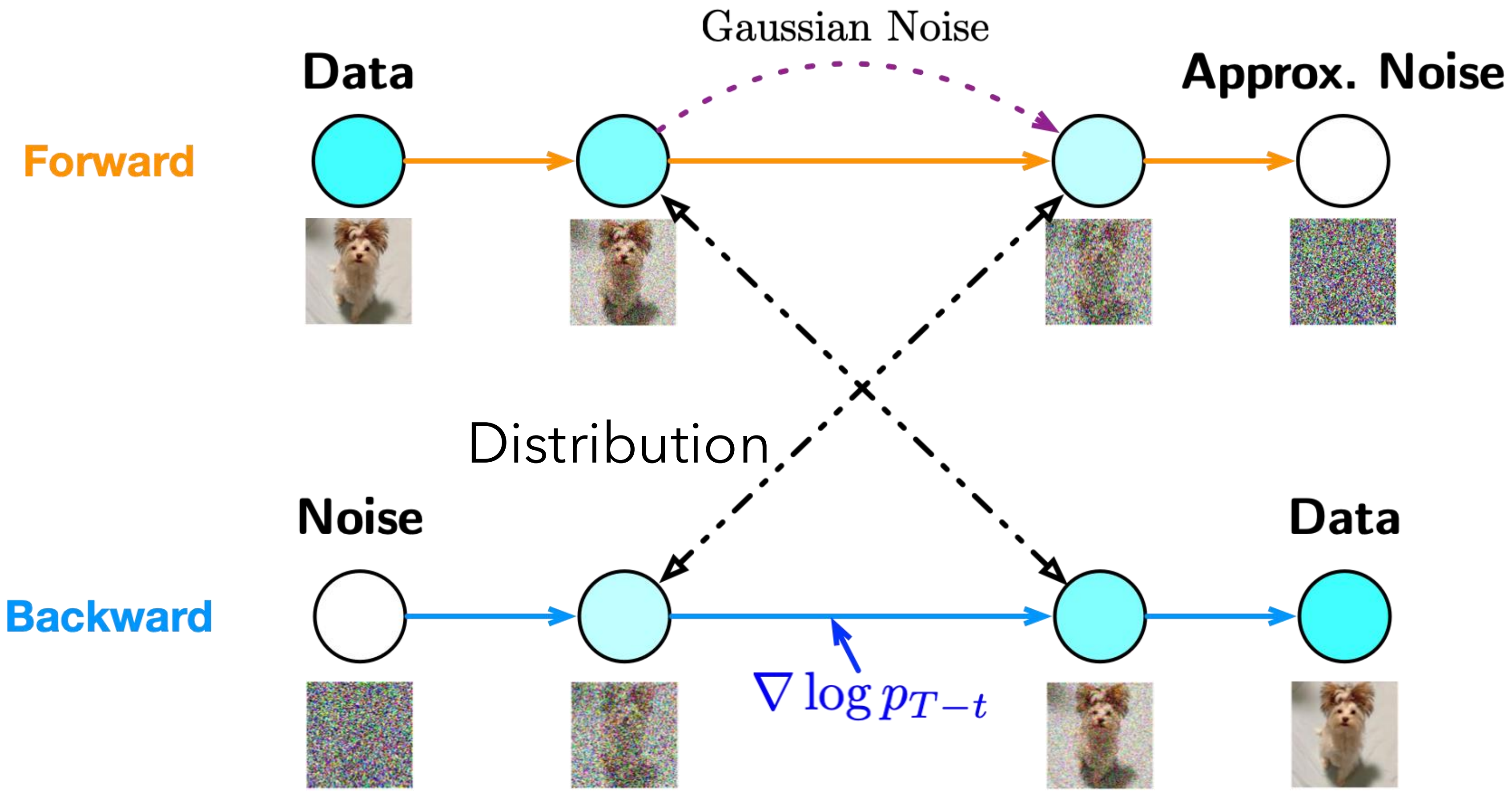
(iii) *A reverse time model for x_t is defined by*

$$\mathrm{d}x_t = \bar{f}(x_t, t)\,\mathrm{d}t + g(x_t, t)\,\mathrm{d}\bar{w}_t \qquad (3.11)$$

where

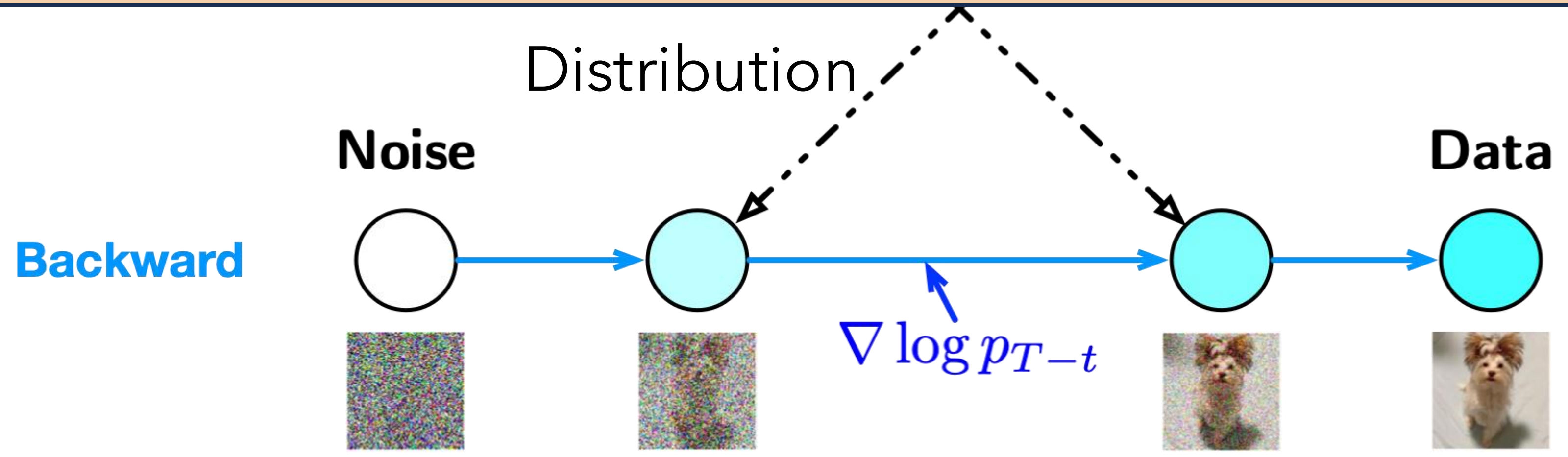
$$\bar{f}^i(x_t, t) = f^i(x_t, t) - \frac{1}{p(x_t, t)}\sum_{j,k}\frac{\partial}{\partial x_t^j}[p(x_t, t)g^{ik}(x_t, t)g^{jk}(x_t, t)]. \qquad (3.12)$$

Forward and Backward Coupling



Forward and Backward Coupling

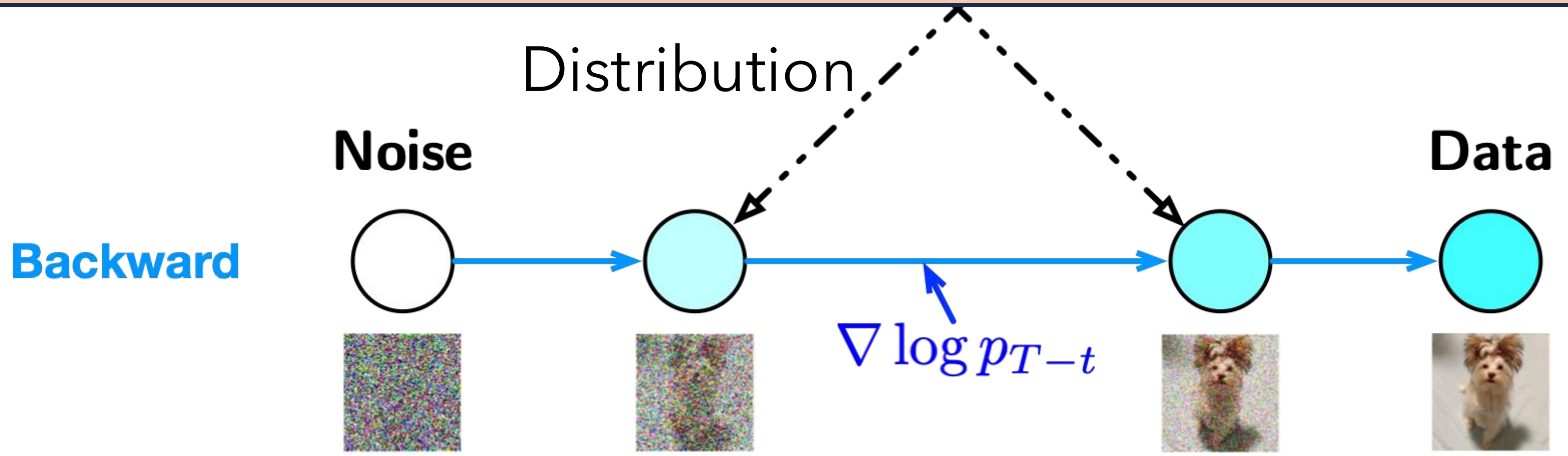
Training $\int_0^T \mathbb{E}_{x_t}\left[\|\nabla \log p_t(x_t) - s(x_t, t)\|_2^2\right] \mathrm{d}t$



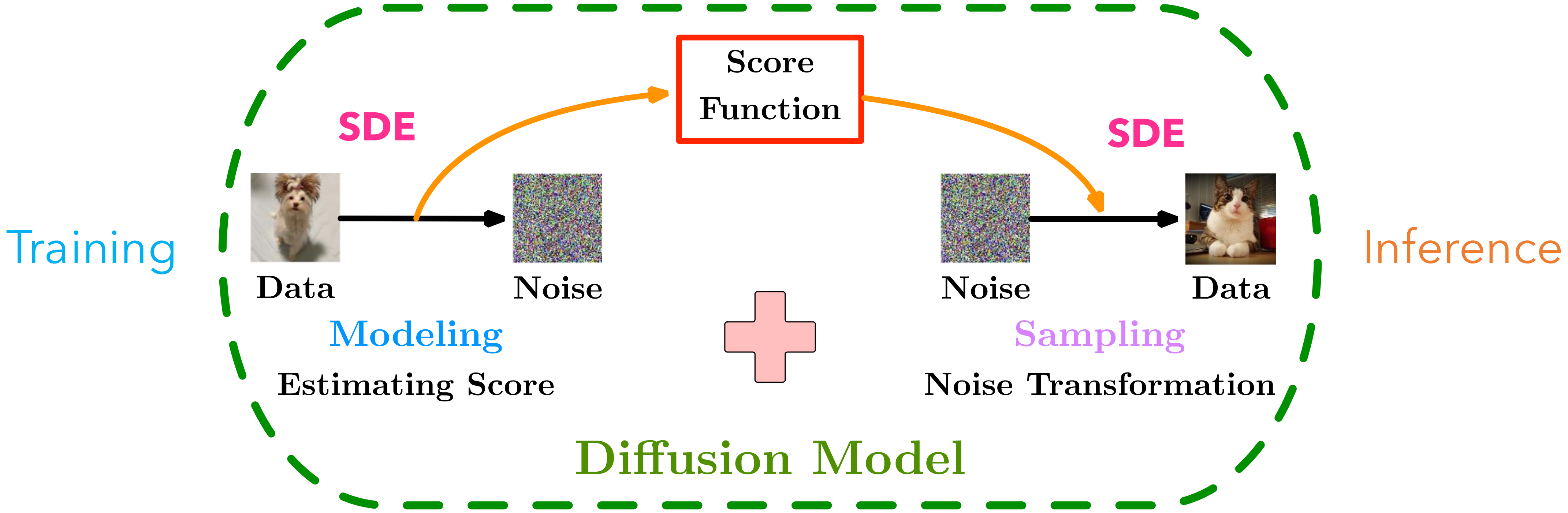
Forward and Backward Coupling

Training $\int_0^T \mathbb{E}_{x_t}\left[\|\nabla \log p_t(x_t) - s(x_t, t)\|_2^2\right] \mathrm{d}t$

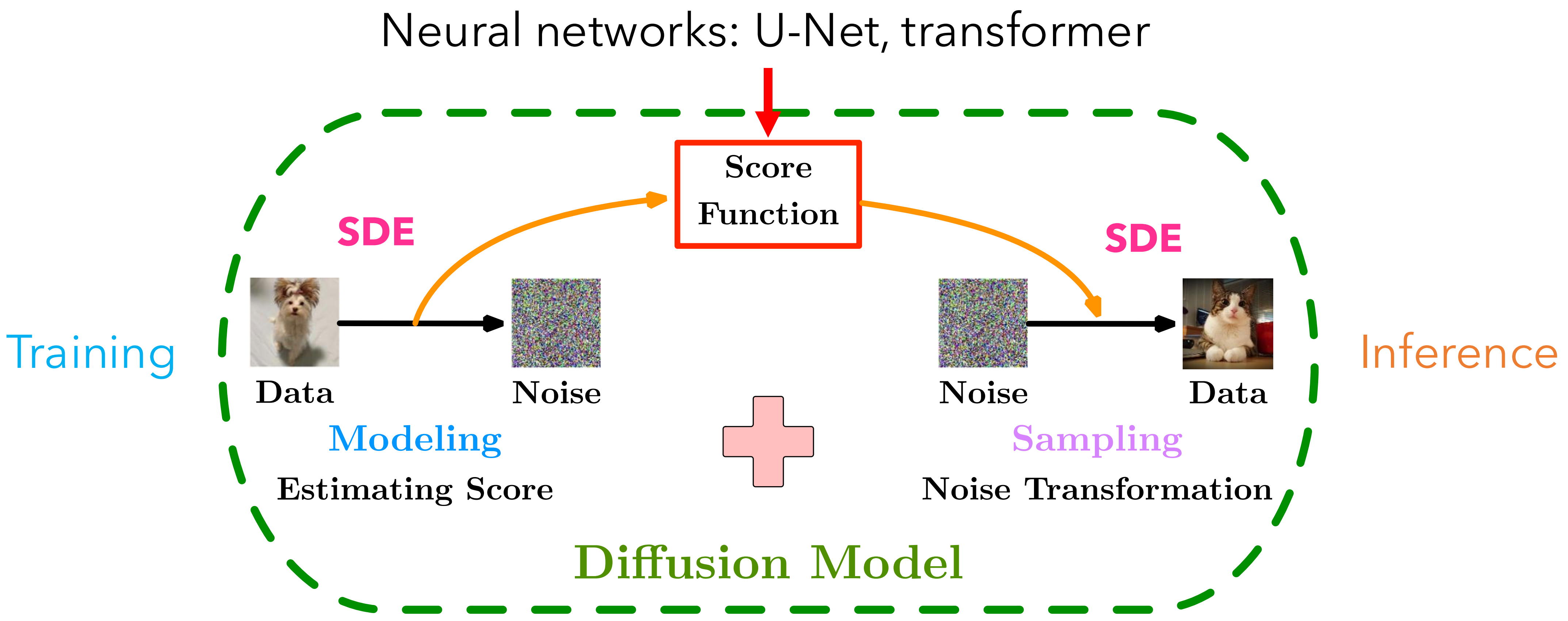
Unsupervised learning → **Regression!**



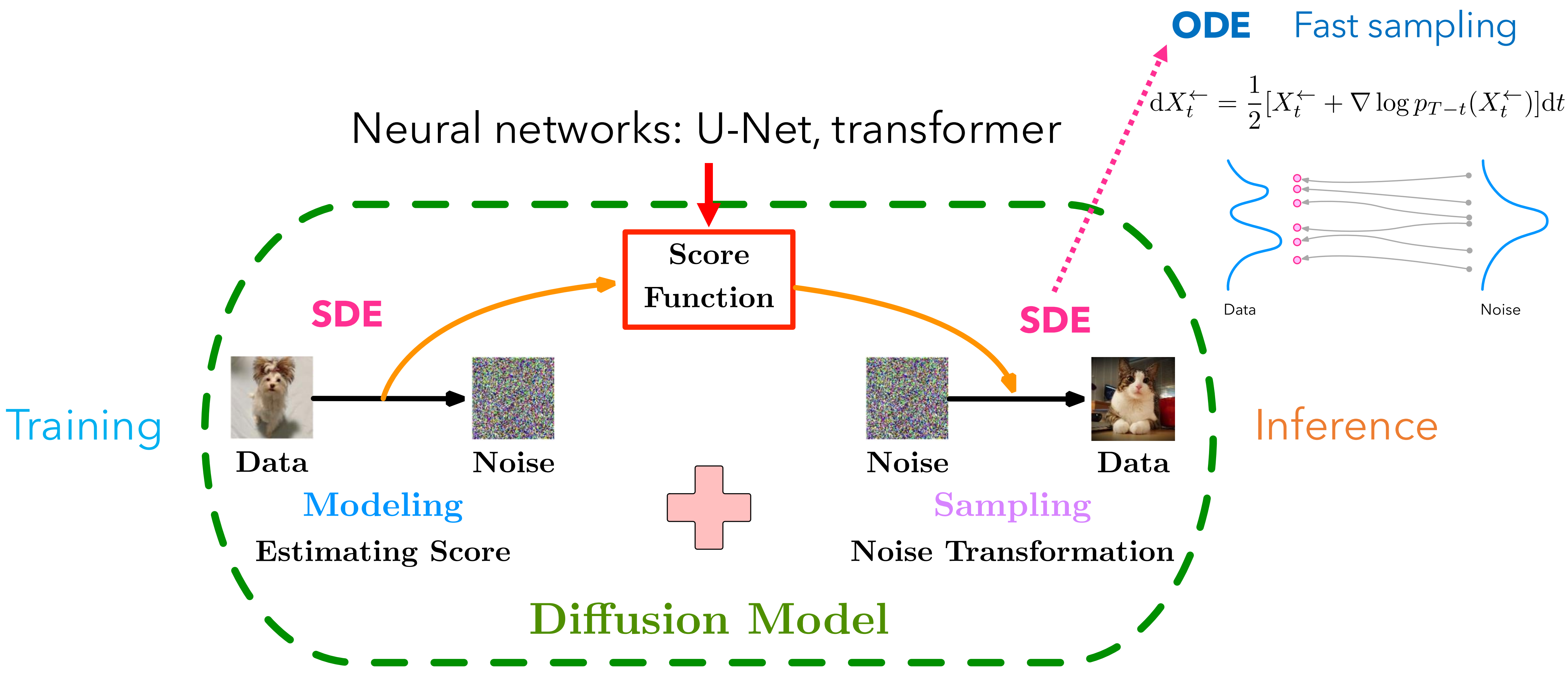
Decomposition of Diffusion Models



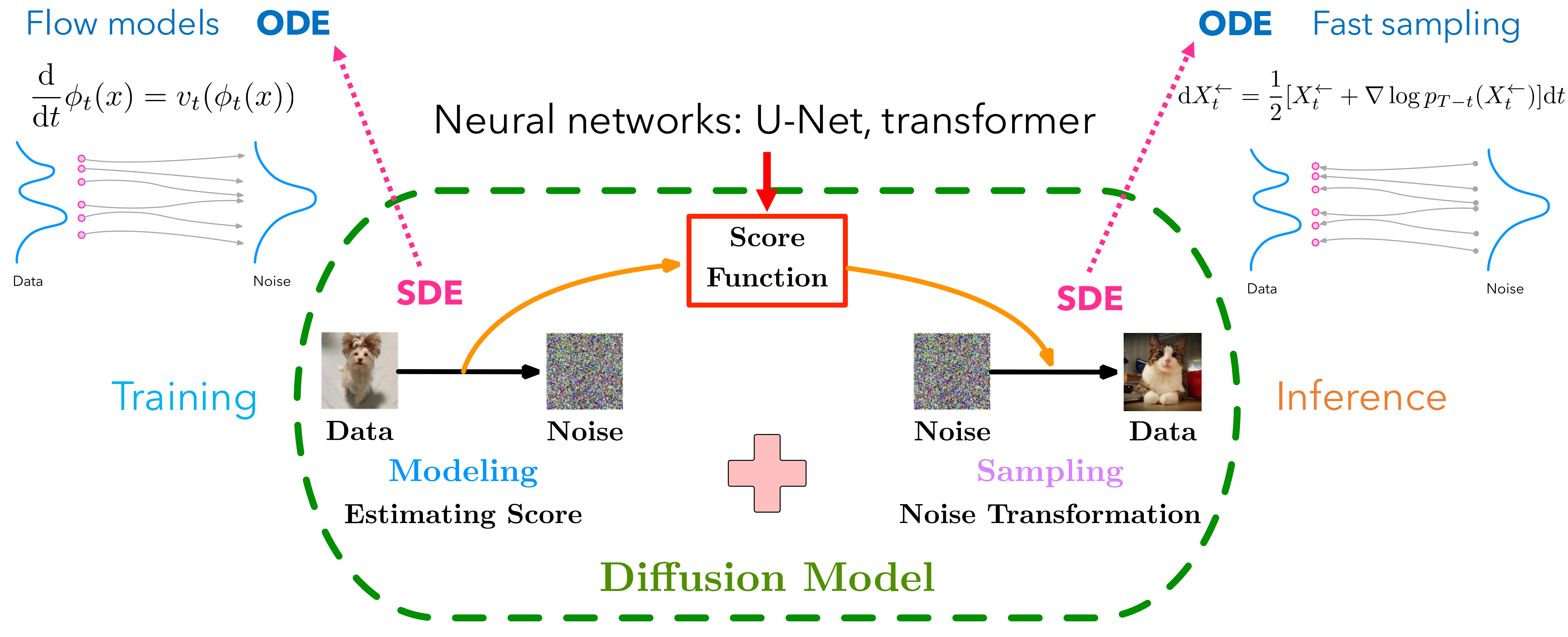
Decomposition of Diffusion Models



Decomposition of Diffusion Models

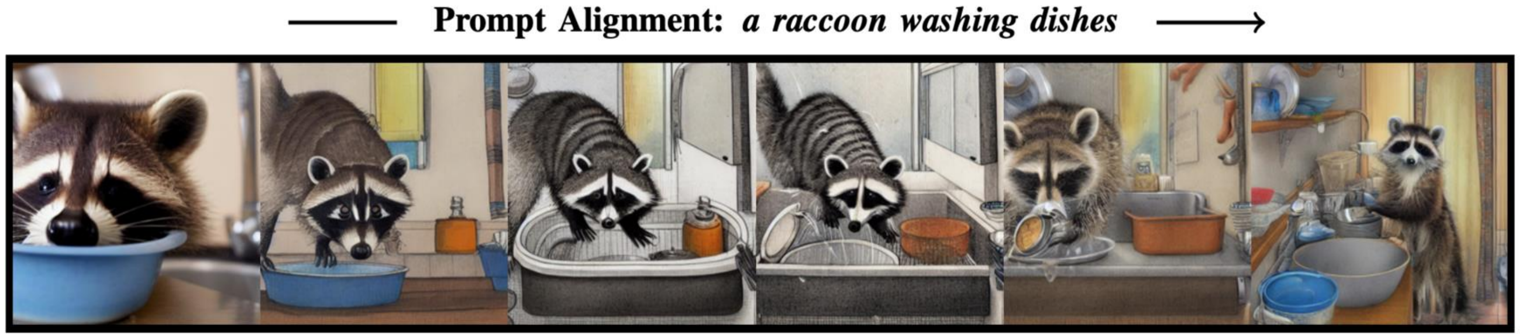


Decomposition of Diffusion Models

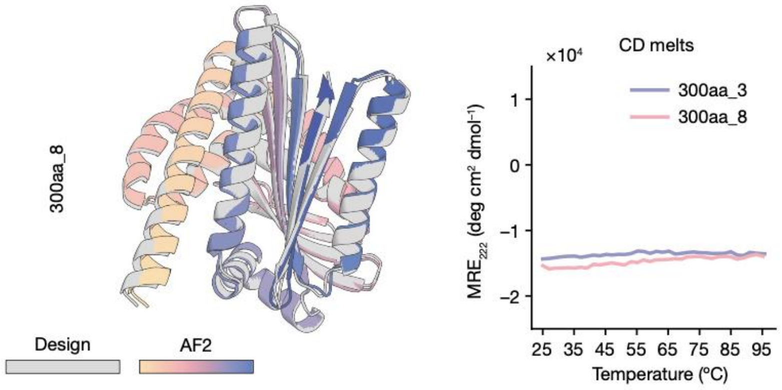


From $P(x)$ to $P(x|y)$

- Text-to-image generation (Black et al., 2023)



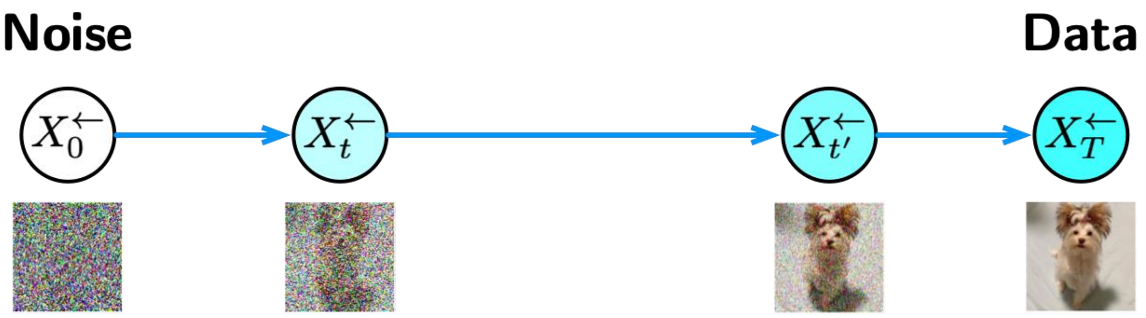
- Protein generation with biochemical properties (Watson et al., 2023; Gruver et al., 2023)



Adding Guidance to Diffusion Models

- Conditioned sample generation

$y = \mathrm{dog}$

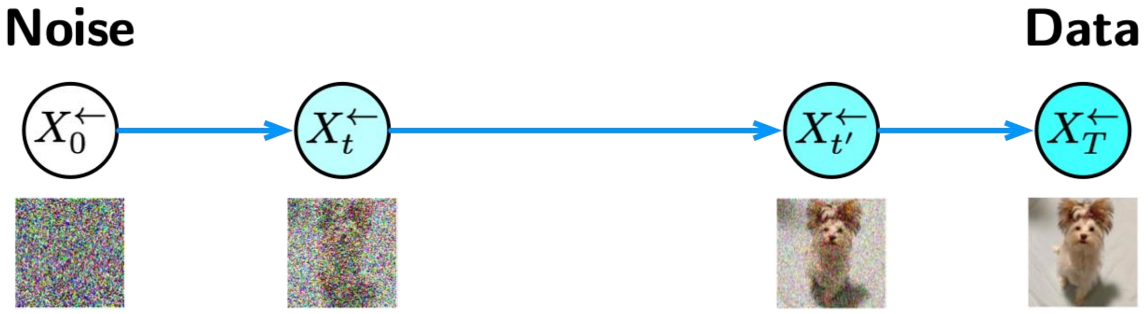


$$\mathrm{d}X_t^{\leftarrow} = \left[\frac{1}{2}X_t^{\leftarrow} + \boxed{\nabla \log p_{T-t}(X_t^{\leftarrow} \mid y)}\right]\mathrm{d}t + \mathrm{d}\overline{W}_t$$

Conditional Score **Brownian**

Adding Guidance to Diffusion Models

- Conditioned sample generation



$y = \text{dog}$

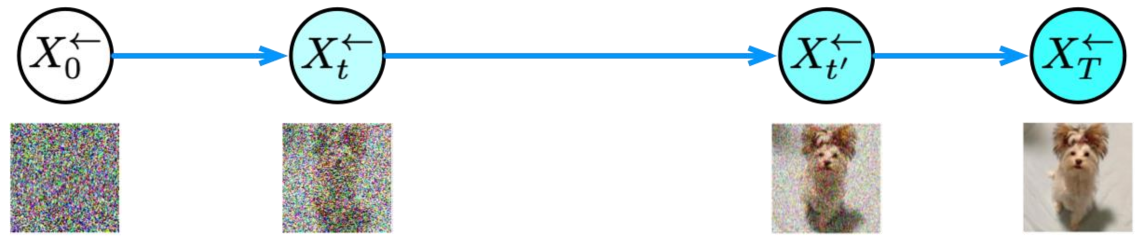
Guidance

$$\mathrm{d}X_t^{\leftarrow} = \left[\frac{1}{2}X_t^{\leftarrow} + \boxed{\nabla \log p_{T-t}(X_t^{\leftarrow} \mid y)}\right]\mathrm{d}t + \mathrm{d}\overline{W}_t$$

Conditional Score **Brownian**

Adding Guidance to Diffusion Models

- Conditioned sample generation



$y = \mathrm{dog}$

Guidance

$$\mathrm{d}X_t^{\leftarrow} = \left[\frac{1}{2}X_t^{\leftarrow} + \boxed{\nabla \log p_{T-t}(X_t^{\leftarrow} \mid y)}\right]\mathrm{d}t + \mathrm{d}\overline{W}_t$$

Conditional Score **Brownian**

Classifier guidance (Dhariwal & Nichol, 2021)
Classifier-free guidance (Ho & Salimans, 2022)

The Bayes Rule – Classifier Guidance

- Discrete label

External Classifier

$$\nabla \log p_t(x_t \mid y) = \nabla \log p_t(x_t) + \nabla \log c_t(y \mid x_t)$$

Unconditioned Logit

The Bayes Rule – Classifier Guidance

- Discrete label

External Classifier

$$\nabla \log p_t(x_t \mid y) = \nabla \log p_t(x_t) + \nabla \log c_t(y \mid x_t)$$

Unconditioned Logit

- The magic in practical implementation

$$s_{\text{practice}}(x_t, y, t) = \nabla \log p_t(x_t) + \eta \nabla \log c_t(y \mid x_t)$$

The Bayes Rule – Classifier Guidance

- Discrete label

External Classifier

$$\nabla \log p_t(x_t \mid y) = \underbrace{\nabla \log p_t(x_t)}_{\text{Unconditioned}} + \underbrace{\nabla \log c_t(y \mid x_t)}_{\text{Logit}}$$

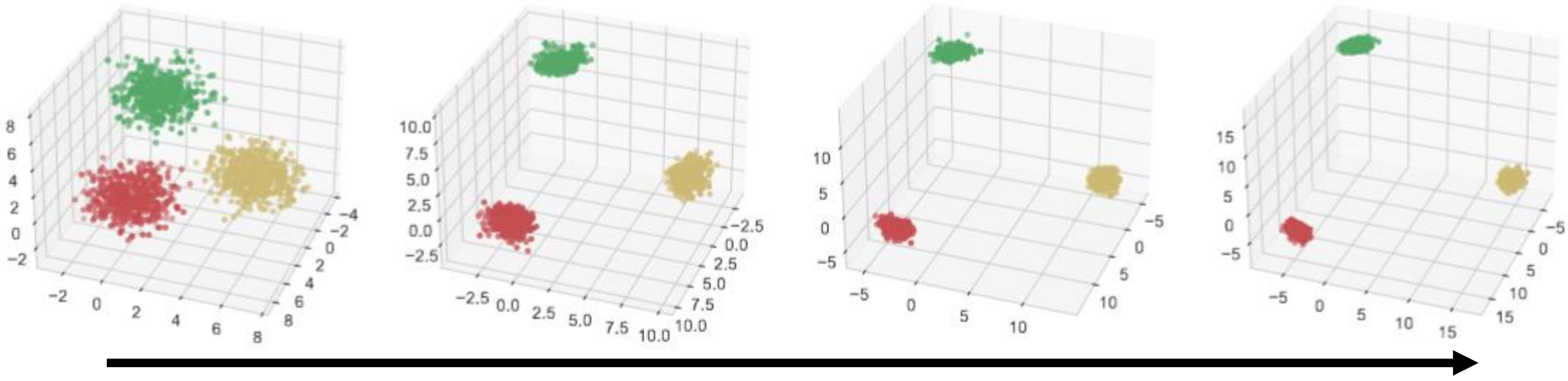
- The magic in practical implementation

$$s_{\text{practice}}(x_t, y, t) = \nabla \log p_t(x_t) + \eta \nabla \log c_t(y \mid x_t)$$

🤔

A Glimpse of Influence in 3D Gaussian Mixture

$\eta = 1$ $\eta = 10$ $\eta = 50$ $\eta = 100$



Class alignment increase

Diversity decrease

-- Y. Wu, M. Chen, Z. Li, M. Wang, Y. Wei. "Theoretical Insights for Diffusion Guidance: A Case Study for Gaussian Mixture Models", ICML 2024.

Classifier-Free Guidance

- Limitations of classifier guidance

Discrete label and external training

Classifier-Free Guidance

- Limitations of classifier guidance

Discrete label and external training

- Classifier-free guidance introduces a mask signal

$$\tau \in \{\emptyset, \mathrm{id}\}$$

Classifier-Free Guidance

- Limitations of classifier guidance

Discrete label and external training

- Classifier-free guidance introduces a mask signal

$$\tau \in \{\emptyset, \mathrm{id}\} \xrightarrow{\tau = \emptyset} s_1(x, \emptyset, t)$$

Classifier-Free Guidance

- Limitations of classifier guidance

Discrete label and external training

- Classifier-free guidance introduces a mask signal

$$
\begin{tikzcd}
 & s_1(x, \emptyset, t) \\
\tau \in \{\emptyset, \mathrm{id}\} \arrow[ur, "\tau = \emptyset"] \arrow[dr, "\tau = \mathrm{id}"'] & \\
 & s_2(x, y, t)
\end{tikzcd}
$$

Classifier-Free Guidance

- Limitations of classifier guidance

Discrete label and external training

- Classifier-free guidance introduces a mask signal

$$\tau \in \{\emptyset, \mathrm{id}\} \quad \xrightarrow{\tau = \emptyset} \; s_1(x, \emptyset, t)$$

Ignore label to extract common features

$$\tau \in \{\emptyset, \mathrm{id}\} \quad \xrightarrow{\tau = \mathrm{id}} \; s_2(x, y, t)$$

Classifier-Free Guidance

- Limitations of classifier guidance

Discrete label and external training

- Classifier-free guidance introduces a mask signal

$$
\begin{tikzcd}
 & s_1(x, \emptyset, t) \\
\tau \in \{\emptyset, \mathrm{id}\} \arrow[ur, "\tau = \emptyset"] \arrow[dr, "\tau = \mathrm{id}"'] & \\
 & s_2(x, y, t)
\end{tikzcd}
$$

$s_1(x, \emptyset, t)$ Ignore label to extract common features

$s_2(x, y, t)$ Respect label to find specific information

Classifier-Free Guidance

- Limitations of classifier guidance

Discrete label and external training

- Classifier-free guidance introduces a mask signal

$$\tau \in \{\emptyset, \mathrm{id}\} \quad \begin{cases} \xrightarrow{\tau = \emptyset} s_1(x, \emptyset, t) & \text{Ignore label to extract common features} \\ \xrightarrow{\tau = \mathrm{id}} s_2(x, y, t) & \text{Respect label to find specific information} \end{cases}$$

Trained together
Share parameters

Modeling Diverse Data

Practical Data Is High-D And Complex

- ImageNet resolution: $D = 224 \times 224 \times 3$

Practical Data Is High-D And Complex

- ImageNet resolution: $D = 224 \times 224 \times 3$

Curse of Dimensionality

Practical Data Is High-D And Complex

- ImageNet resolution: $D = 224 \times 224 \times 3$

Curse of Dimensionality

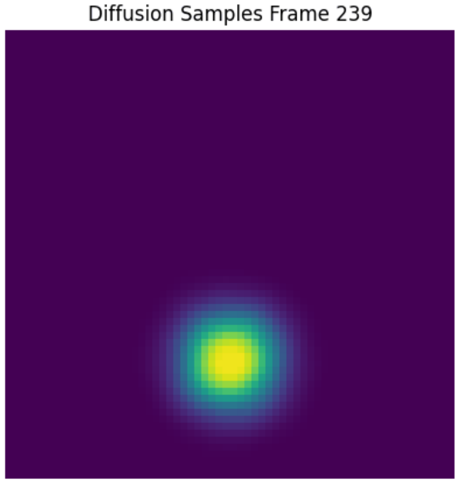
- Sequential data enlarges the dimension heavily

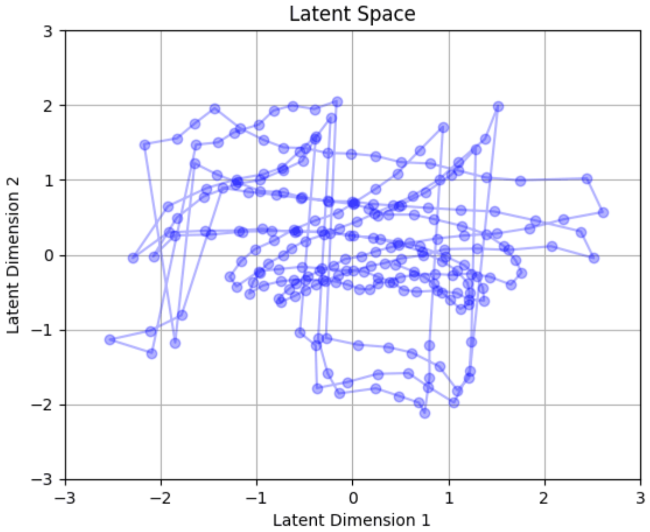
Practical Data Is High-D And Complex

- ImageNet resolution: $D = 224 \times 224 \times 3$

Curse of Dimensionality 😰

- Sequential data enlarges the dimension heavily





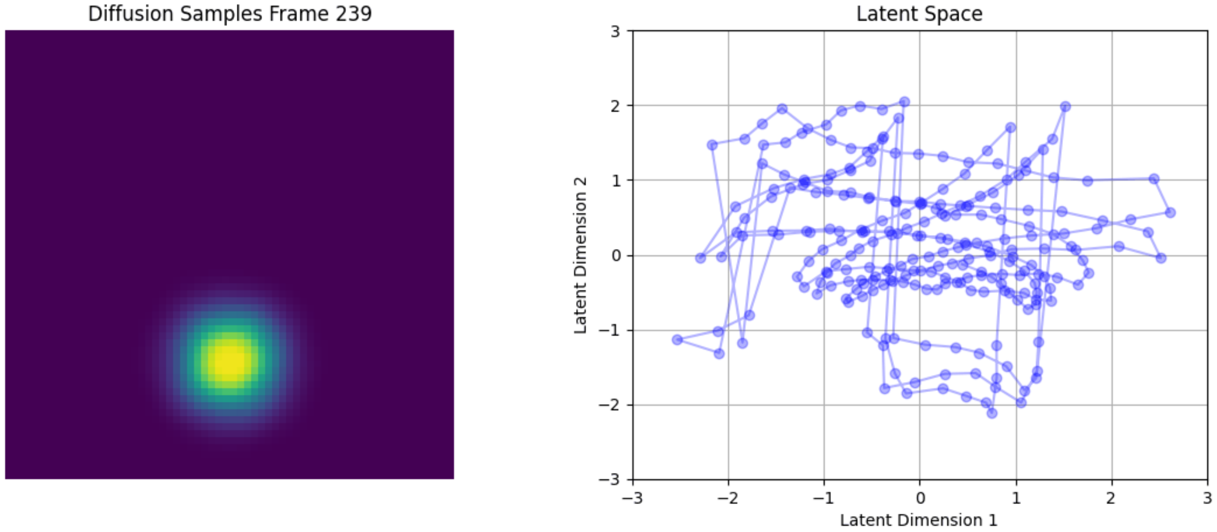
2D bouncing ball

Practical Data Is High-D And Complex

- ImageNet resolution: $D = 224 \times 224 \times 3$

Curse of Dimensionality

- Sequential data enlarges the dimension heavily
- also introduces **spatial-temporal dependencies**



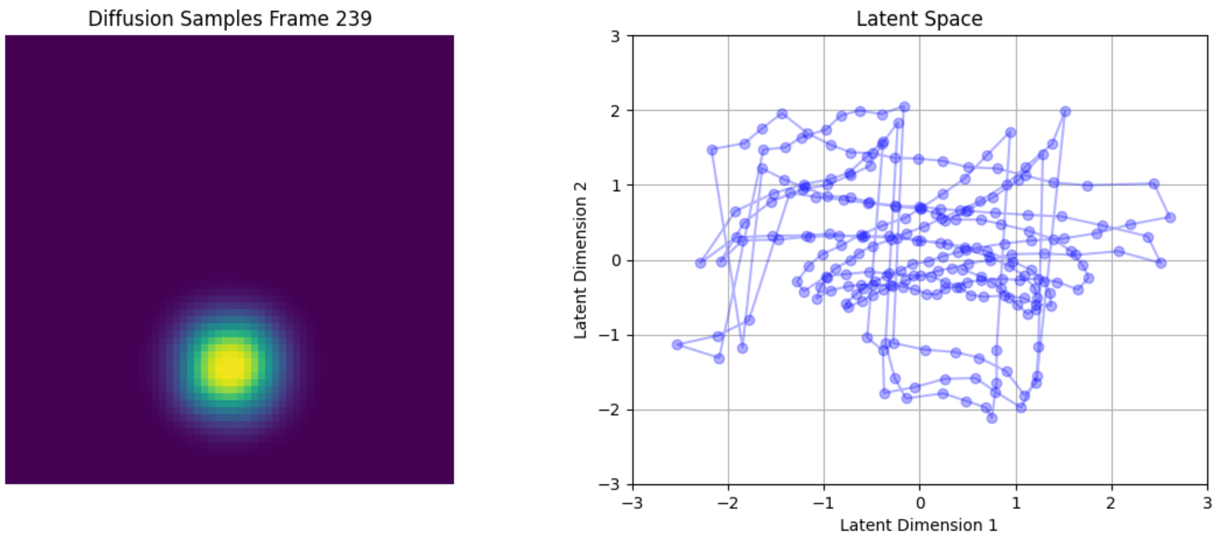
2D bouncing ball

Practical Data Is High-D And Complex

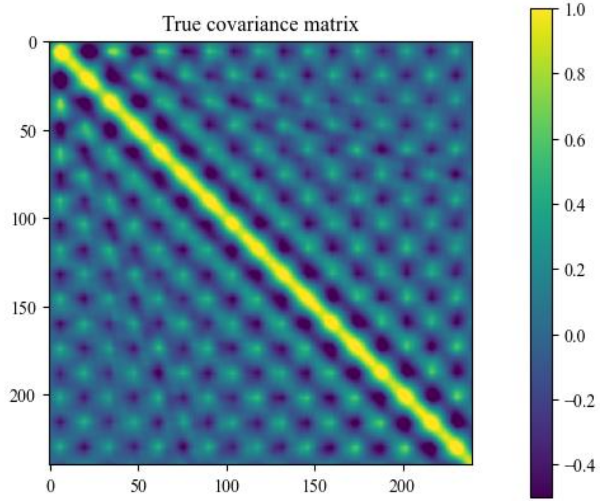
- ImageNet resolution: $D = 224 \times 224 \times 3$

Curse of Dimensionality

- Sequential data enlarges the dimension heavily
- also introduces **spatial-temporal dependencies**

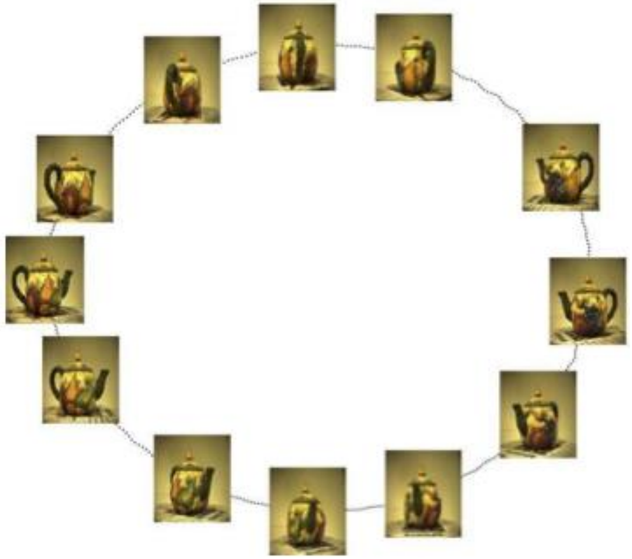


2D bouncing ball

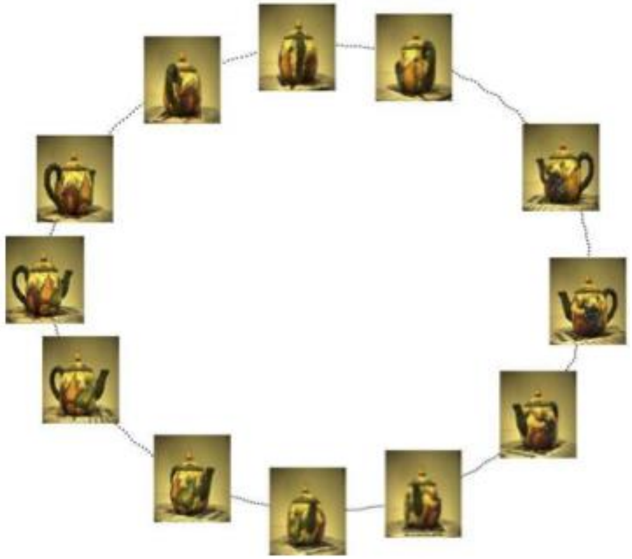


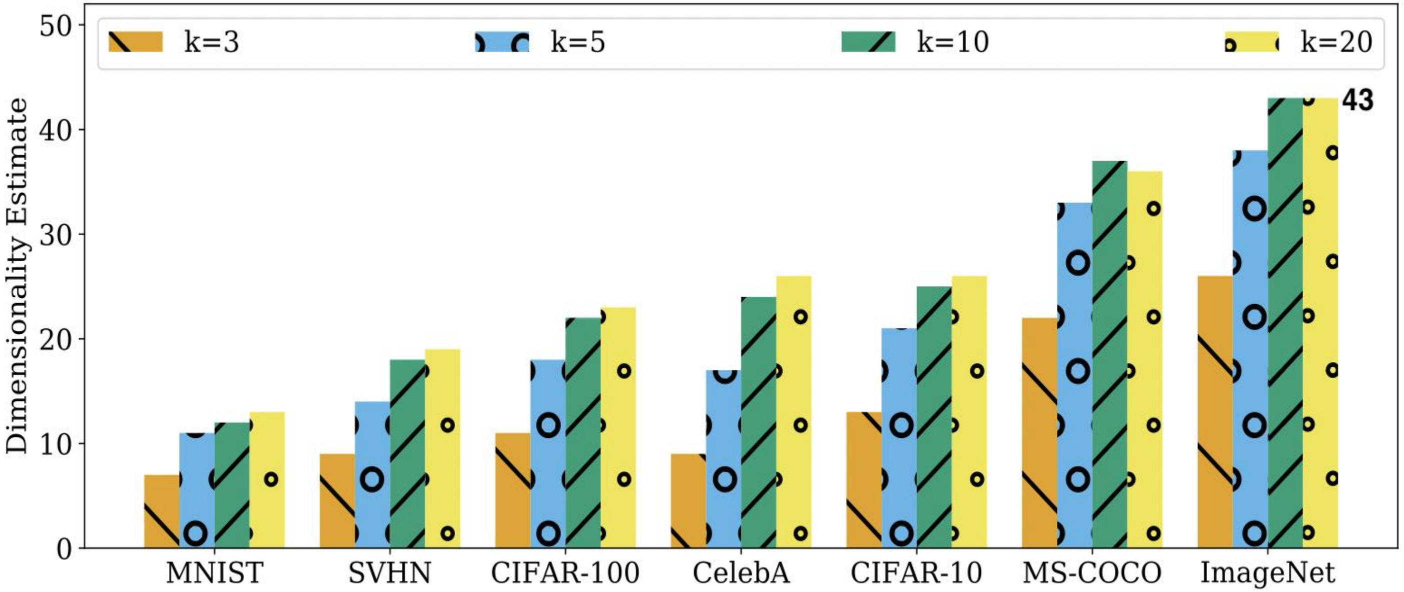
Correlation between frames

Practical High-D Data Is Low-Dimensional



Practical High-D Data Is Low-Dimensional





$$224 \times 224 \times 3 \qquad \text{v.s.} \qquad \leq 43$$

-- Figure credit: (Weinberger & Saul, 2006; P. Pope et al., 2021)

How Diffusion Model Finds Structures?

- Dynamic evolution

Backward $$\mathrm{d}X_t^{\leftarrow} = \left[\frac{1}{2}X_t^{\leftarrow} + \boxed{\nabla \log p_{T-t}(X_t^{\leftarrow})}\right]\mathrm{d}t + \mathrm{d}\overline{W}_t$$

Score Function **Brownian**

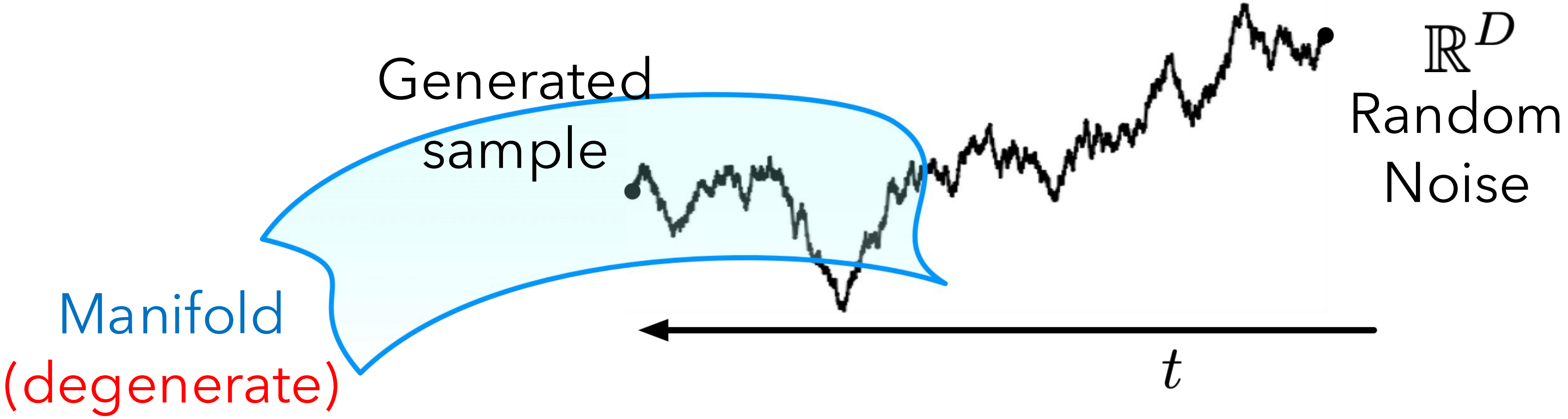
How Diffusion Model Finds Structures?

- Dynamic evolution

Backward

$$\mathrm{d}X_t^{\leftarrow} = \left[\frac{1}{2}X_t^{\leftarrow} + \underbrace{\boxed{\nabla \log p_{T-t}(X_t^{\leftarrow})}}_{\text{Score Function}}\right]\mathrm{d}t + \underbrace{\mathrm{d}\overline{W}_t}_{\text{Brownian}}$$

- Start from high-D but land in a manifold?



Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \quad \text{with} \quad z \sim P_z \quad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1 - e^{-t}} \left(I_D - AA^\top\right) x}$$

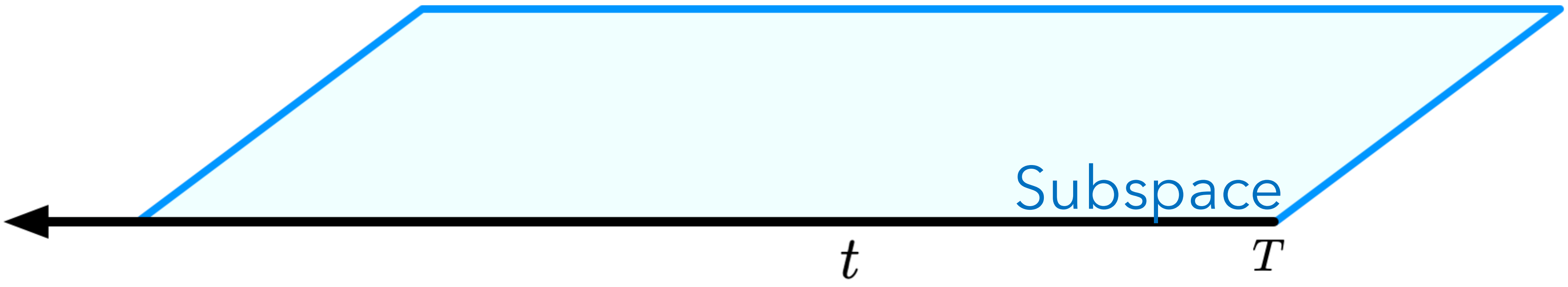
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \quad \text{with} \quad z \sim P_z \quad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}}\left(I_D - AA^\top\right)x}$$



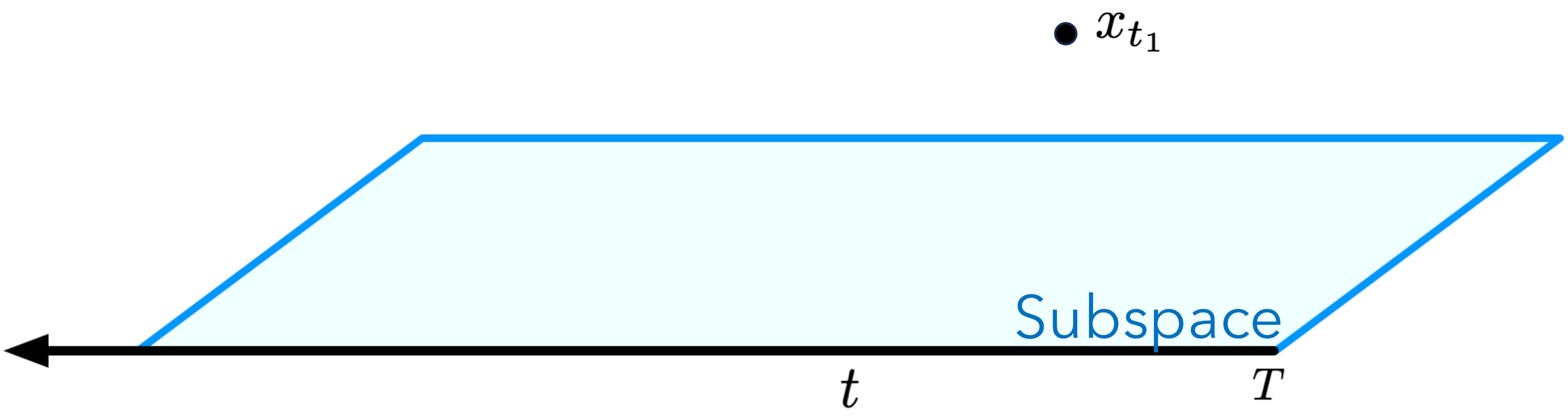
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \quad \text{with} \quad z \sim P_z \quad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}}\left(I_D - AA^\top\right)x}$$



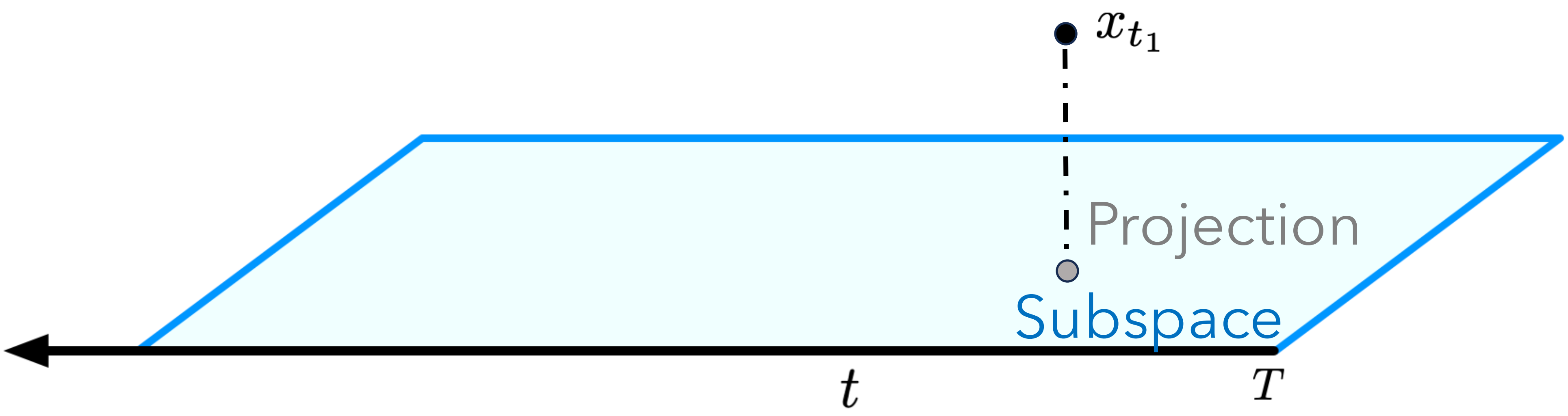
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \qquad \text{with} \qquad z \sim P_z \qquad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}} \left(I_D - AA^\top\right) x}$$



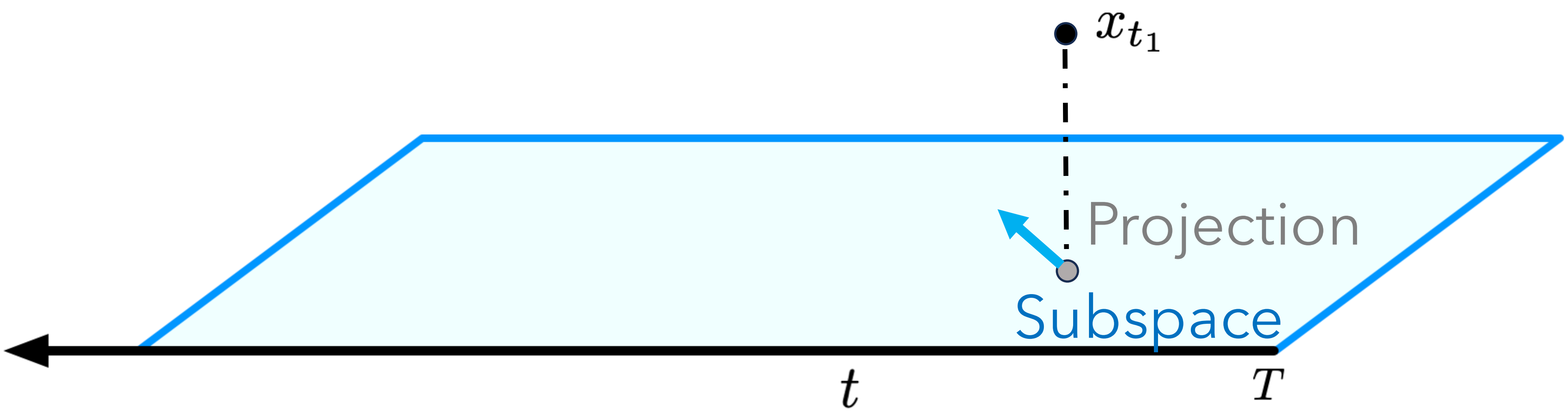
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \qquad \text{with} \qquad z \sim P_z \qquad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}} \left(I_D - AA^\top\right) x}$$



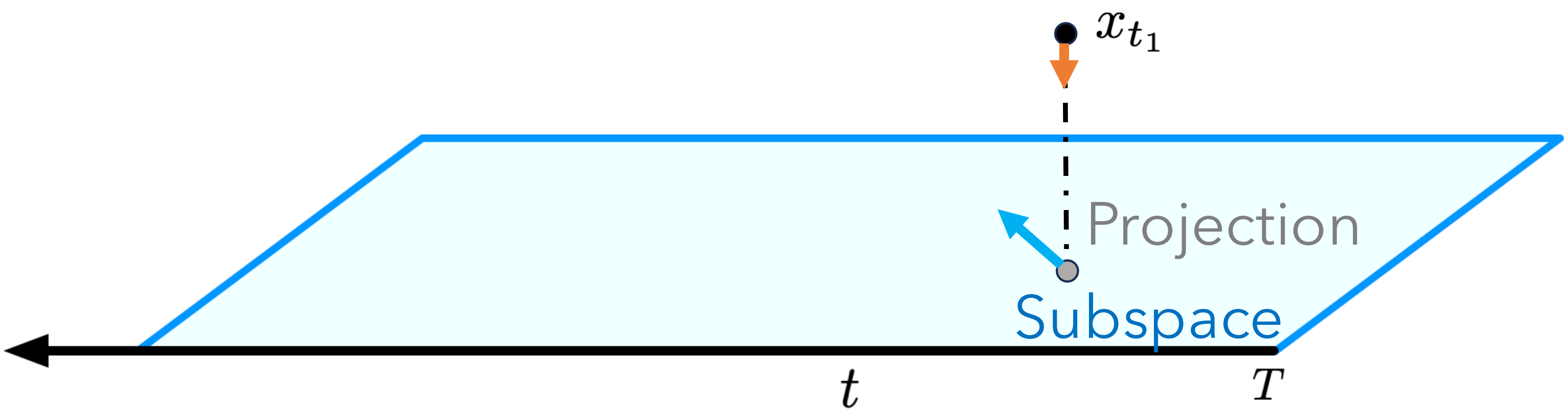
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \quad \text{with} \quad z \sim P_z \quad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1 - e^{-t}} \left(I_D - AA^\top \right) x}$$



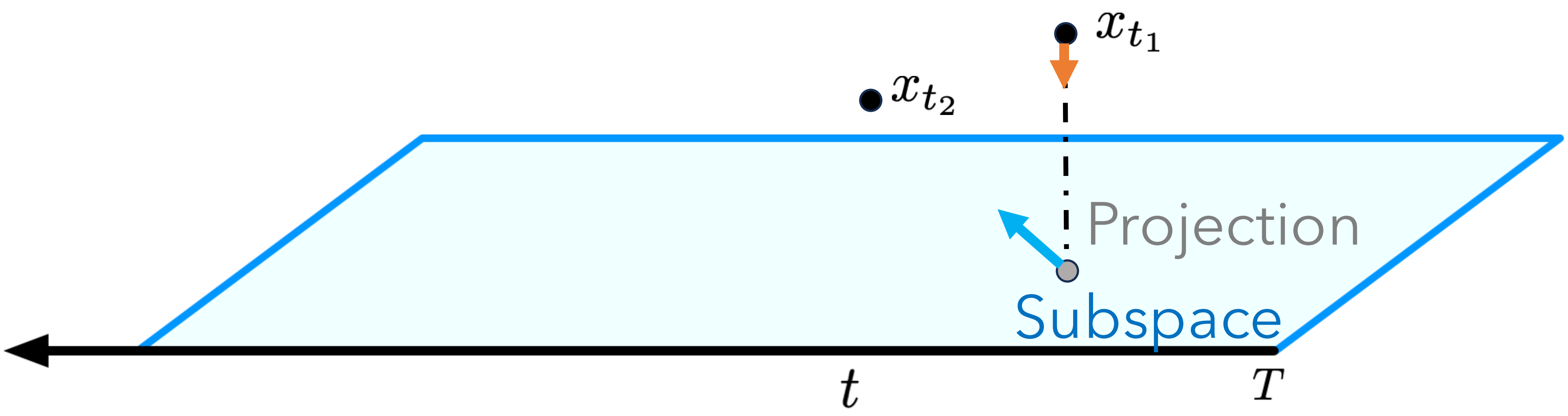
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \qquad \text{with} \qquad z \sim P_z \qquad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}} \left(I_D - AA^\top\right) x}$$



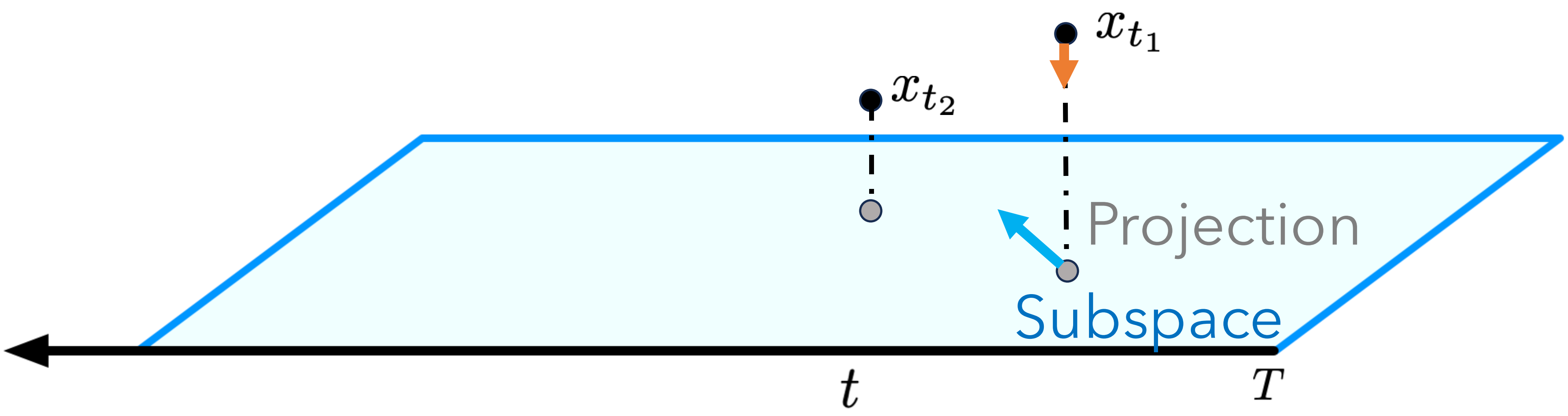
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \qquad \text{with} \qquad z \sim P_z \qquad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}} \left(I_D - AA^\top\right) x}$$



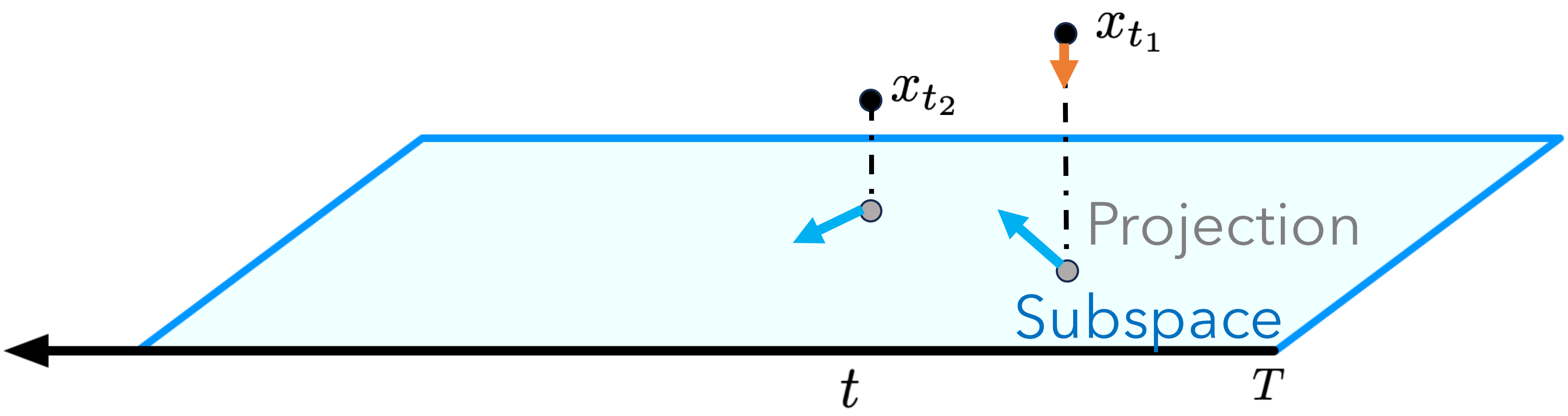
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \quad \text{with} \quad z \sim P_z \quad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}} \left(I_D - AA^\top\right) x}$$



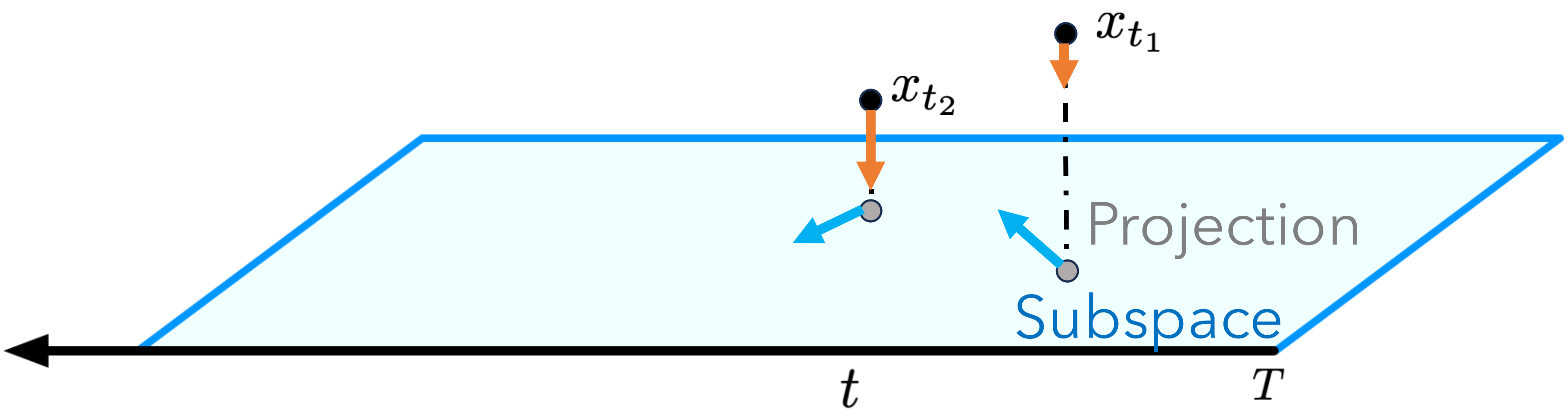
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \quad \text{with} \quad z \sim P_z \quad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}} \left(I_D - AA^\top\right) x}$$



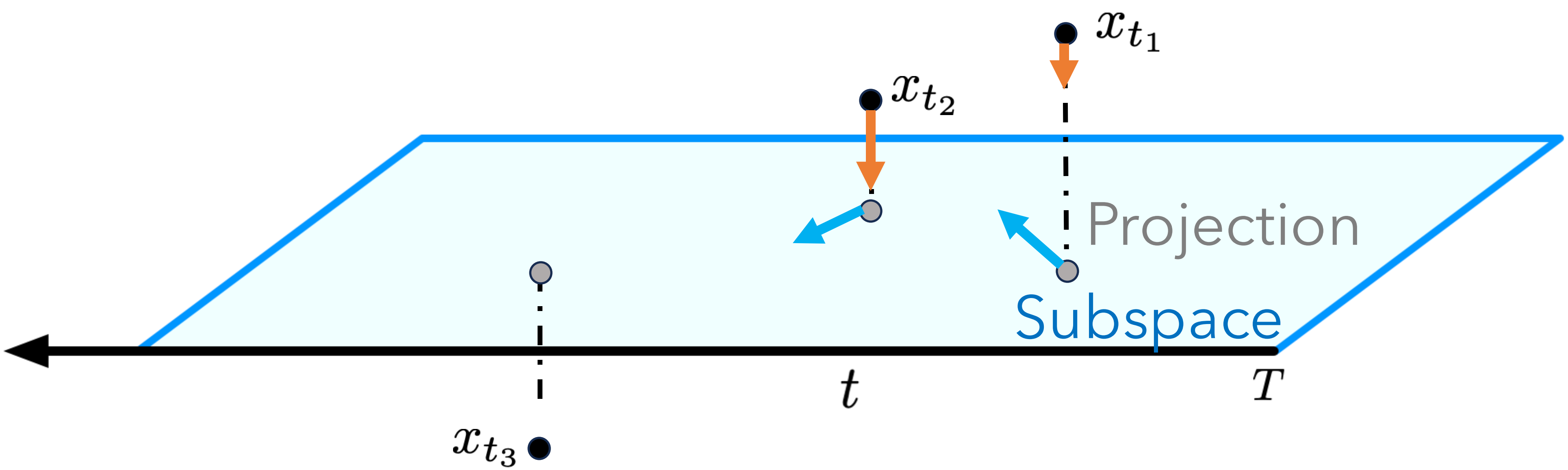
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \qquad \text{with} \qquad z \sim P_z \qquad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}}\left(I_D - AA^\top\right)x}$$



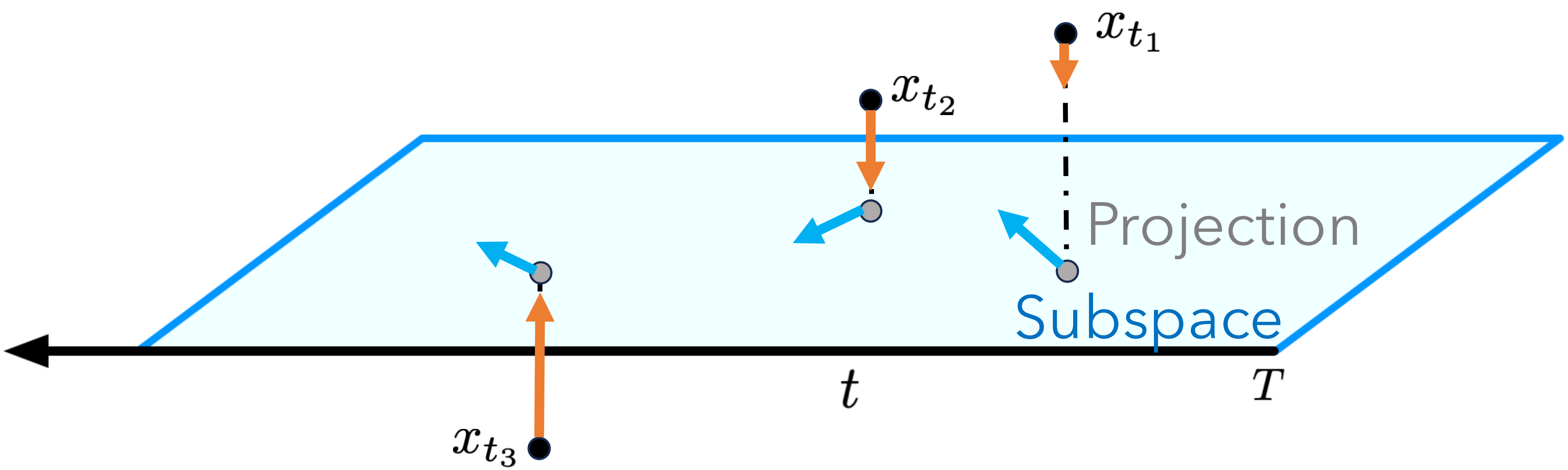
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \qquad \text{with} \qquad z \sim P_z \qquad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}} \left(I_D - AA^\top\right) x}$$



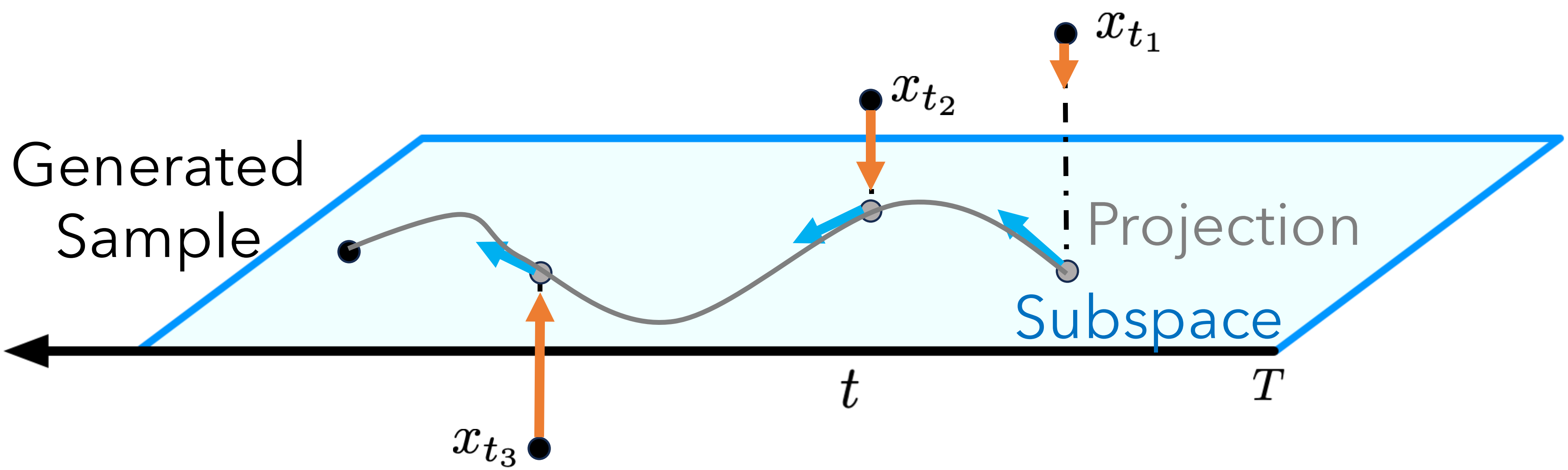
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \qquad \text{with} \qquad z \sim P_z \qquad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}}\left(I_D - AA^\top\right)x}$$



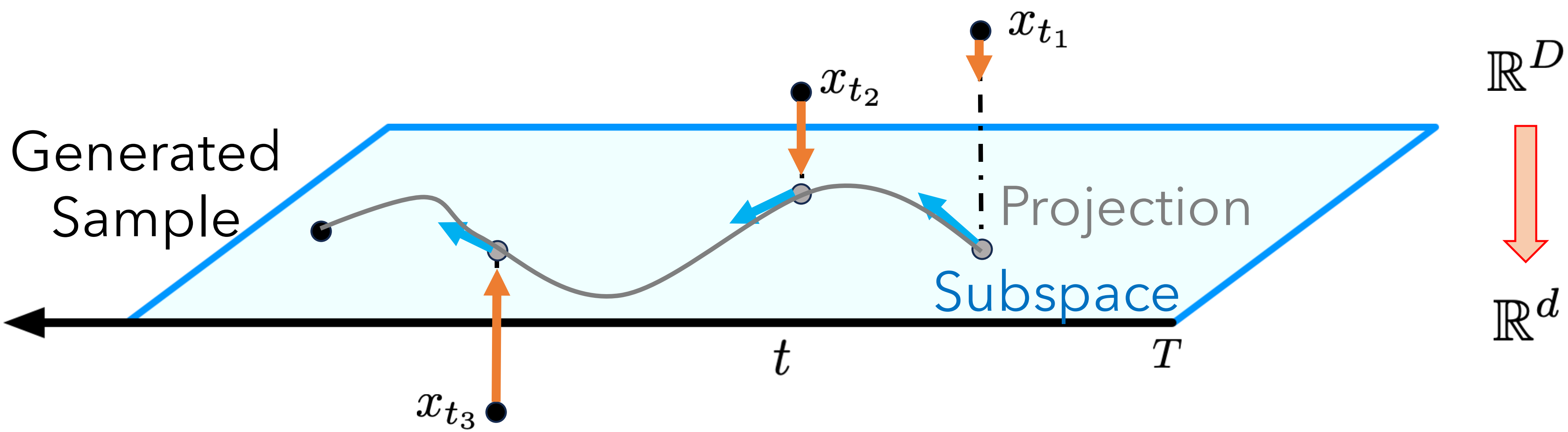
Score Function Adapts to Data Structures

- Linear subspace data

$$x = Az \quad \text{with} \quad z \sim P_z \quad z \in \mathbb{R}^d$$

- Score decomposition

$$\nabla \log p_t(x) = \boxed{A \nabla \log p_t^z(A^\top x)} - \boxed{\frac{1}{1-e^{-t}} \left(I_D - AA^\top\right) x}$$



Diffusion Model Efficiently Learns Low-D Data

Theorem

✓ Score function can be learned efficiently at the rate

$$\tilde{\mathcal{O}}\left(n^{-\frac{1}{2(d+5)}}\right)$$

✓ Underlying distribution is learned at the same rate.

-- M. Chen, K. Huang, T. Zhao, M. Wang. "Score Approximation, Estimation and Distribution Recovery of Diffusion Models on Low-Dimensional Data", ICML 2023

Diffusion Model Efficiently Learns Low-D Data

Theorem

✓ Score function can be learned efficiently at the rate

$$\tilde{O}\left(n^{-\frac{1}{2(d+5)}}\right)$$

✓ Underlying distribution is learned at the same rate.

Take-home message:

✓ **Accurate** in learning data distributions

✓ **Efficient**: no curse of dimensionality

✓ **Generalizable** to manifold data (Tang and Yang, 2024)

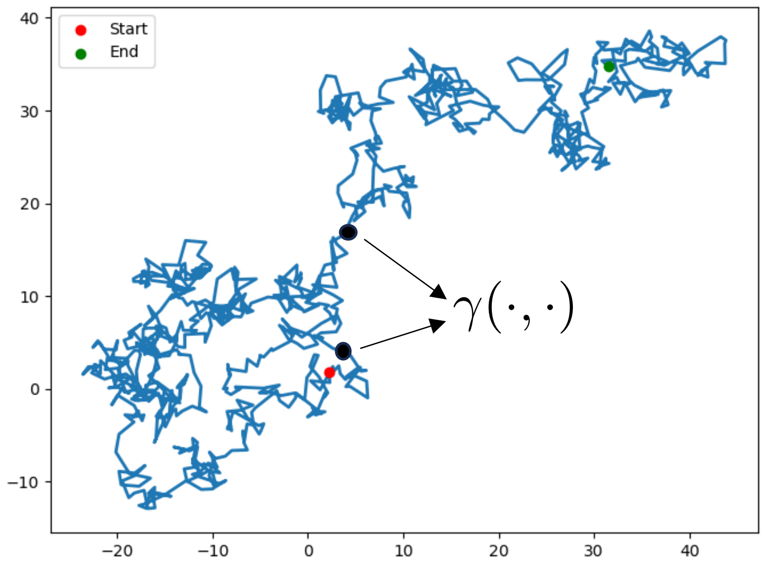
-- M. Chen, K. Huang, T. Zhao, M. Wang. "Score Approximation, Estimation and Distribution Recovery of Diffusion Models on Low-Dimensional Data", ICML 2023

Sequence Data with Dependencies

- We consider

$$X_1, \ldots, X_{h_N} \sim \mathcal{GP}(\mu(\cdot), \gamma(\cdot,\cdot), \Lambda) \in \mathbb{R}^D$$

  - $0 = h_1 < \cdots < h_N = H$ sampling times
  - $\mu(\cdot)$ time varying mean function
  - $\gamma(\cdot,\cdot)$ covariance function (kernel)
  - $\Lambda = \mathrm{Cov}[X_h]$ marginal covariance matrix



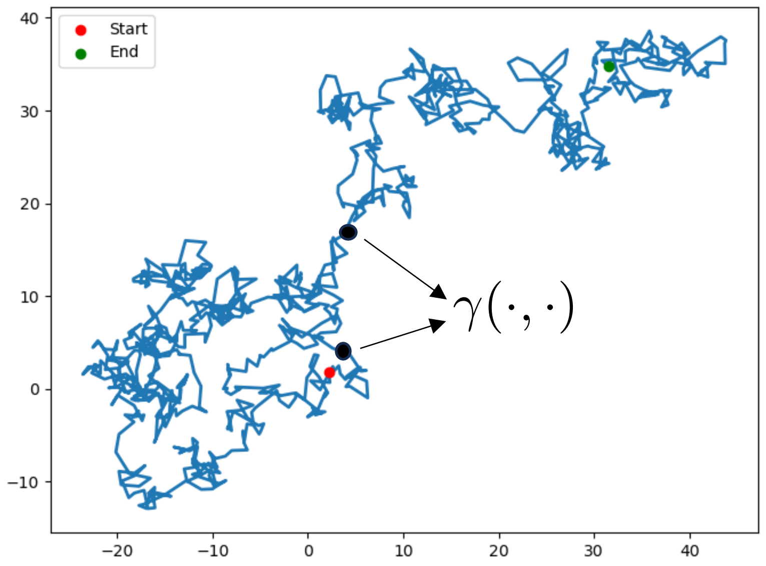
2D Brownian motion

Sequence Data with Dependencies

- We consider

$$X_1, \dots, X_{h_N} \sim \mathcal{GP}(\mu(\cdot), \gamma(\cdot,\cdot), \Lambda) \in \mathbb{R}^D$$

  - $0 = h_1 < \cdots < h_N = H$ sampling times
  - $\mu(\cdot)$ time varying mean function
  - $\gamma(\cdot,\cdot)$ covariance function (kernel)
  - $\Lambda = \mathrm{Cov}[X_h]$ marginal covariance matrix



2D Brownian motion

Simplification

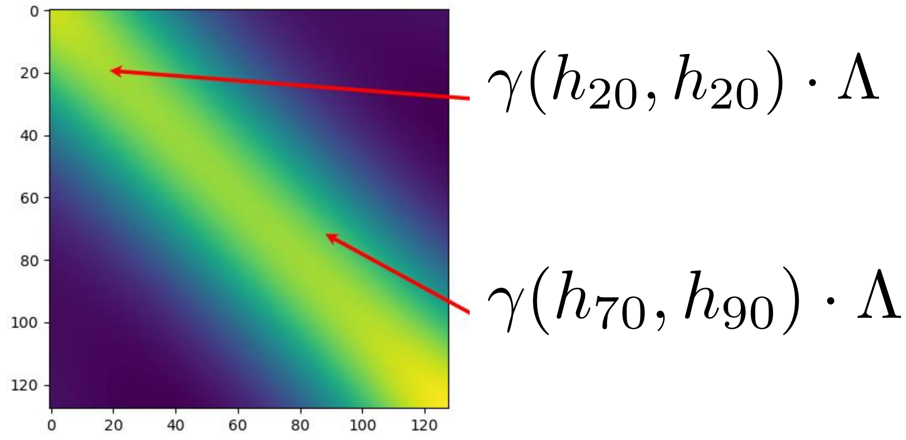
✓ $\gamma(\cdot,\cdot)$ only depends on time gaps, i.e.,

$$\gamma(t_1, t_2) = g(|t_1 - t_2|)$$

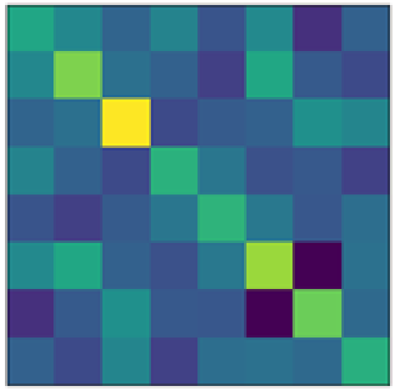
Description of Spatial-Temporal Dependencies

- We stack data together as a vector in $\mathbb{R}^{DN}$, whose distribution is Gaussian

$$\mathrm{N}\left(\begin{bmatrix}\mu(h_1)\\ \vdots \\ \mu(h_N)\end{bmatrix}, \Gamma\otimes\Lambda = \begin{bmatrix}\gamma(h_1,h_1)\Lambda,\ldots,\gamma(h_1,h_N)\Lambda\\ \vdots \\ \gamma(h_N,h_1)\Lambda,\ldots,\gamma(h_N,h_N)\Lambda\end{bmatrix}\right)$$



Temporal dependencies



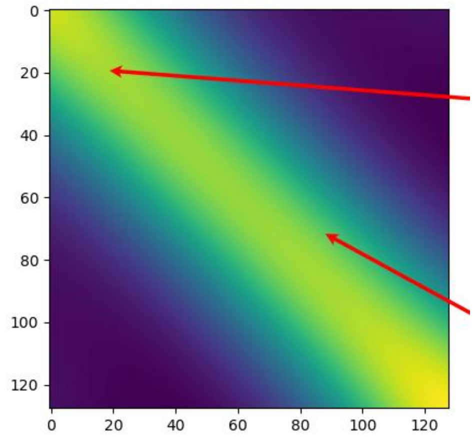
Spatial dependencies

Description of Spatial-Temporal Dependencies

- We stack data together as a vector in $\boxed{\mathbb{R}^{DN}}$, whose distribution is Gaussian

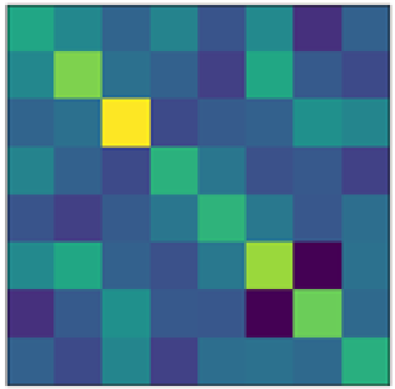
Huge dimension?

$$\mathrm{N}\left(\begin{bmatrix}\mu(h_1)\\ \vdots \\ \mu(h_N)\end{bmatrix}, \Gamma\otimes\Lambda=\begin{bmatrix}\gamma(h_1,h_1)\Lambda,\ldots,\gamma(h_1,h_N)\Lambda\\ \vdots\\ \gamma(h_N,h_1)\Lambda,\ldots,\gamma(h_N,h_N)\Lambda\end{bmatrix}\right)$$



$\gamma(h_{20},h_{20})\cdot\Lambda$

$\gamma(h_{70},h_{90})\cdot\Lambda$

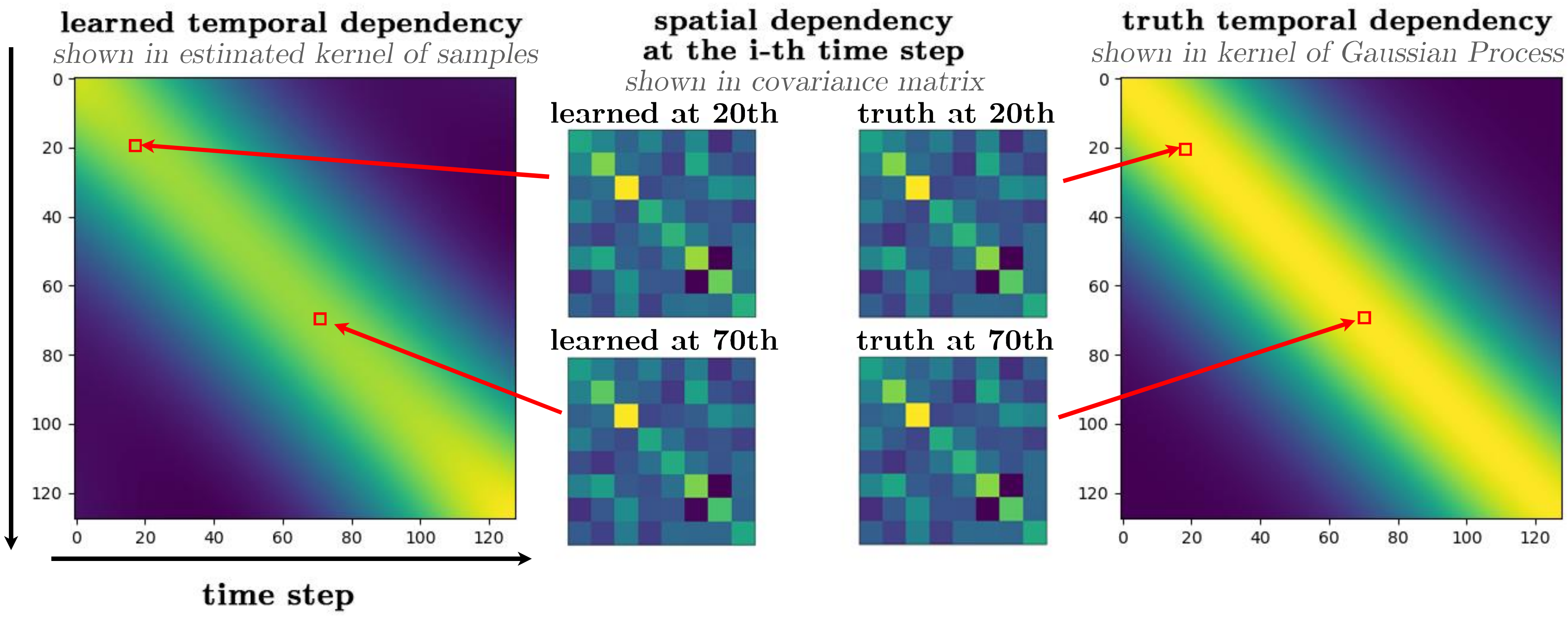


Temporal dependencies

Spatial dependencies

Proper Score Network Learns Dependencies

- We simulate a Gaussian process with 128 length
- Diffusion model with transformer learns very well!



Represent Score via Algorithm Unrolling

Score Function

$$\boxed{\nabla \log p_t(\mathbf{v}_t)} = \arg\min_{\mathbf{s} \in \mathbb{R}^{dN}} L_t(\mathbf{s}) = \frac{1}{2}\mathbf{s}^\top \left(\alpha_t^2 (\Gamma \otimes \mathbf{I}) + \sigma_t^2 \mathbf{I} \right) \mathbf{s} - (\mathbf{v}_t - \alpha_t \mu)^\top \mathbf{s}$$

Represent Score via Algorithm Unrolling

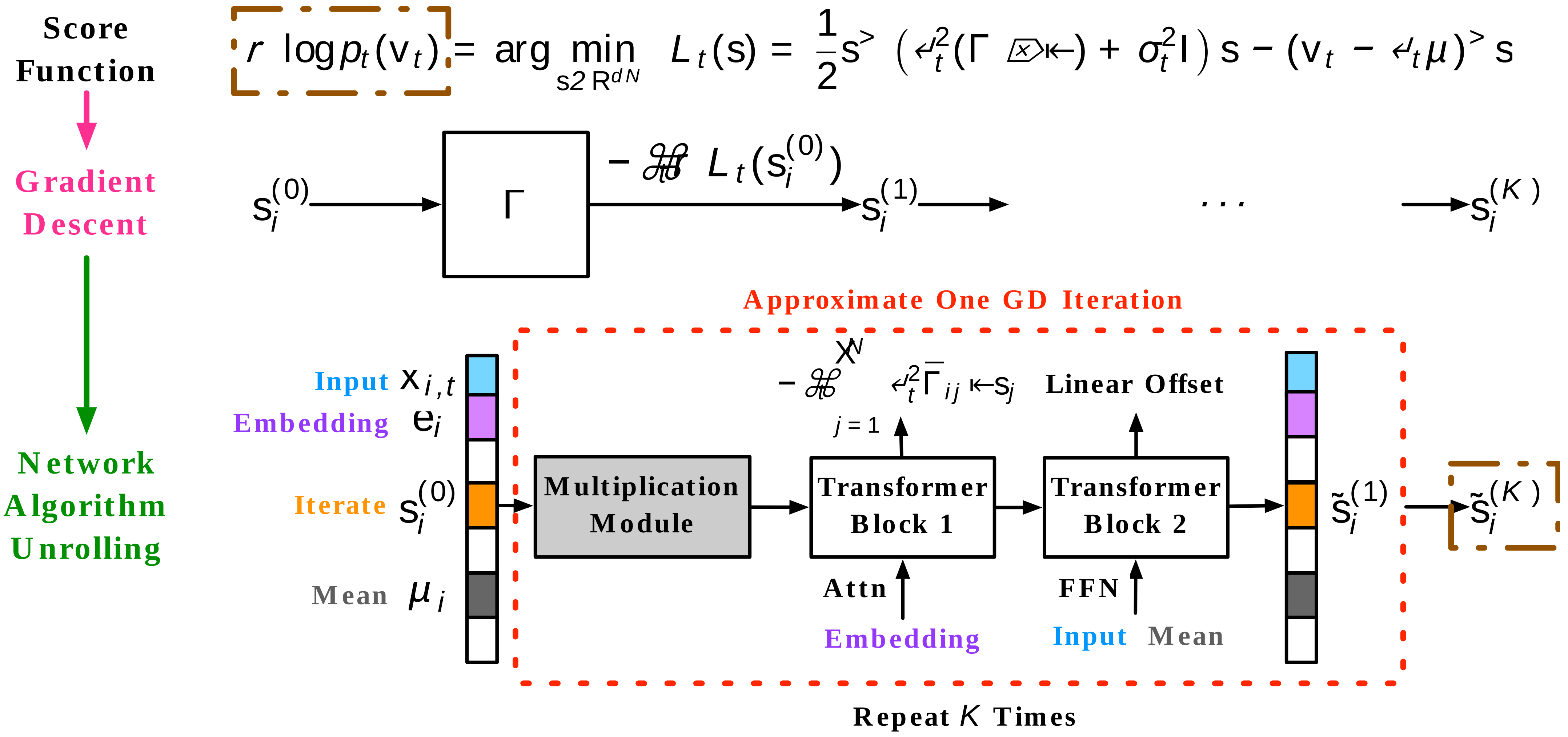
Score Function

Gradient Descent

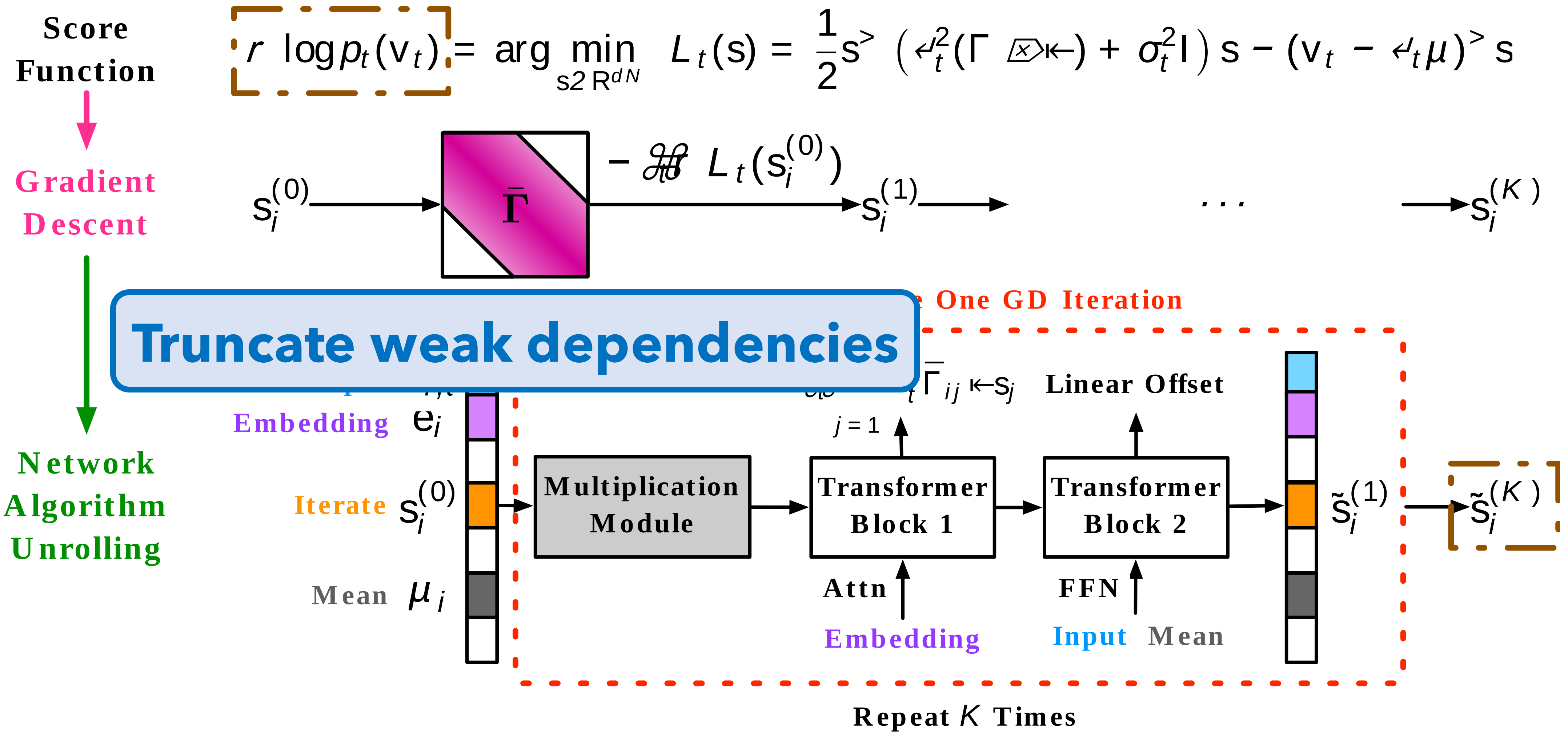
$$\boxed{\nabla \log p_t(\mathrm{v}_t)} = \arg\min_{\mathrm{s} \in \mathbb{R}^{dN}} L_t(\mathrm{s}) = \frac{1}{2}\mathrm{s}^\top \left(\alpha_t^2(\Gamma \otimes \mathrm{I}) + \sigma_t^2 \mathrm{I}\right)\mathrm{s} - (\mathrm{v}_t - \alpha_t \mu)^\top \mathrm{s}$$

$$\mathrm{s}_i^{(0)} \longrightarrow \boxed{\Gamma} \xrightarrow{-\eta_t \nabla L_t(\mathrm{s}_i^{(0)})} \mathrm{s}_i^{(1)} \longrightarrow \cdots \longrightarrow \mathrm{s}_i^{(K)}$$

Represent Score via Algorithm Unrolling



Represent Score via Algorithm Unrolling



Diffusion Model Learns Gaussian Process

Theorem

- ✓ Score function can be learned efficiently at the rate

$$\tilde{O}\left(\sqrt{\frac{\text{dependency-decay} \cdot \texttt{sequence-length} \cdot D^3}{n}}\right)$$

- ✓ Gaussian process distribution is learned at the same rate.

-- H. Fu, Z. Dou, J. Guo, M. Wang, M. Chen. “Diffusion Transformer Captures Spatial-Temporal Dependencies: A Theory for Gaussian Process Data”, 2024

Diffusion Model Learns Gaussian Process

Theorem

✓ Score function can be learned efficiently at the rate

$$\tilde{O}\left(\sqrt{\frac{\text{dependency-decay}\cdot\texttt{sequence-length}\cdot D^3}{n}}\right)$$

✓ Gaussian process distribution is learned at the same rate.

Take-home message:

✓ **Weak** dependence on the length of sequence

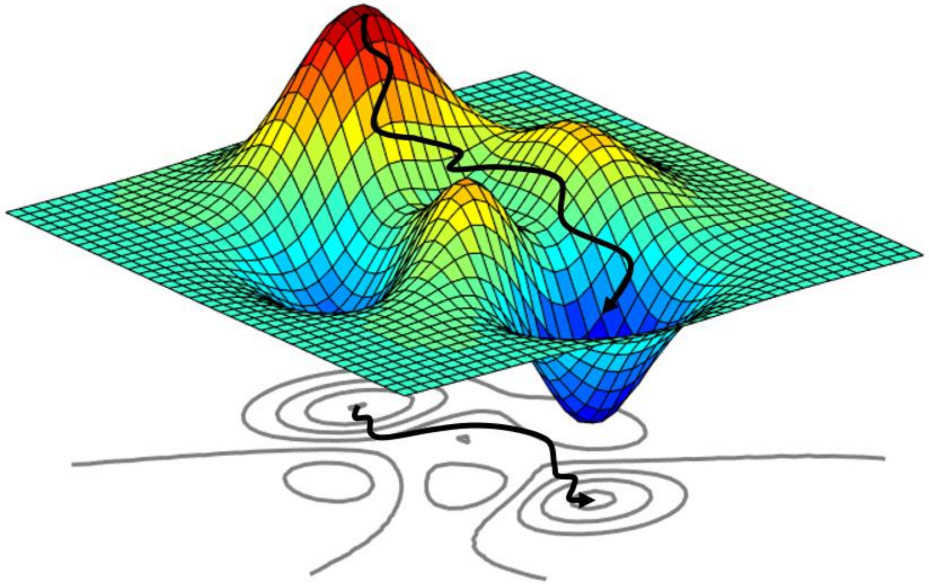
✓ **Adaptive** to spatial-temporal dependencies

-- H. Fu, Z. Dou, J. Guo, M. Wang, M. Chen. "Diffusion Transformer Captures Spatial-Temporal Dependencies: A Theory for Gaussian Process Data", 2024

Leverage Diffusion Models

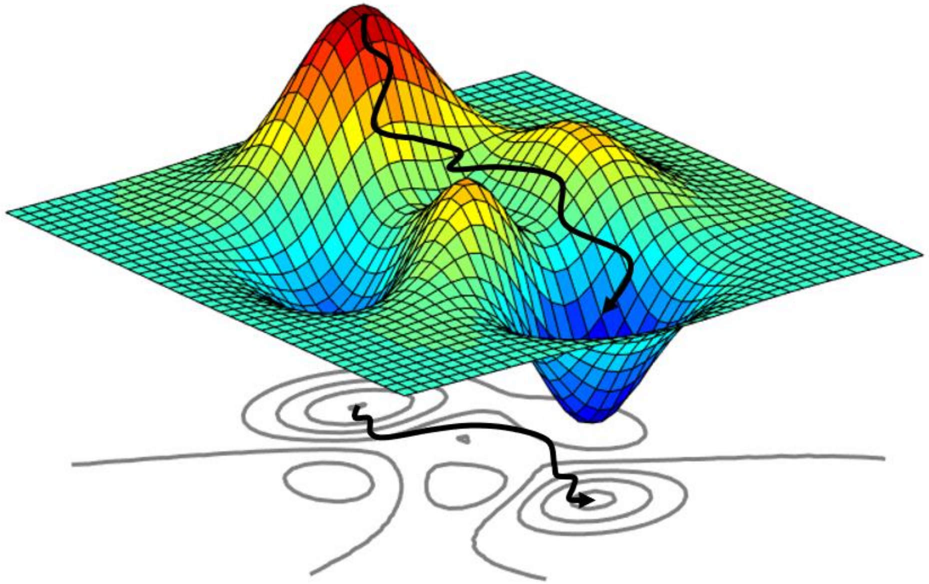
Rethinking Optimization

$$x \in \arg\max \;\; f^*(\cdot)$$



Rethinking Optimization

$$x \in \arg\max \;\; f^*(\cdot)$$

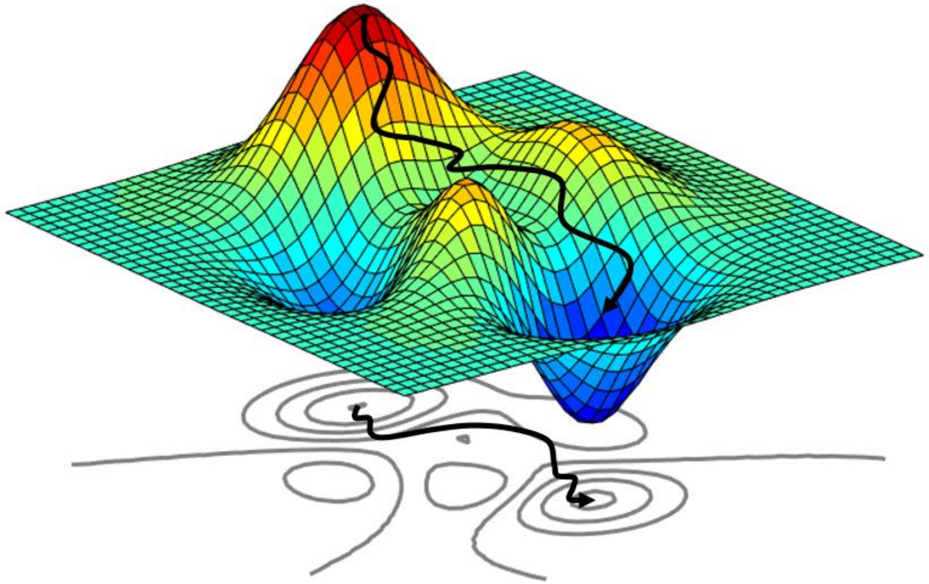


High-D

Nonconvex

Rethinking Optimization

$$x \in \arg\max \ f^*(\cdot)$$



High-D

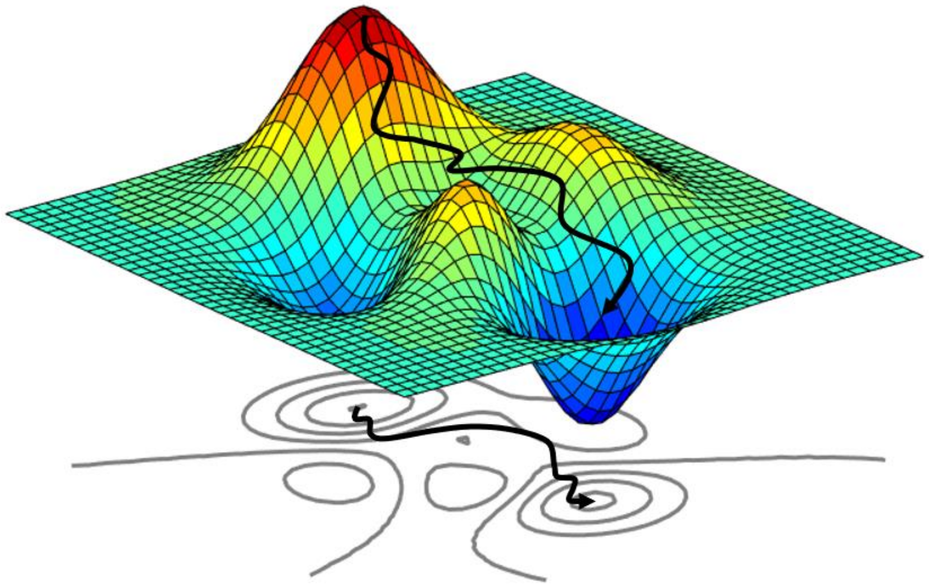
Nonconvex

Generate solution

$$x \sim \mathbb{P}(\cdot|\ f^*(\cdot) \geq a)$$

Rethinking Optimization

$$x \in \arg\max \; f^*(\cdot)$$



High-D

Nonconvex

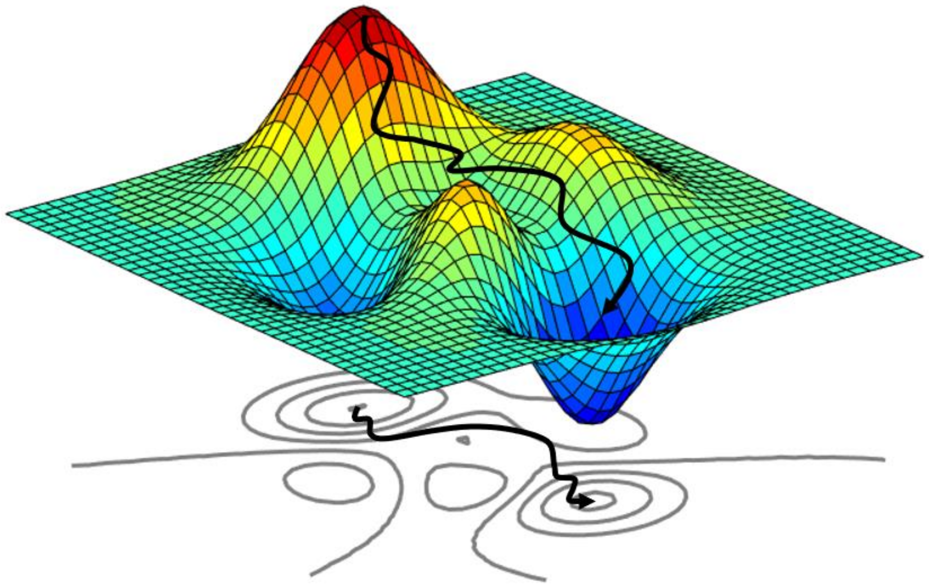
Generative Optimization

Generate solution

$$x \sim \mathbb{P}(\cdot | \; f^*(\cdot) \geq a)$$

Rethinking Optimization

$$x \in \arg\max \ \ f^*(\cdot)$$



High-D

Nonconvex

Generative Optimization

Generate solution

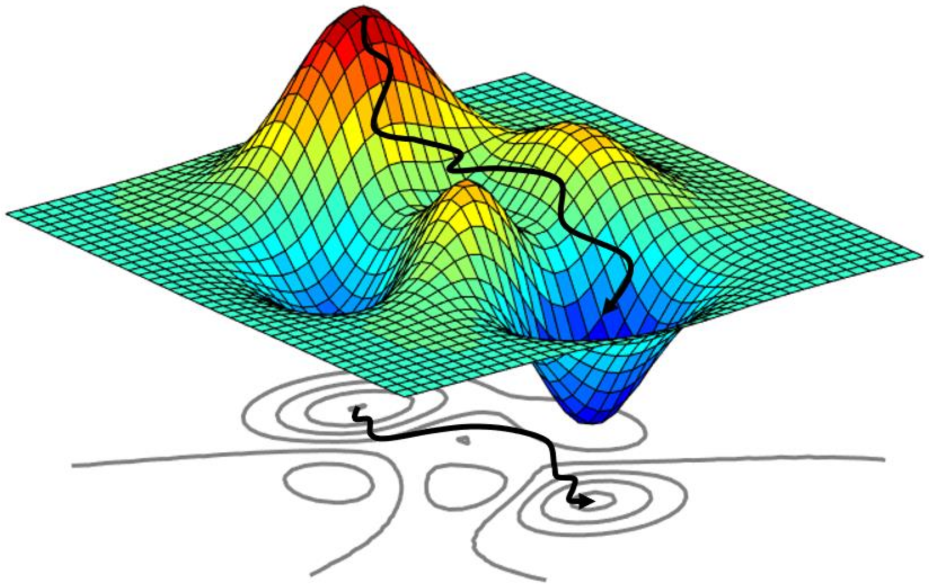
$$x \sim \mathbb{P}(\cdot | \ f^*(\cdot) \geq a)$$

High-D

Conditional distribution

Rethinking Optimization

$$x \in \arg\max \; f^*(\cdot)$$



Generative Optimization

Generate solution

$$x \sim \mathbb{P}(\cdot | f^*(\cdot) \geq a)$$

Guidance

High-D

High-D

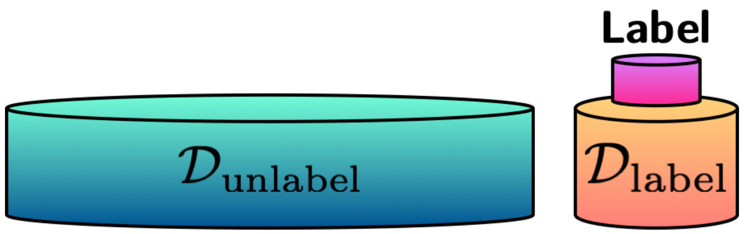
Nonconvex

Conditional distribution

Problem Setup: Offline Reward Maximization

- Given a training data set, generate new x
- Training data set

$$\mathcal{D}_{\text{unlabel}} = \{x_j\}_{j=1}^{n_{\text{unlabel}}}$$
$$\mathcal{D}_{\text{label}} = \{x_i, y_i = f^*(x_i) + \epsilon_i\}_{i=1}^{n_{\text{label}}}$$



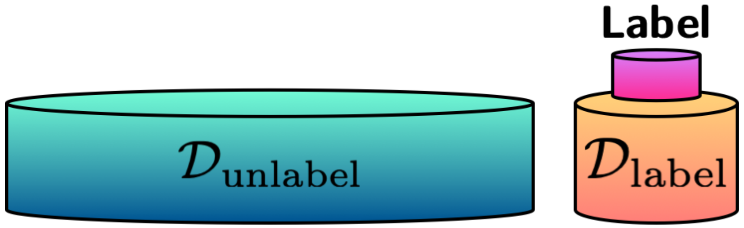
  - ϵ_i is observation noise
  - f^* is reward function

Problem Setup: Offline Reward Maximization

- Given a training data set, generate new x
- Training data set

$$\mathcal{D}_{\text{unlabel}} = \{x_j\}_{j=1}^{n_{\text{unlabel}}}$$

$$\mathcal{D}_{\text{label}} = \{x_i, y_i = f^*(x_i) + \epsilon_i\}_{i=1}^{n_{\text{label}}}$$



  - ϵ_i is observation noise
  - f^* is reward function

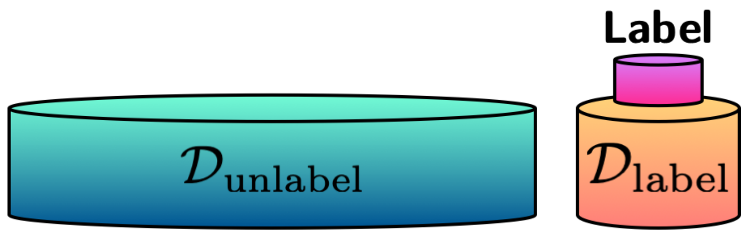
❑ Example: a large collection of unlabeled protein structures; only a few has measured properties.

Problem Setup: Offline Reward Maximization

- Given a training data set, generate new x
- Training data set

$$\mathcal{D}_{\text{unlabel}} = \{x_j\}_{j=1}^{n_{\text{unlabel}}}$$

$$\mathcal{D}_{\text{label}} = \{x_i, y_i = f^*(x_i) + \epsilon_i\}_{i=1}^{n_{\text{label}}}$$

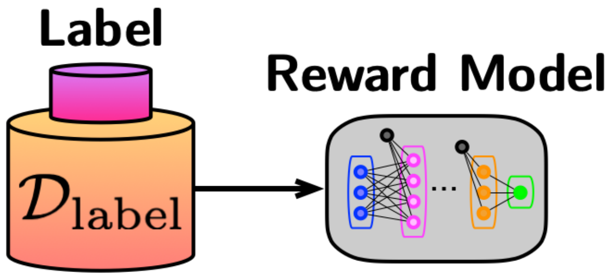


  - ➢ ϵ_i is observation noise
  - ➢ f^* is reward function

Off-policy bandit problem
(Jin et al., 2021; Nguyen-Tang et al., 2021)

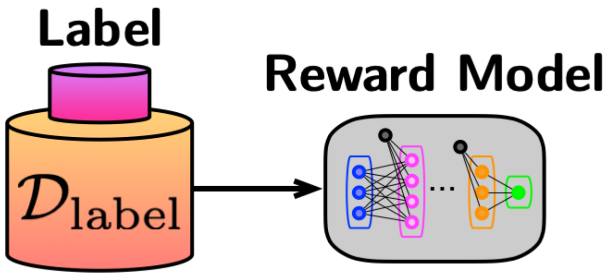
❑ Example: a large collection of unlabeled protein structures; only a few has measured properties.

Meta Algorithm

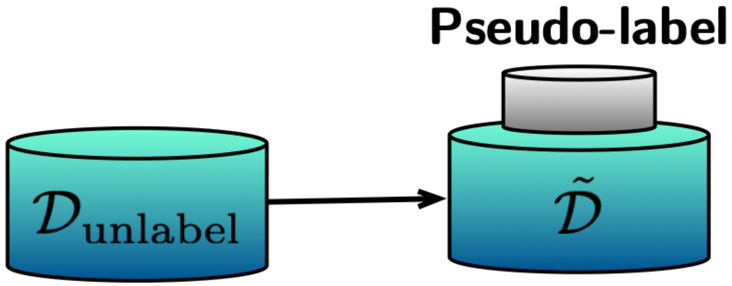


Step 1: Reward Learning

Meta Algorithm

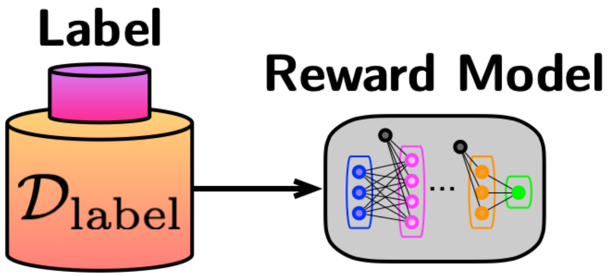


Step 1: Reward Learning

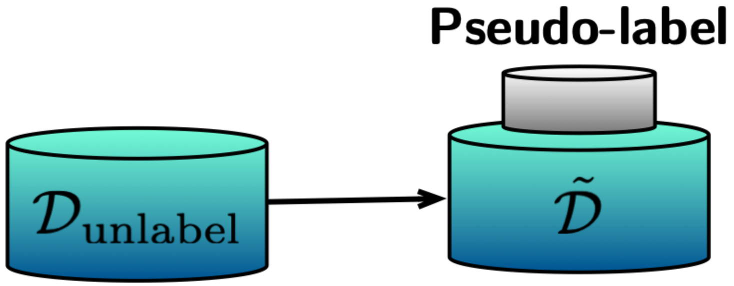


Step 2: Pseudo Labeling

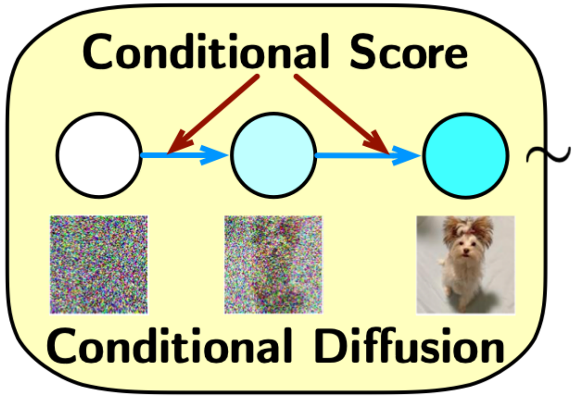
Meta Algorithm



Step 1: Reward Learning



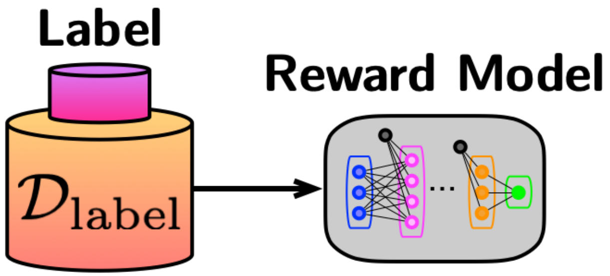
Step 2: Pseudo Labeling



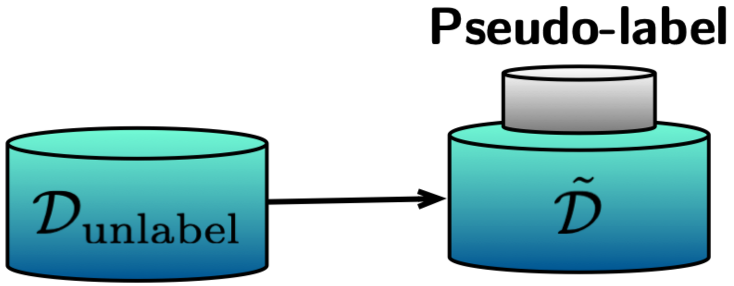
$$\sim \hat{\mathbb{P}}(\cdot \mid \text{reward})$$

Step 3: Conditional Diffusion Training

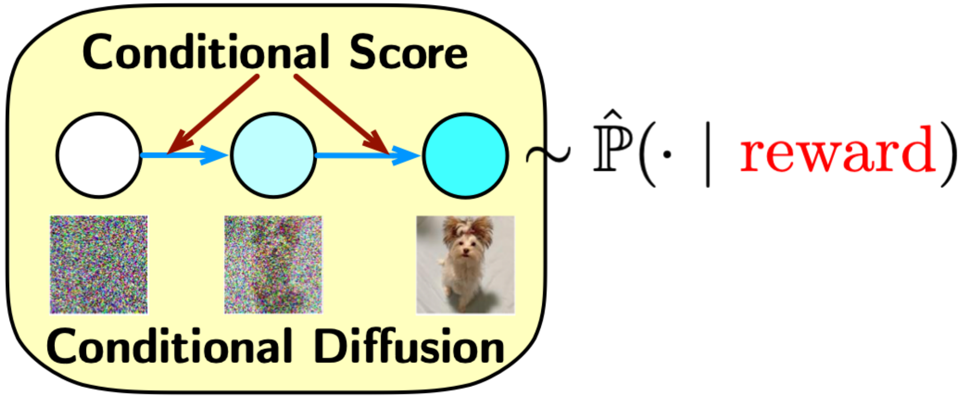
Meta Algorithm



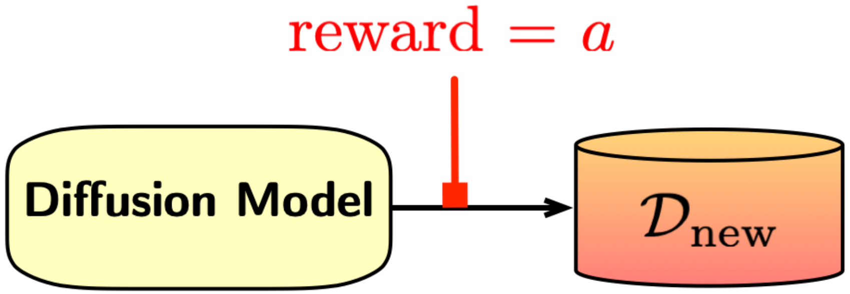
Step 1: Reward Learning



Step 2: Pseudo Labeling



Step 3: Conditional Diffusion Training



Step 4: Guided Generation

How Far Are We from The Targeted Reward

- Let a be the target reward of generation

$$\texttt{SubOpt}(a) = a - \texttt{Generated Average Reward}$$

How Far Are We from The Targeted Reward

- Let a be the target reward of generation

$$\texttt{SubOpt}(a) = a - \texttt{Generated Average Reward}$$

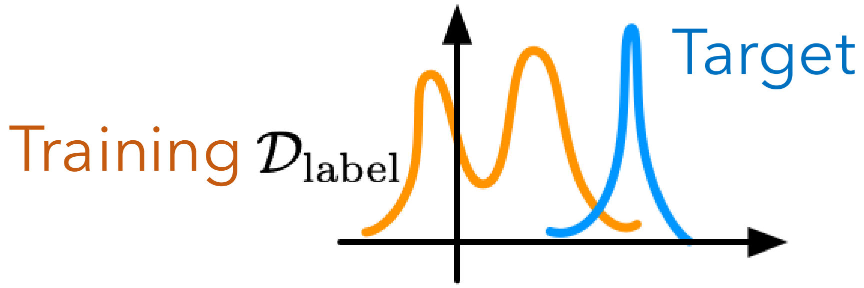
(Reward estimation error)

How Far Are We from The Targeted Reward

- Let a be the target reward of generation

$$\texttt{SubOpt}(a) = a - \texttt{Generated Average Reward}$$

(Reward estimation error) $\circ$ (Reward distribution shift)



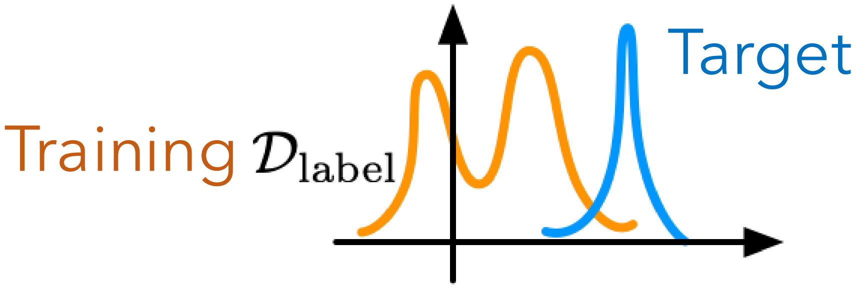
How Far Are We from The Targeted Reward

- Let a be the target reward of generation

$$\texttt{SubOpt}(a) = a - \texttt{Generated Average Reward}$$

(Reward estimation error) ∘ (Reward distribution shift)

(Conditional diffusion error)



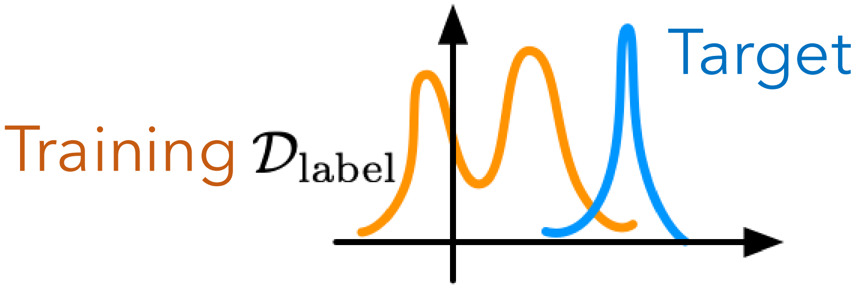
How Far Are We from The Targeted Reward

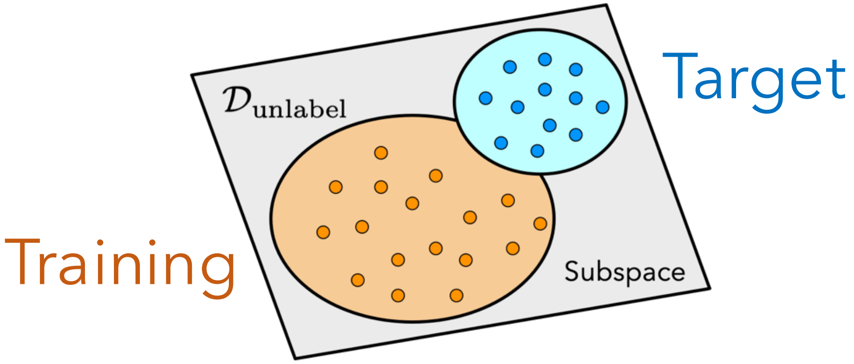
- Let a be the target reward of generation

$$\texttt{SubOpt}(a) = a - \texttt{Generated Average Reward}$$

(Reward estimation error) $\circ$ (Reward distribution shift)

(Conditional diffusion error) $\circ$ (Diffusion distribution shift)





Case Study: Subspace Data + Linear Reward

Theorem

✓ The sub-optimality satisfies

$$\texttt{SubOpt}(a) = \tilde{\mathcal{O}}\left(\sqrt{\mathrm{Trace}\left(\hat{\Sigma}_\lambda^{-1}\Sigma_a\right)} \cdot \sqrt{\frac{d\log(n_{\mathrm{label}})}{n_{\mathrm{label}}}} + \min\{a, d\} \cdot \frac{a \cdot \mathrm{poly}(D, d)}{n_{\mathrm{unlabel}}^{1/6}}\right)$$

where $\hat{\Sigma}_\lambda = (X^\top X + \lambda I)/n_{\mathrm{label}}$ for X the data matrix, $\lambda > 0$, and Σ_a is the covariance matrix of $P_a(\cdot \mid \mathrm{reward} = a)$.

-- H. Yuan, K. Huang, C. Ni, Y. Ye, M. Chen, M. Wang. "Diffusion Model for Data-Driven Black-Box Optimization", NeurIPS 2023

Case Study: Subspace Data + Linear Reward

Theorem

✓ The sub-optimality satisfies

$$\texttt{SubOpt}(a) = \tilde{\mathcal{O}}\left(\sqrt{\mathrm{Trace}\left(\hat{\Sigma}_\lambda^{-1}\Sigma_a\right)} \cdot \sqrt{\frac{d\log(n_{\mathrm{label}})}{n_{\mathrm{label}}}} + \min\{a, d\} \cdot \frac{a \cdot \mathrm{poly}(D, d)}{n_{\mathrm{unlabel}}^{1/6}}\right)$$

where $\hat{\Sigma}_\lambda = (X^\top X + \lambda I)/n_{\mathrm{label}}$ for X the data matrix, $\lambda > 0$, and Σ_a is the covariance matrix of $P_a(\cdot \mid \mathrm{reward} = a)$.

-- H. Yuan, K. Huang, C. Ni, Y. Ye, M. Chen, M. Wang. "Diffusion Model for Data-Driven Black-Box Optimization", NeurIPS 2023

Case Study: Subspace Data + Linear Reward

Theorem

✓ The sub-optimality satisfies

$$\texttt{SubOpt}(a) = \tilde{\mathcal{O}}\left(\sqrt{\mathrm{Trace}\left(\hat{\Sigma}_\lambda^{-1}\Sigma_a\right)} \cdot \sqrt{\frac{d\log(n_{\mathrm{label}})}{n_{\mathrm{label}}}} + \min\{a,d\} \cdot \frac{a \cdot \mathrm{poly}(D,d)}{n_{\mathrm{unlabel}}^{1/6}}\right)$$

where $\hat{\Sigma}_\lambda = (X^\top X + \lambda I)/n_{\mathrm{label}}$ for X the data matrix, $\lambda > 0$, and Σ_a is the covariance matrix of $P_a(\cdot \mid \mathrm{reward} = a)$.

-- H. Yuan, K. Huang, C. Ni, Y. Ye, M. Chen, M. Wang. "Diffusion Model for Data-Driven Black-Box Optimization", NeurIPS 2023

Case Study: Subspace Data + Linear Reward

Theorem

✓ The sub-optimality satisfies

$$\texttt{SubOpt}(a) = \tilde{\mathcal{O}}\left(\sqrt{\mathrm{Trace}\left(\hat{\Sigma}_\lambda^{-1}\Sigma_a\right)} \cdot \sqrt{\frac{d\log(n_{\mathrm{label}})}{n_{\mathrm{label}}}} + \quad\right)$$

where $\hat{\Sigma}_\lambda = (X^\top X + \lambda I)/n_{\mathrm{label}}$ for X the data matrix, $\lambda > 0$, and Σ_a is the covariance matrix of $P_a(\cdot \mid \mathrm{reward} = a)$.

-- H. Yuan, K. Huang, C. Ni, Y. Ye, M. Chen, M. Wang. "Diffusion Model for Data-Driven Black-Box Optimization", NeurIPS 2023

Case Study: Subspace Data + Linear Reward

Theorem

✓ The sub-optimality satisfies

$$\mathtt{SubOpt}(a) = \tilde{\mathcal{O}}\left(\sqrt{\mathrm{Trace}\left(\hat{\Sigma}_\lambda^{-1}\Sigma_a\right)} \cdot \sqrt{\frac{d\log(n_{\mathrm{label}})}{n_{\mathrm{label}}}} + \boxed{}\right)$$

where $\hat{\Sigma}_\lambda = (X^\top X + \lambda I)/n_{\mathrm{label}}$ for X the data matrix, $\lambda > 0$, and Σ_a is the covariance matrix of $P_a(\cdot \mid \mathrm{reward} = a)$.

❖ Match optimal off-policy bandit learning with representation learning (Jin et al., 2021; Nguyen-Tang et al., 2021)

-- H. Yuan, K. Huang, C. Ni, Y. Ye, M. Chen, M. Wang. "Diffusion Model for Data-Driven Black-Box Optimization", NeurIPS 2023

Advantages of Generative Optimization

✓ Meta algorithm provably generates samples of high reward and fidelity, in nonparametric settings.

$$\texttt{SubOpt}(a) = \tilde{O}\left(\kappa_1(a) \cdot n_{\text{label}}^{-\frac{\alpha}{d+2\alpha}} + \kappa_2(a) \cdot n_{\text{unlabel}}^{-\frac{2}{3(d+6)}}\right)$$

-- H. Yuan, Z. Li, K. Huang, C. Ni, Y. Ye, M. Chen, M. Wang. "Diffusion Model for Data-Driven Black-Box Optimization", Major revision at Management Science

Advantages of Generative Optimization

✓ Meta algorithm provably generates samples of high reward and fidelity, in nonparametric settings.

$$\texttt{SubOpt}(a) = \tilde{O}\left(\kappa_1(a) \cdot n_{\text{label}}^{-\frac{\alpha}{d+2\alpha}} + \kappa_2(a) \cdot n_{\text{unlabel}}^{-\frac{2}{3(d+6)}}\right)$$

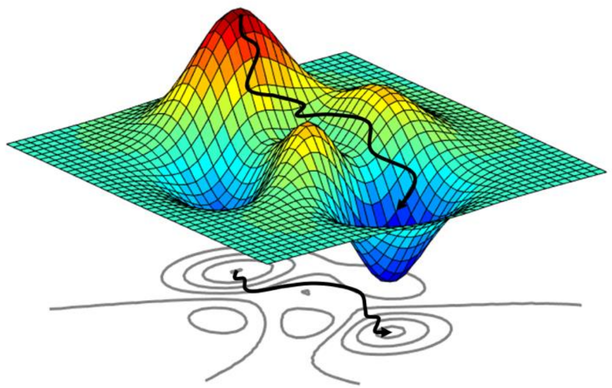
Generative optimization in offline:

✓ **Off-policy bandit optimality**

✓ **High-fidelity** to intrinsic structures

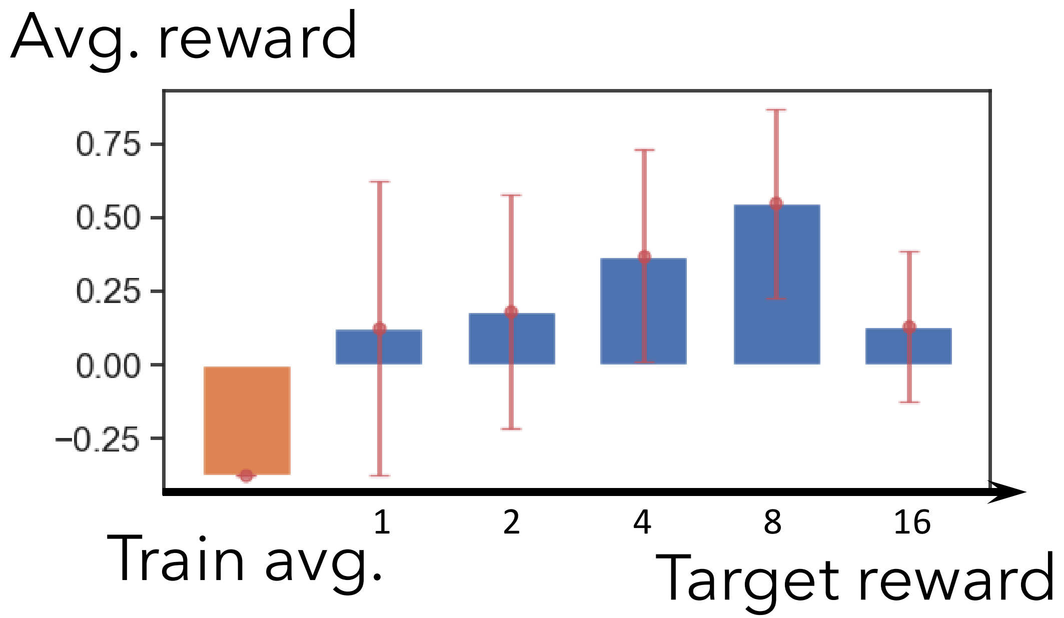
✓ **Efficiency**: no curse of dimensionality

✓ **Generalizable** to human preferences

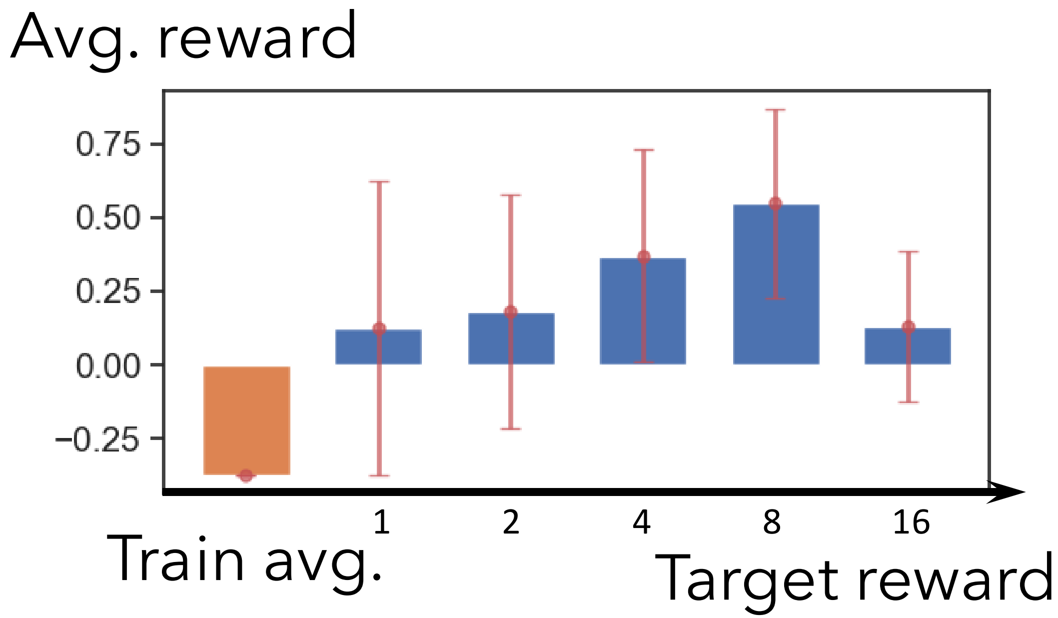


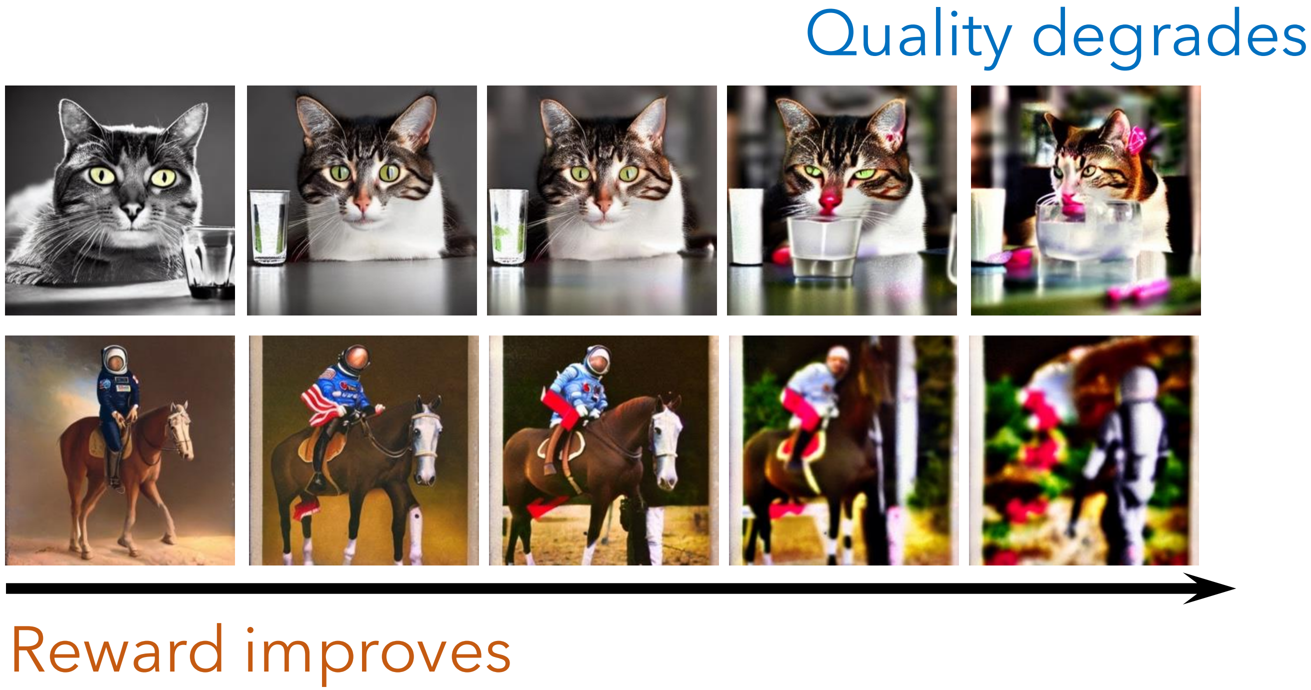
-- H. Yuan, Z. Li, K. Huang, C. Ni, Y. Ye, M. Chen, M. Wang. "Diffusion Model for Data-Driven Black-Box Optimization", Major revision at Management Science

Application 1: CIFAR Reward Optimization



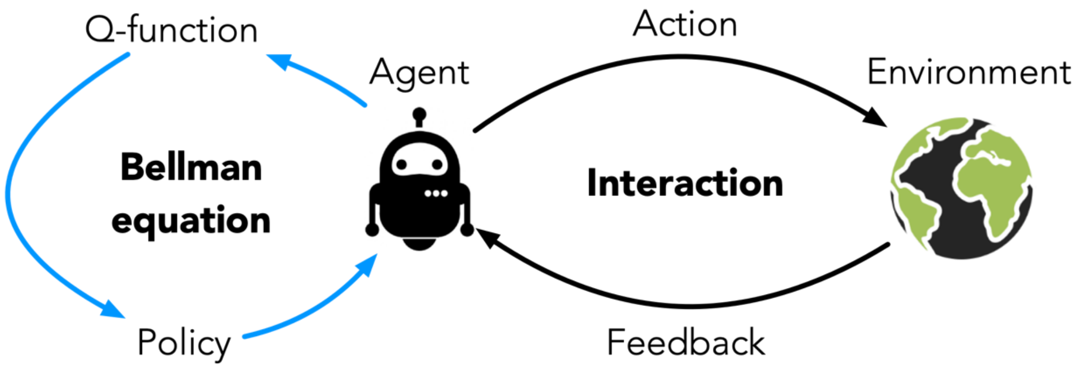
Application 1: CIFAR Reward Optimization





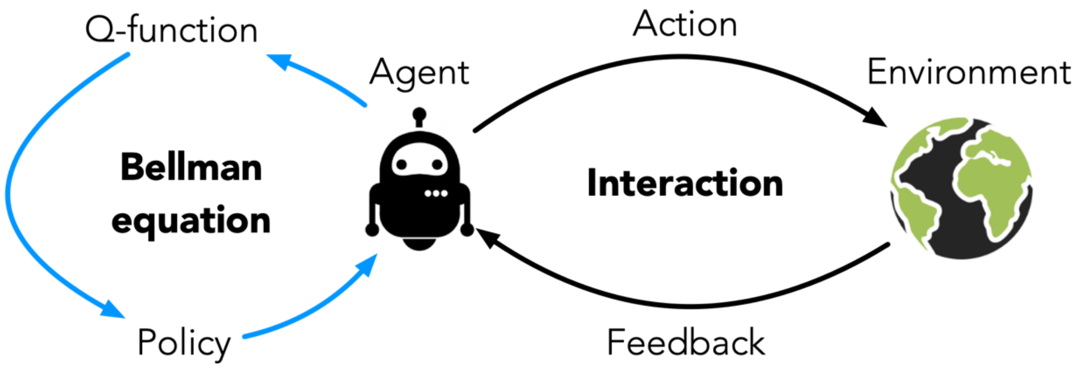
Application 2: Generative Optimization in RL

- Reinforcement learning

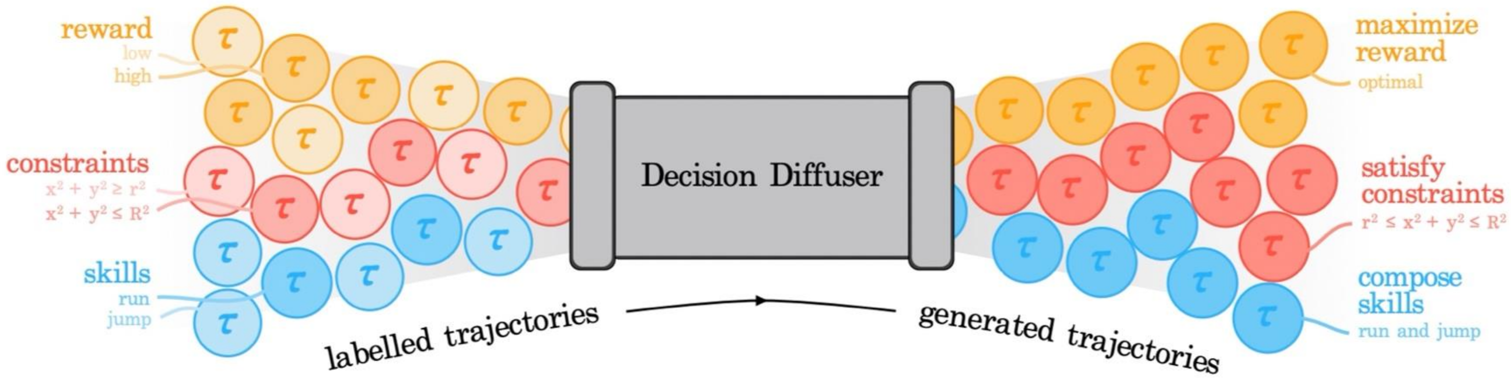


Application 2: Generative Optimization in RL

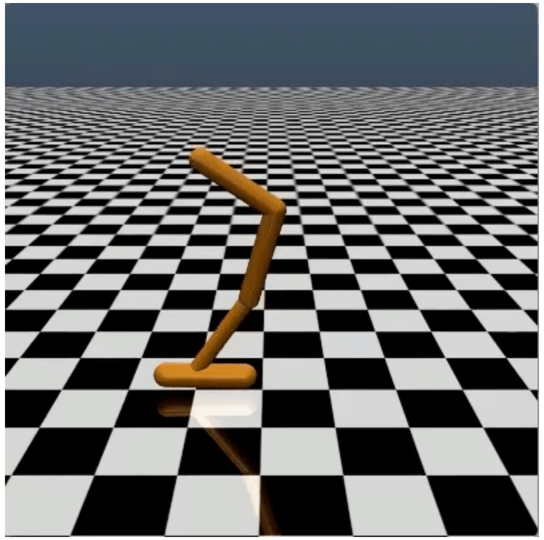
- Reinforcement learning

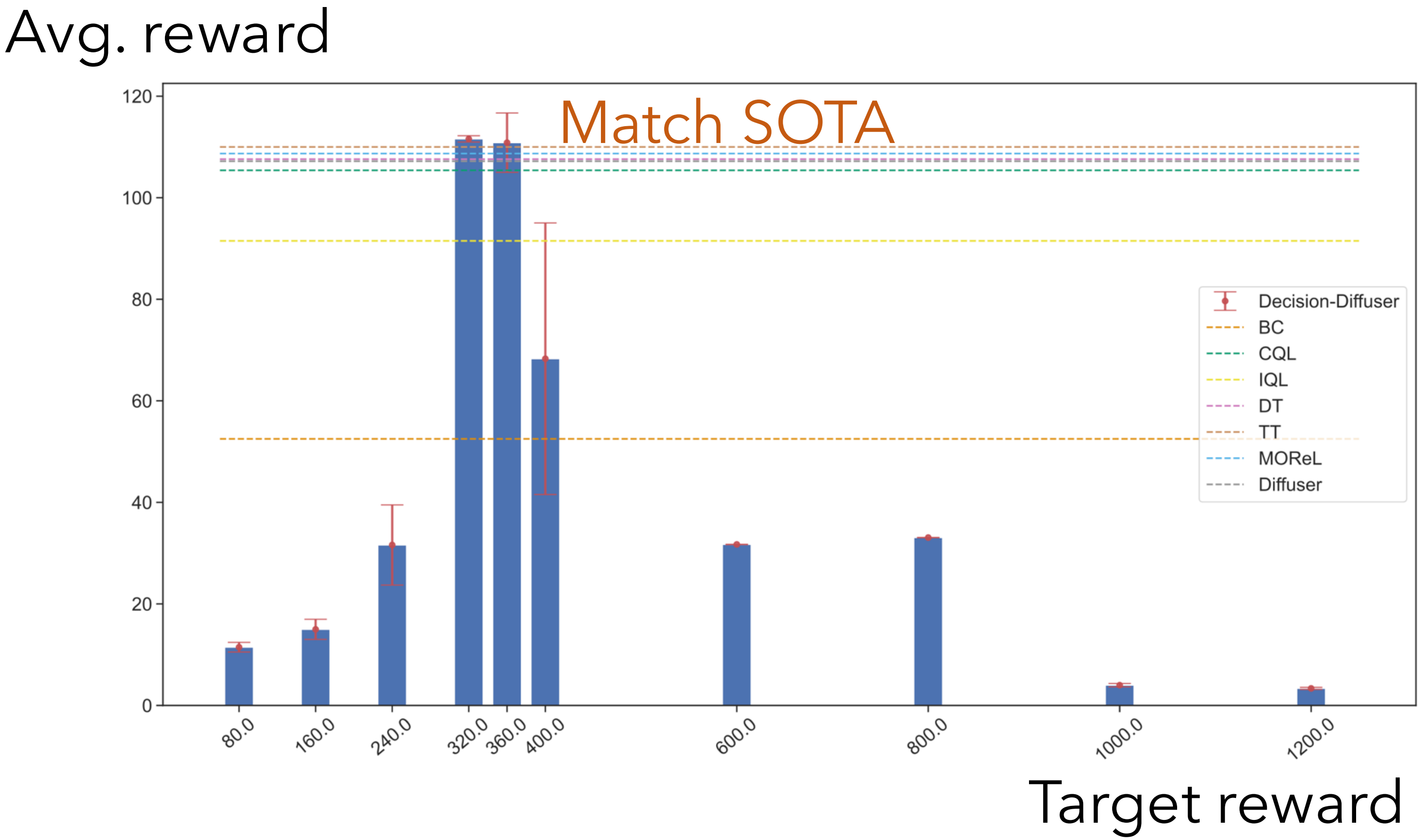


- Generative optimization (Decision diffuser, Ajay et al., 2023)



Hopper Control

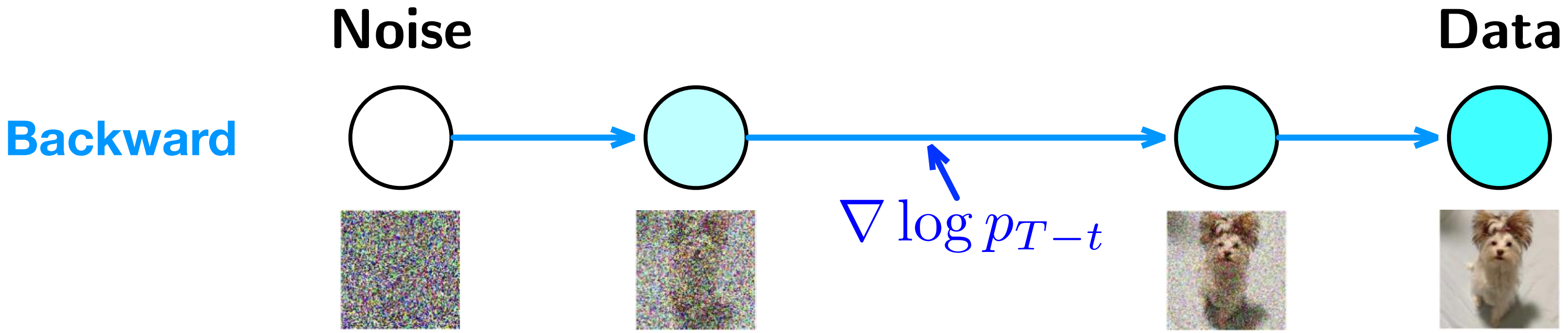




Inspirations and Future Directions

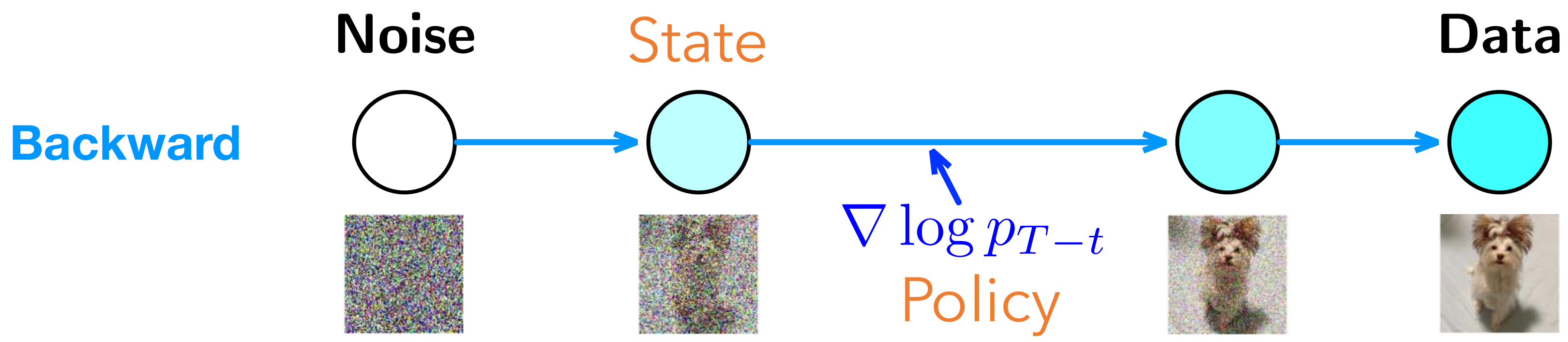
Control/RL Perspective on Diffusion Model

- We design backward process to be Markovian



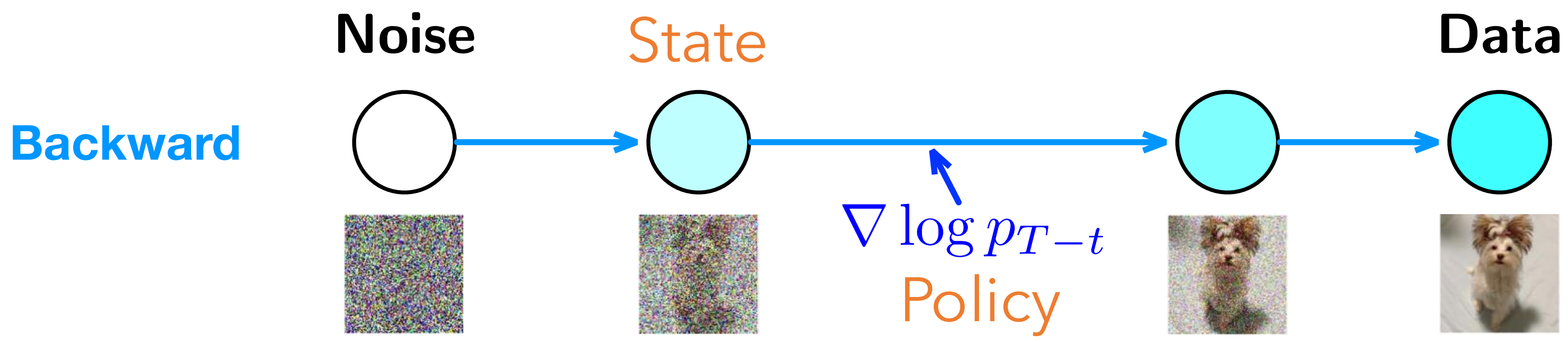
Control/RL Perspective on Diffusion Model

- We design backward process to be Markovian



Control/RL Perspective on Diffusion Model

- We design backward process to be Markovian

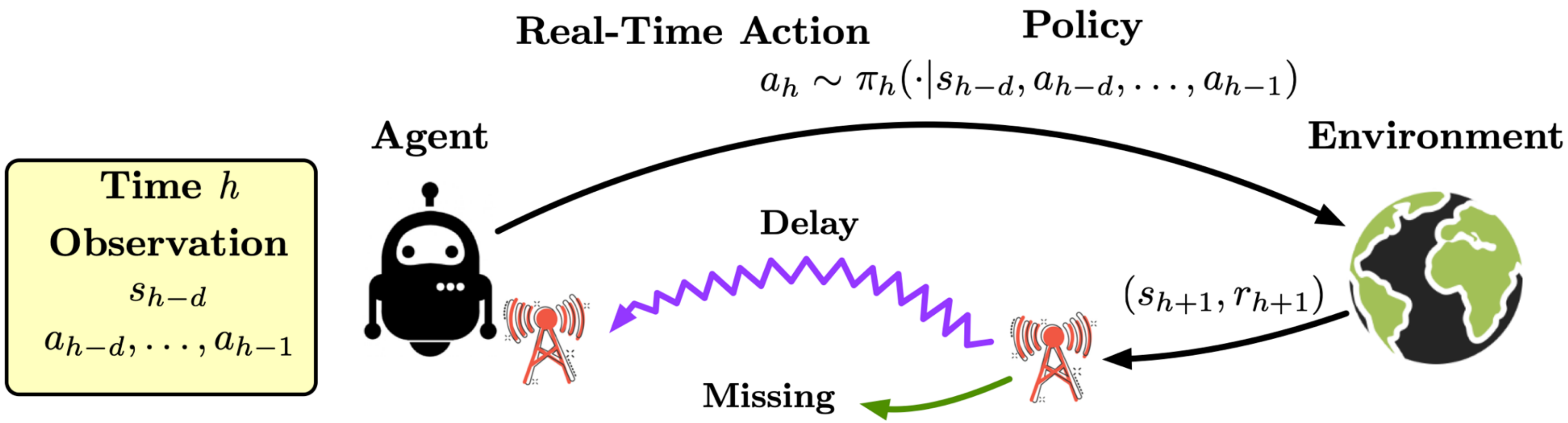


- Reward choice is task dependent
  - ➢ Fidelity metric for image generation
  - ➢ Satisfactory level for product design
  - ➢ Biochemical property for protein synthesis
  - ➢ Etc.

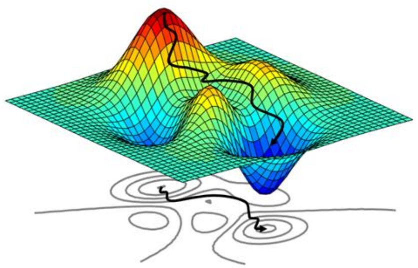
Diffusion Model for Control/RL

- Conditional diffusion models as a rich class for parameterizing **policies** and **transition kernels**

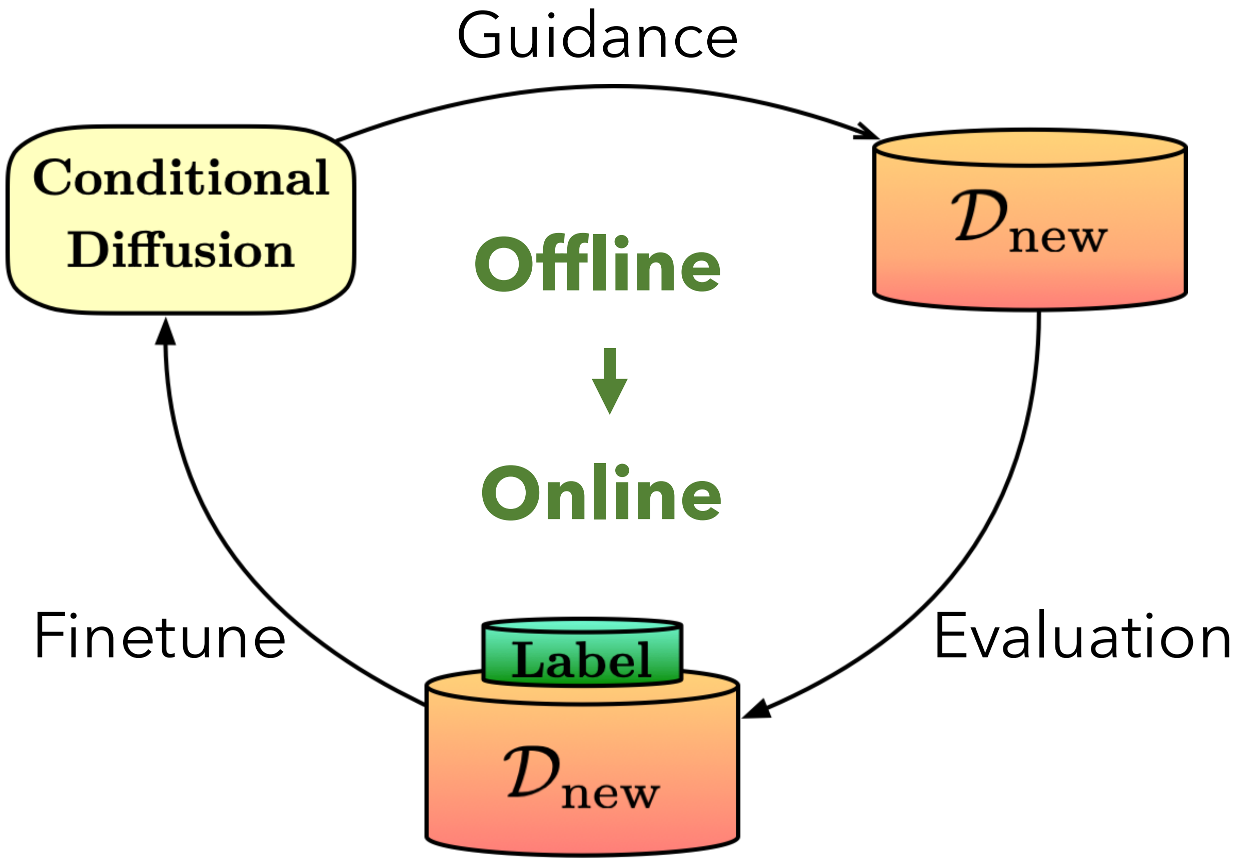
- Diffusion for practical RL with impaired observability (Chen et al., 2023)

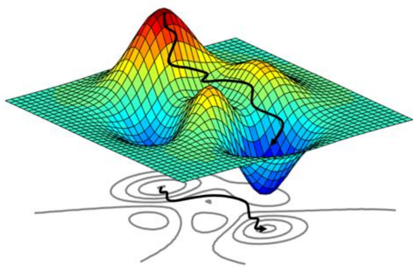


Diffusion Model for Generative Optimization

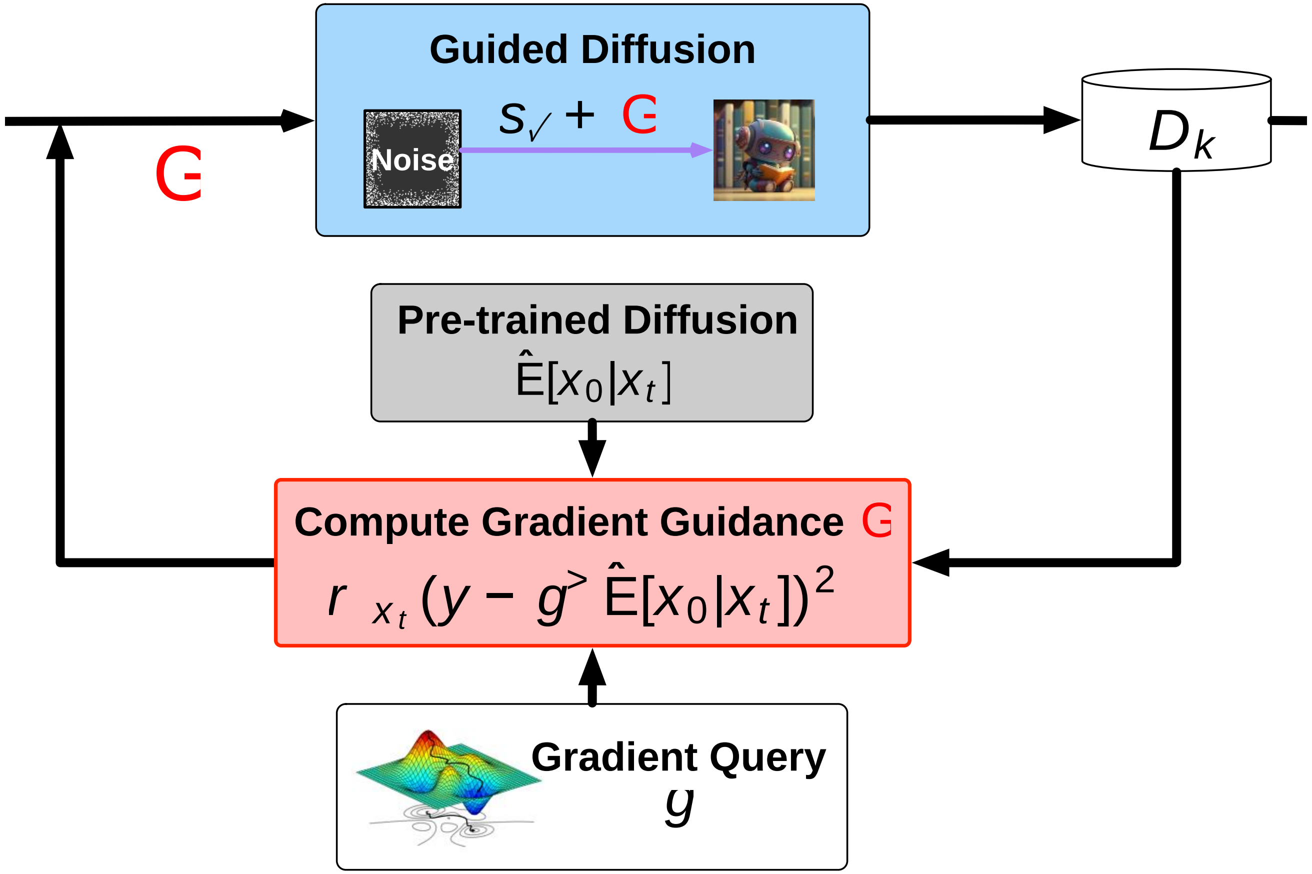


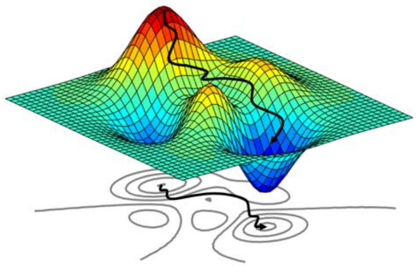
Diffusion Model for Generative Optimization





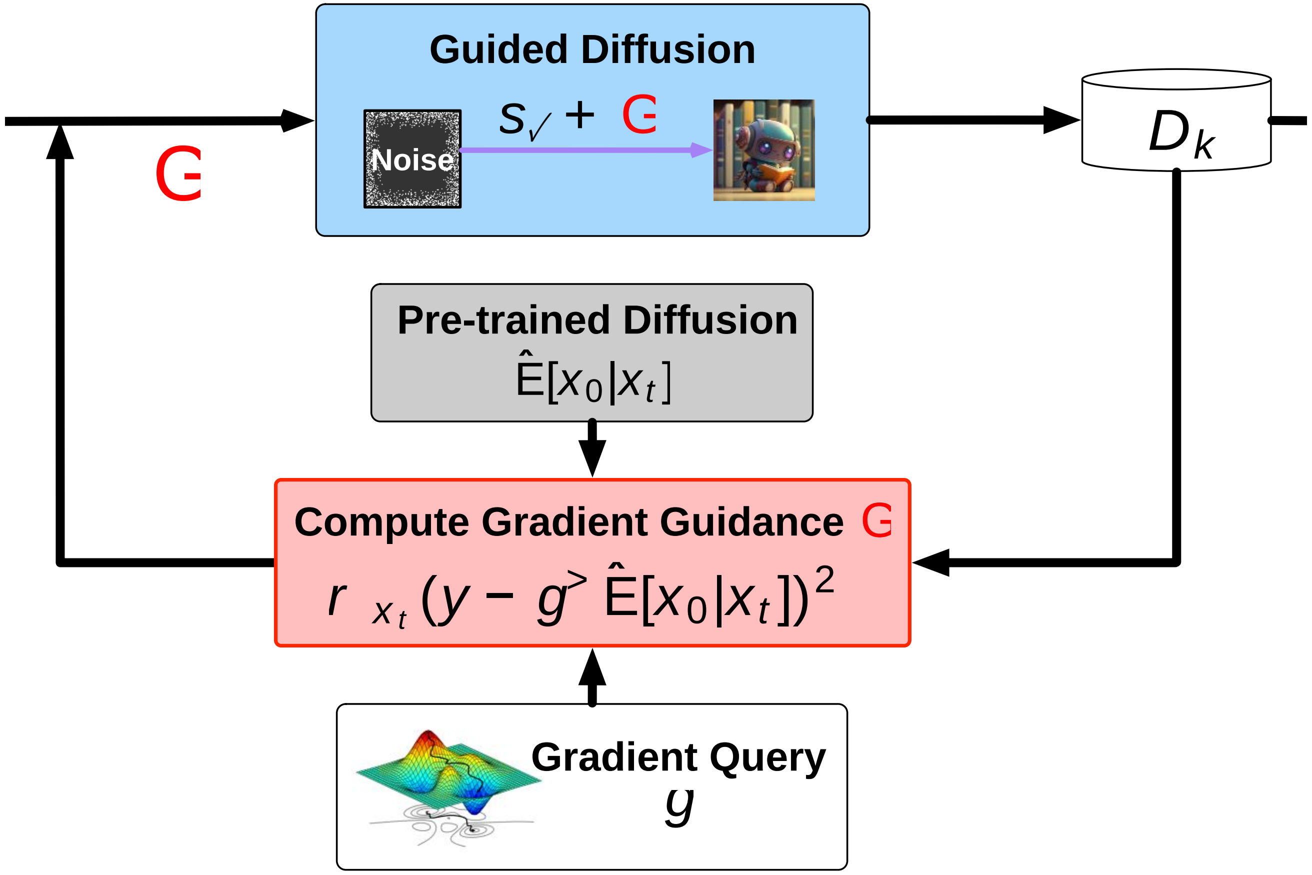
Diffusion Model for Generative Optimization

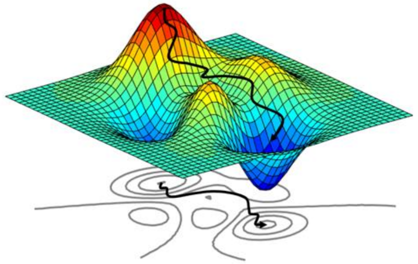




-- Y. Guo, H. Yuan, Y. Yang, M. Chen, M. Wang. "Gradient Guidance for Diffusion Models: An Optimization Perspective", NeurIPS 2024

Diffusion Model for Generative Optimization

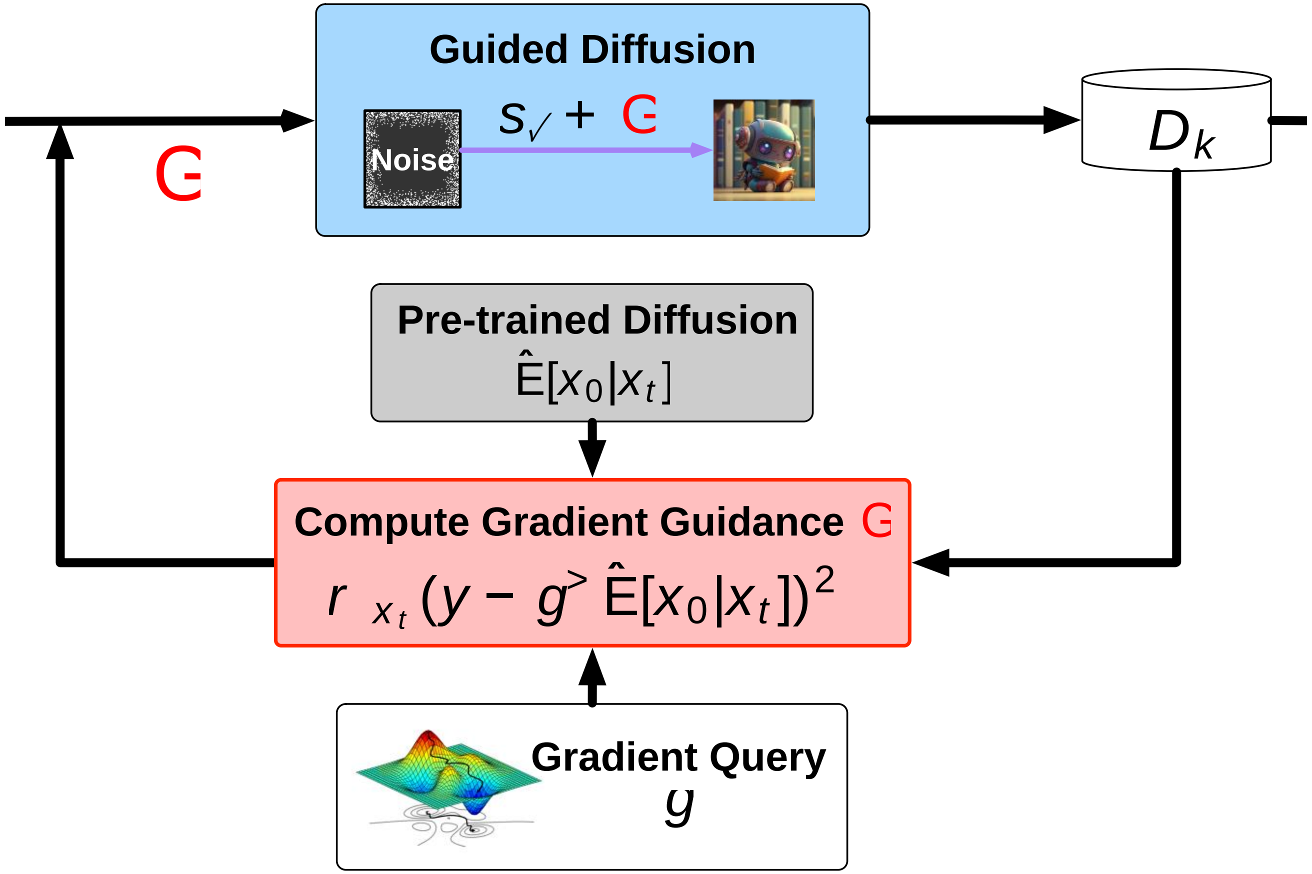


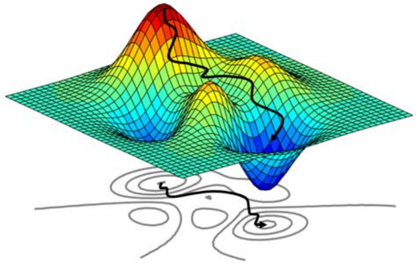


✓ High-D data

-- Y. Guo, H. Yuan, Y. Yang, M. Chen, M. Wang. "Gradient Guidance for Diffusion Models: An Optimization Perspective", NeurIPS 2024

Diffusion Model for Generative Optimization

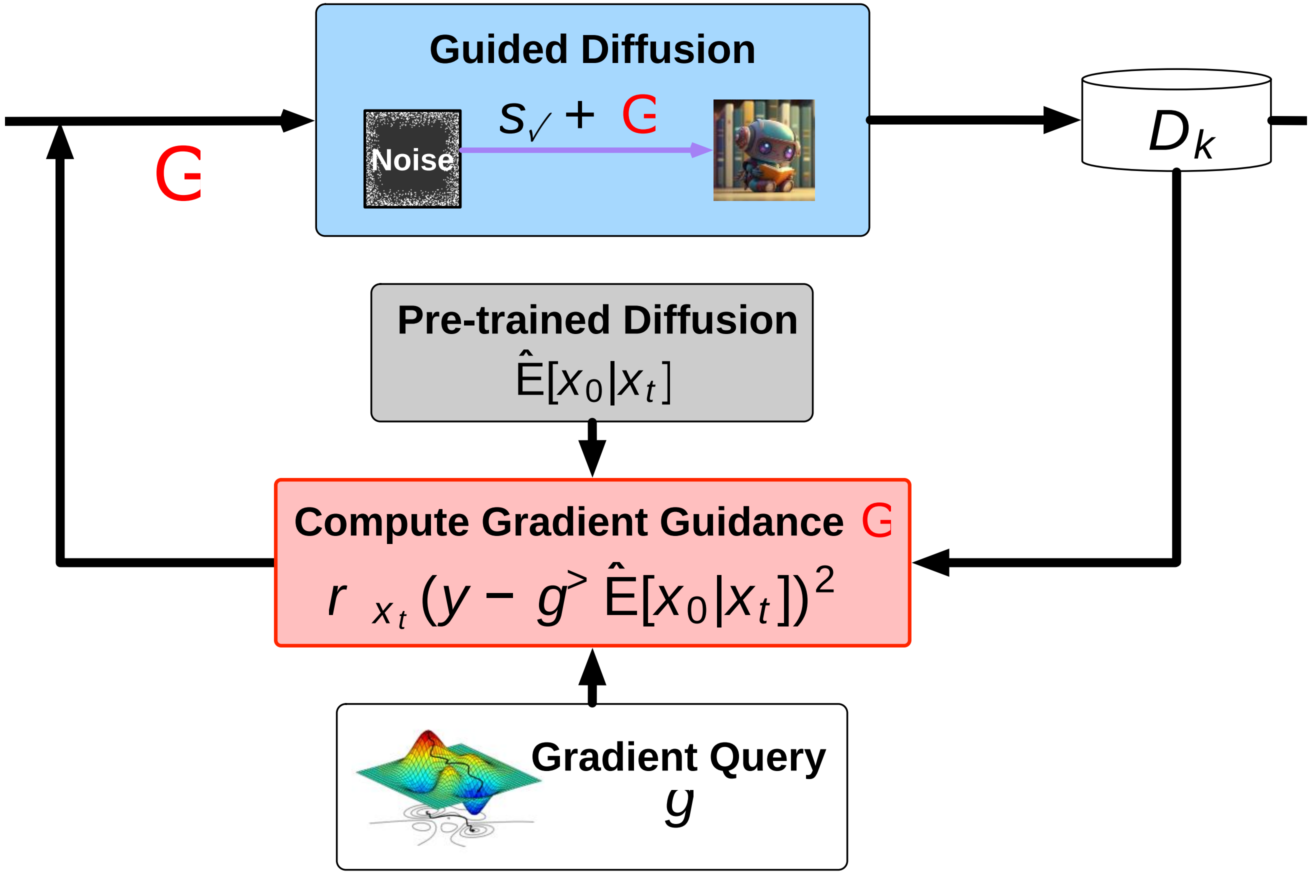


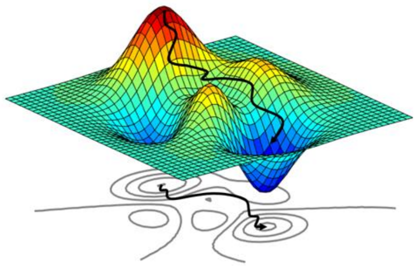


- ✓ High-D data
- ✓ Efficient adaptation

-- Y. Guo, H. Yuan, Y. Yang, M. Chen, M. Wang. "Gradient Guidance for Diffusion Models: An Optimization Perspective", NeurIPS 2024

Diffusion Model for Generative Optimization

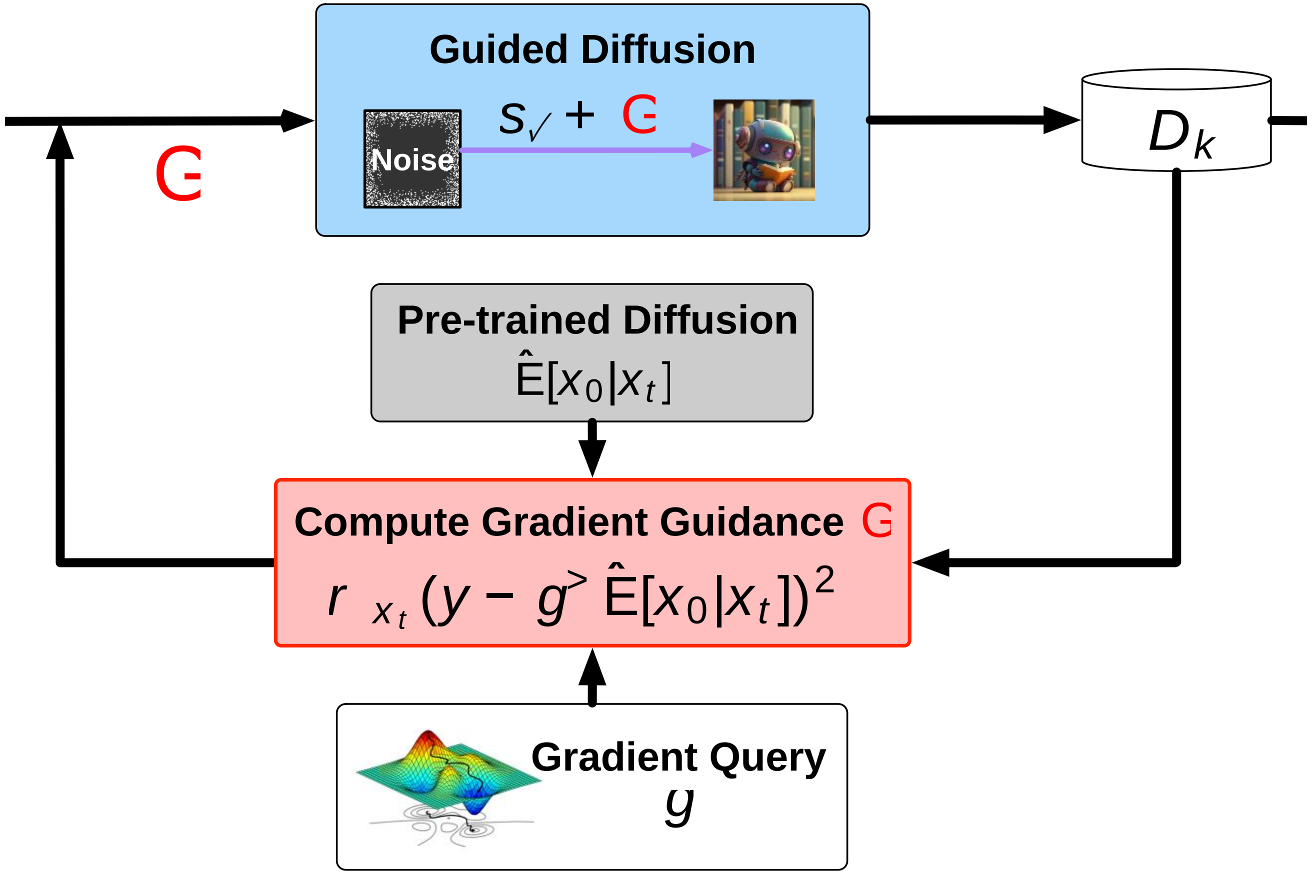


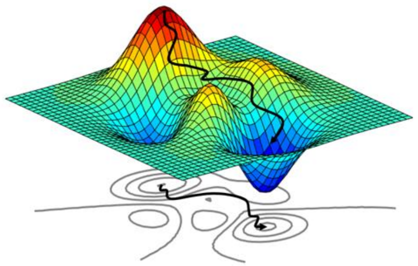


- ✓ High-D data
- ✓ Efficient adaptation
- ✓ Escape from bad local optima and saddle points

-- Y. Guo, H. Yuan, Y. Yang, M. Chen, M. Wang. "Gradient Guidance for Diffusion Models: An Optimization Perspective", NeurIPS 2024

Diffusion Model for Generative Optimization



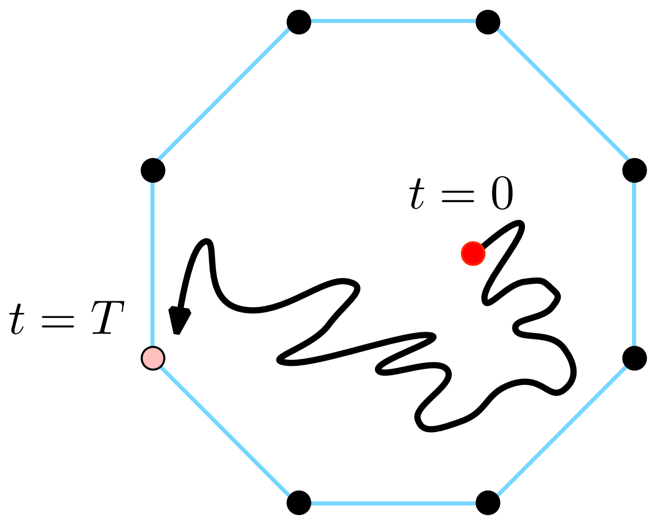


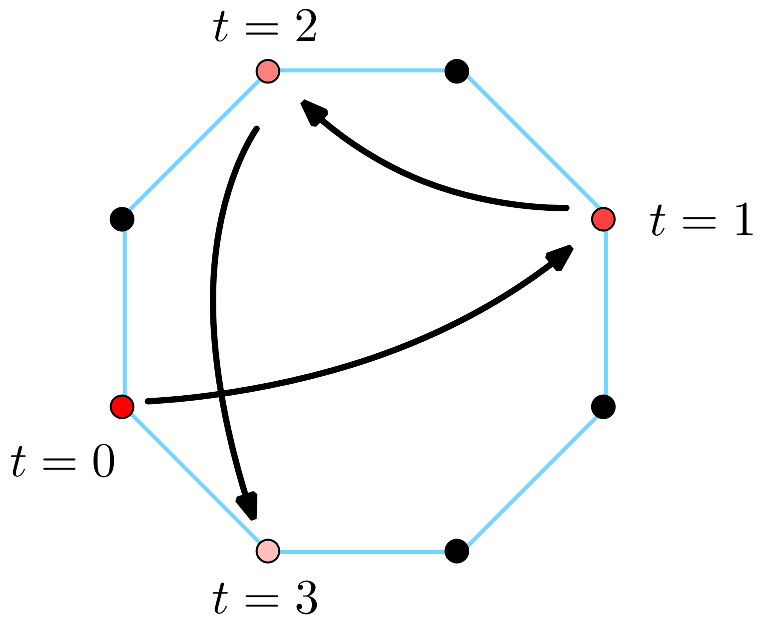
- ✓ High-D data
- ✓ Efficient adaptation
- ✓ Escape from bad local optima and saddle points
- ✓ Connecting to DRO

-- Y. Guo, H. Yuan, Y. Yang, M. Chen, M. Wang. "Gradient Guidance for Diffusion Models: An Optimization Perspective", NeurIPS 2024

Diffusion Model for Discrete Data

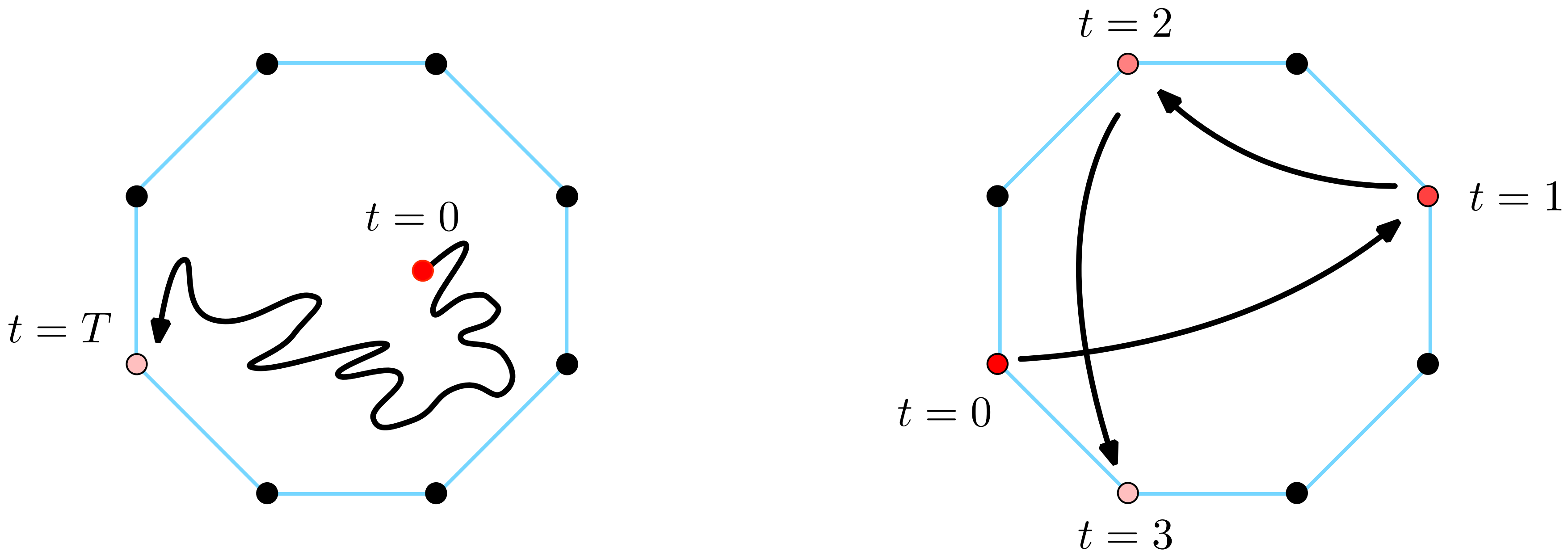
- Gaussian noise may not be suitable for discrete data





Diffusion Model for Discrete Data

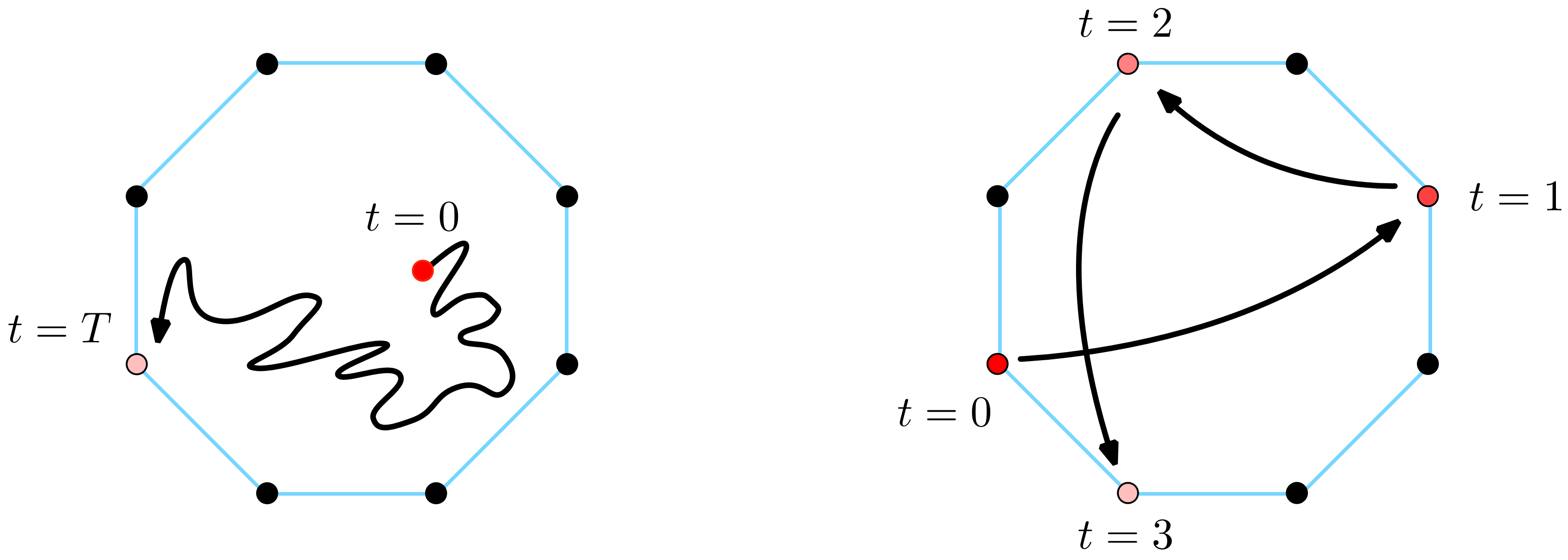
- Gaussian noise may not be suitable for discrete data



- Discrete diffusion jumps inside data support as "corruption"

Diffusion Model for Discrete Data

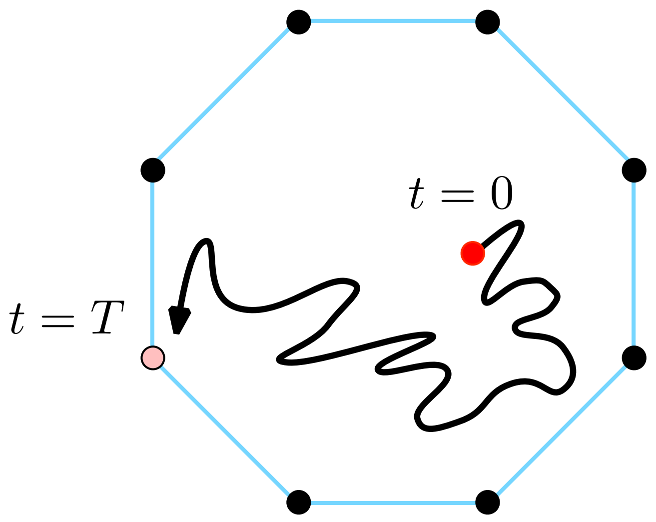
- Gaussian noise may not be suitable for discrete data

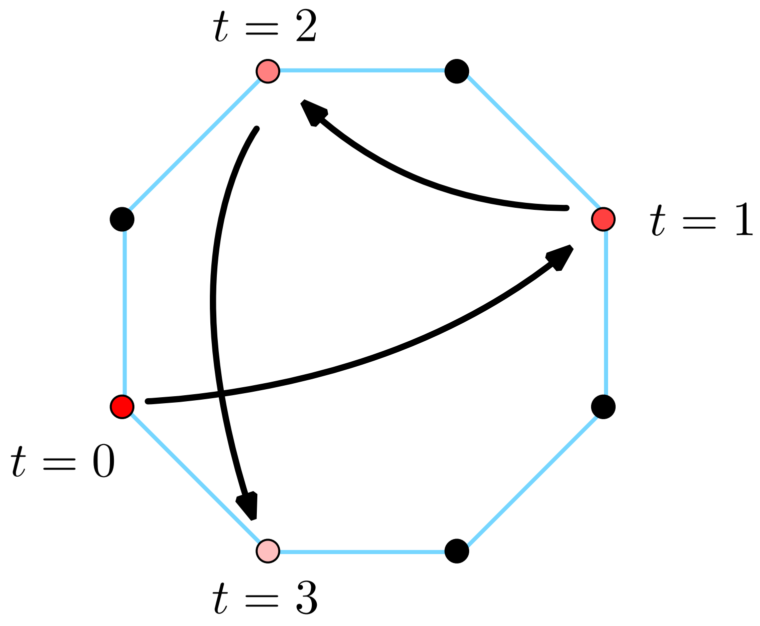


- Discrete diffusion jumps inside data support as "corruption"
  - ➢ Integer optimization

Diffusion Model for Discrete Data

- Gaussian noise may not be suitable for discrete data





- Discrete diffusion jumps inside data support as "corruption"
  - ➢ Integer optimization
  - ➢ Protein generation

Thank You!