An Optimization Perspective on Guidance for Fine-Tuning Diffusion Models

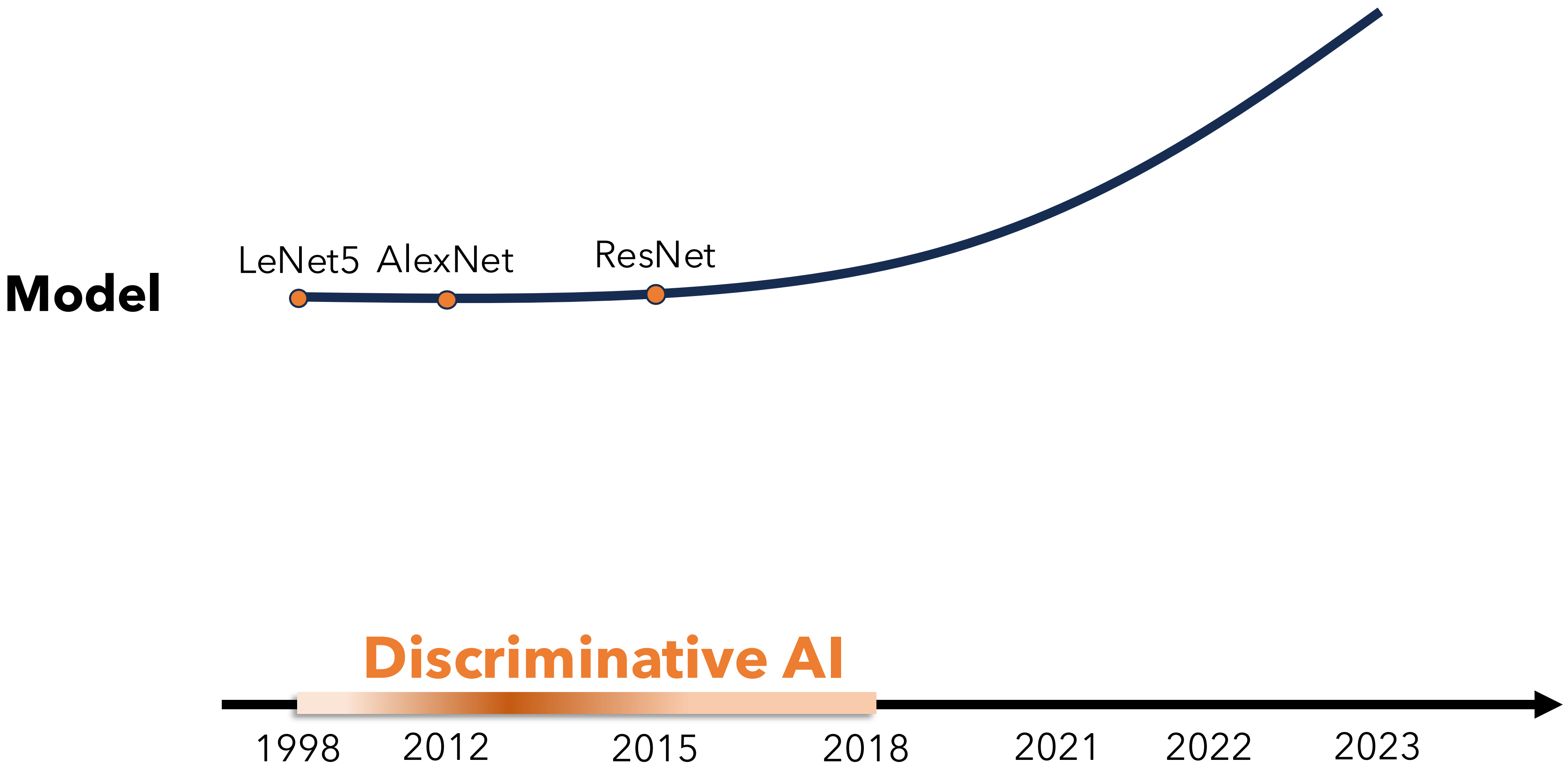
Minshuo Chen

Industrial Engineering & Management Sciences

Northwestern University

Joint work with Yingqing Guo, Zihao Li, Kaixuan Huang, Hui Yuan, Yukang Yang, Yinyu Ye, and Mengdi Wang

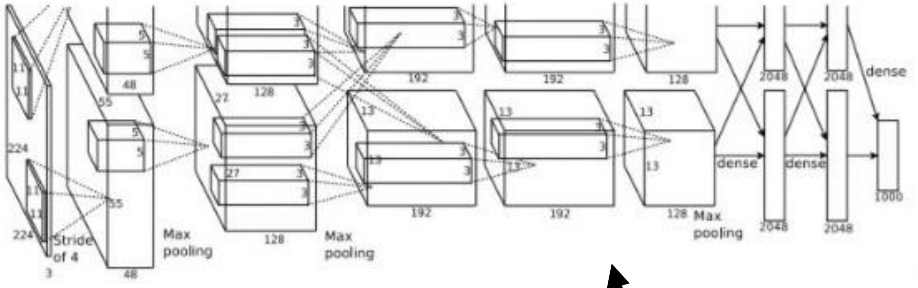
Millennium Growth of AI



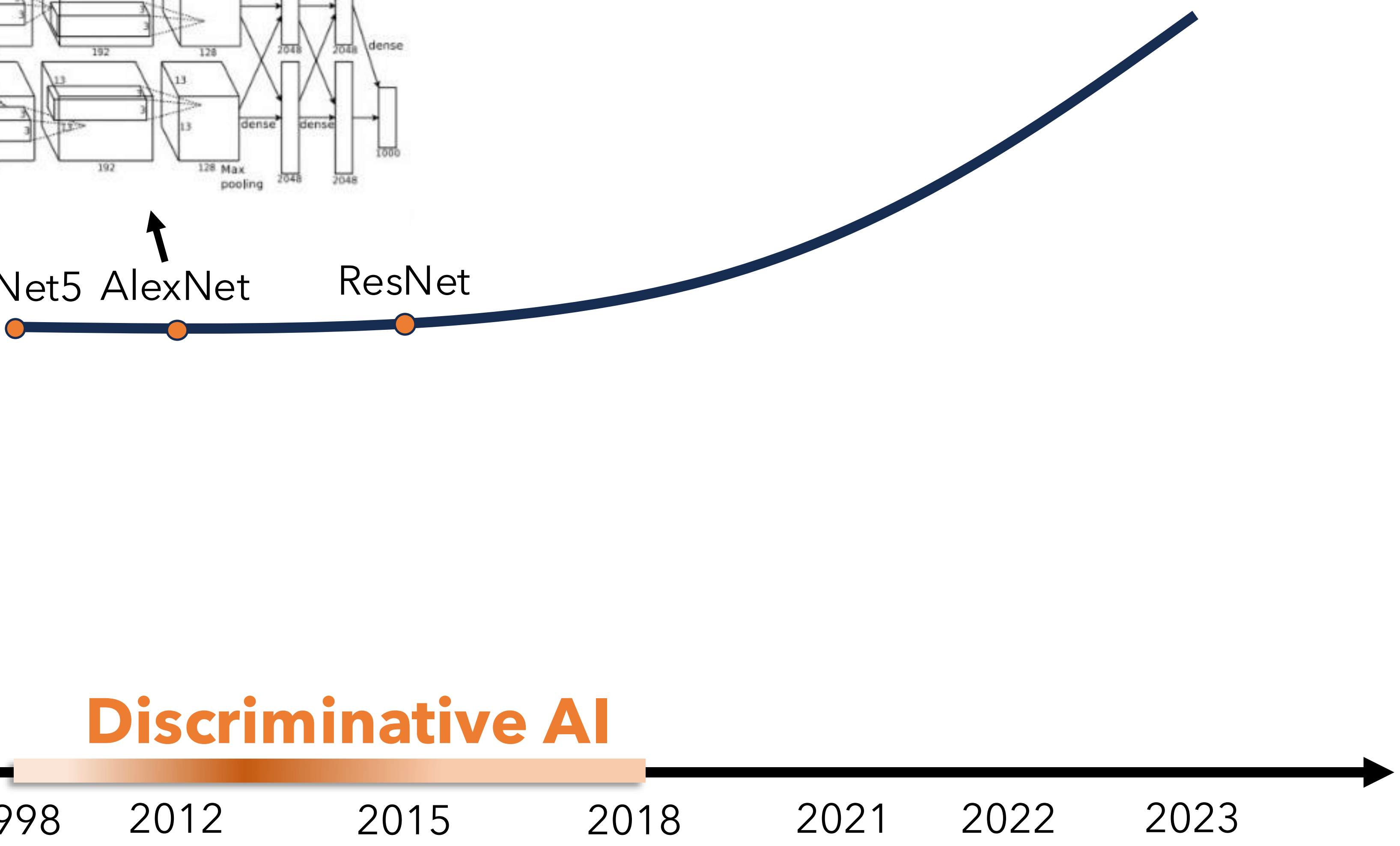
-- Thanks to blogs by Rockwell Anyoha , Toloka Team and Rick Merritt

Millennium Growth of AI

(Krizhevsky et al., 2012)



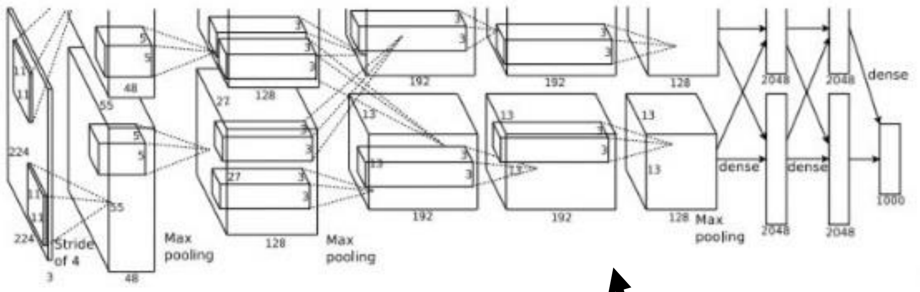
Model

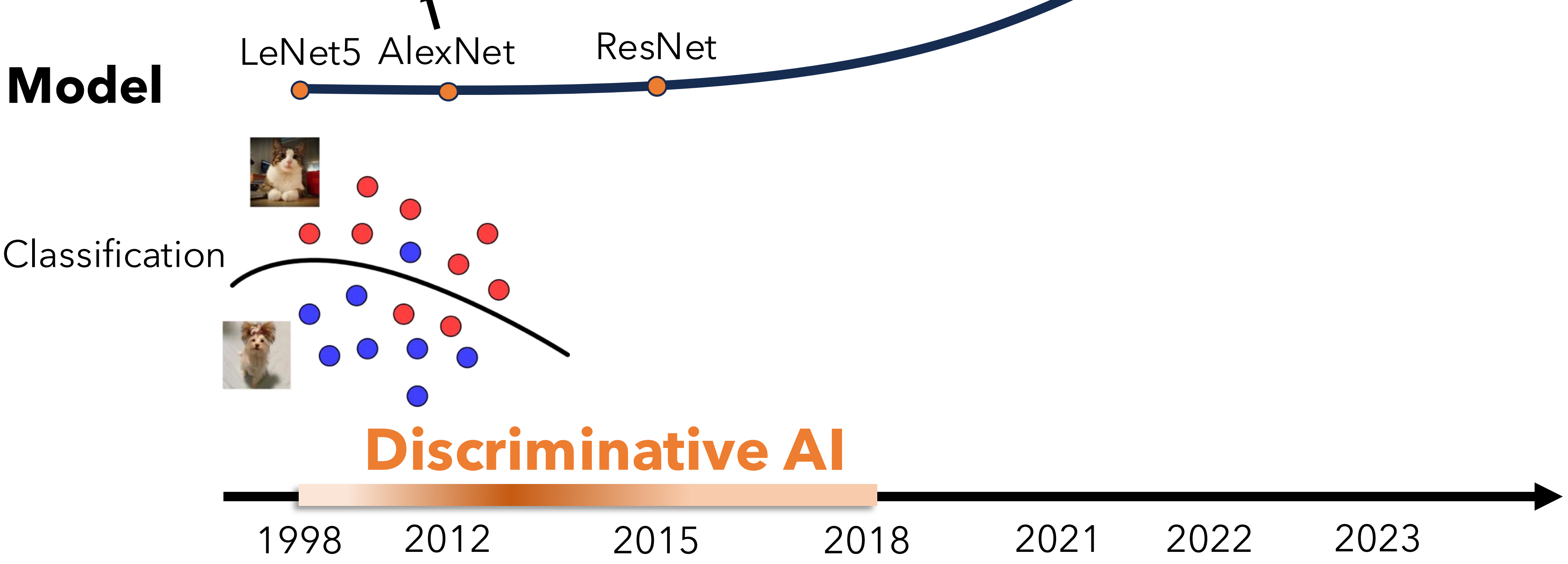


-- Thanks to blogs by Rockwell Anyoha , Toloka Team and Rick Merritt

Millennium Growth of AI

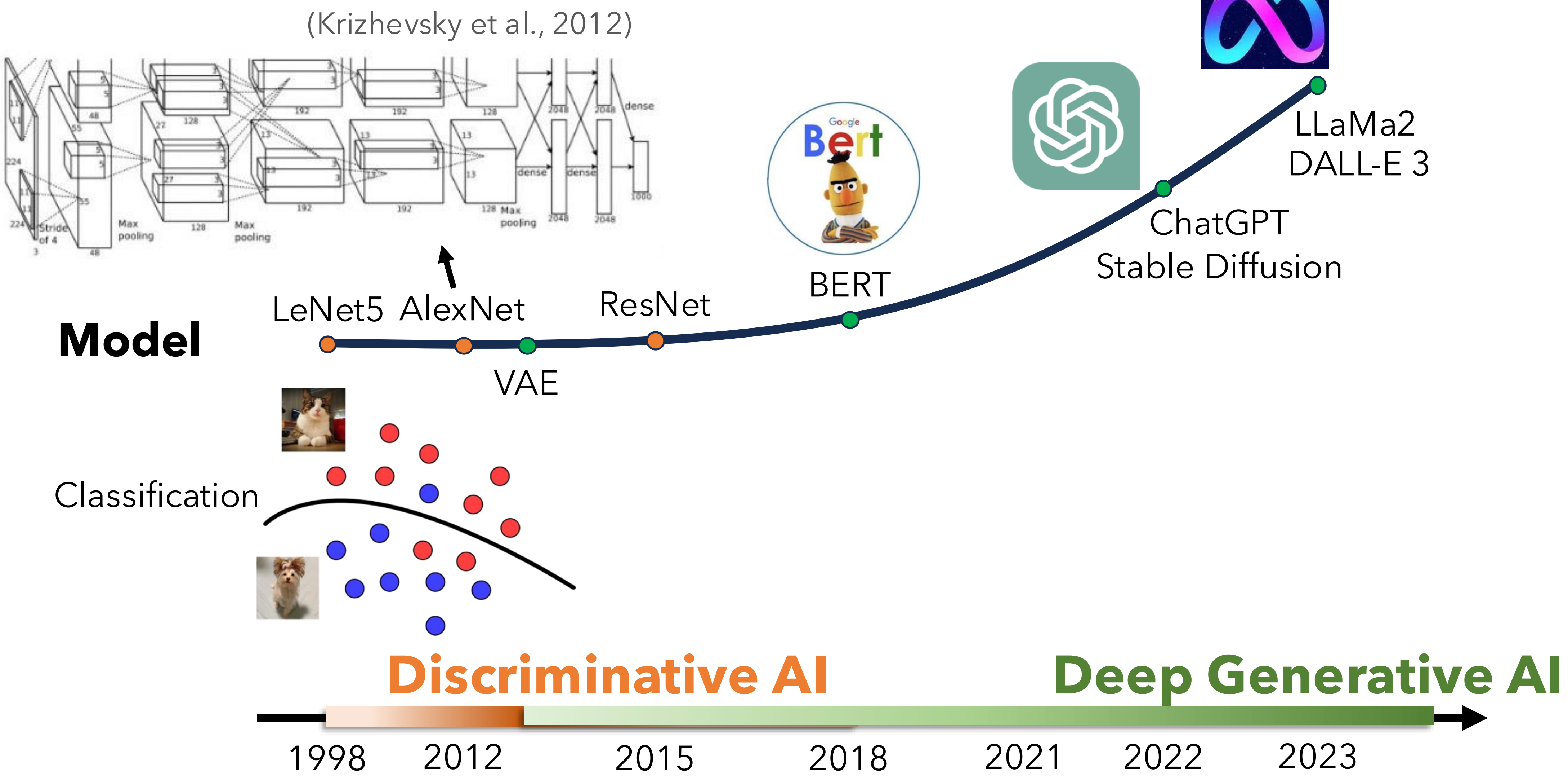
(Krizhevsky et al., 2012)





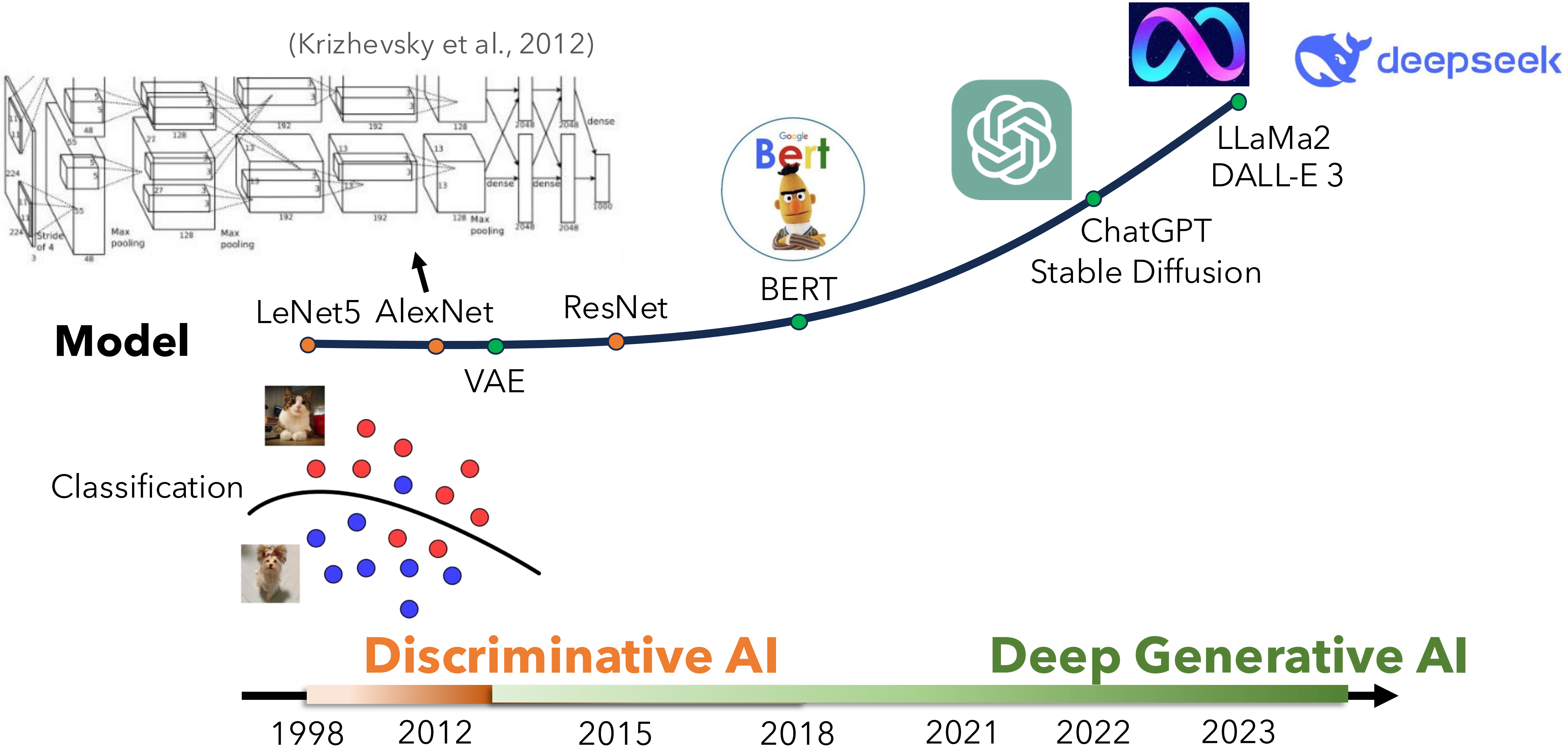
-- Thanks to blogs by Rockwell Anyoha , Toloka Team and Rick Merritt

Millennium Growth of AI



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

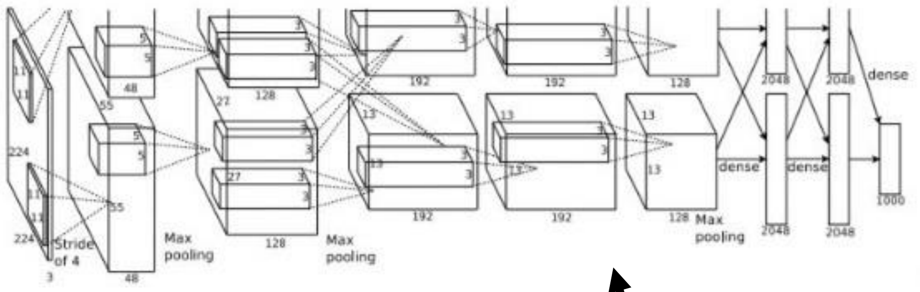
Millennium Growth of AI



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

Millennium Growth of AI

(Krizhevsky et al., 2012)



LLaMa2
DALL-E 3

ChatGPT
Stable Diffusion

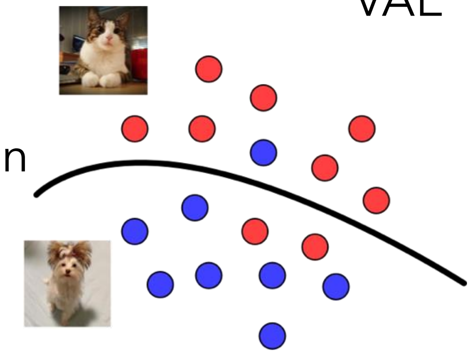
BERT

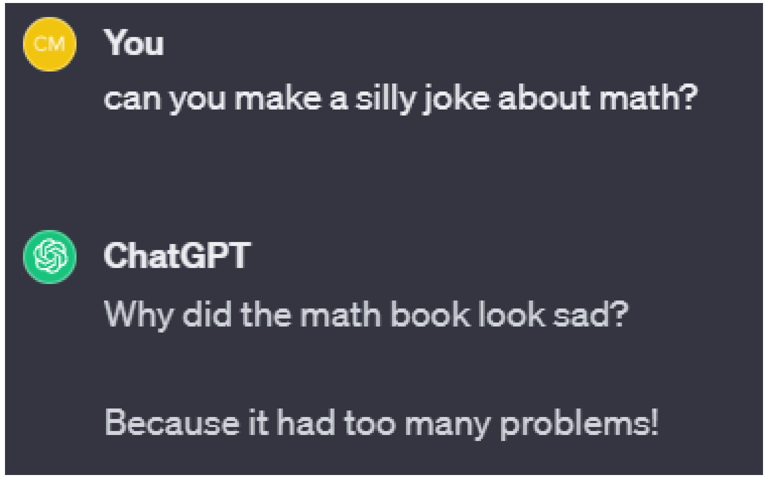
LeNet5 AlexNet ResNet

Model

VAE

Classification





Discriminative AI **Deep Generative AI**

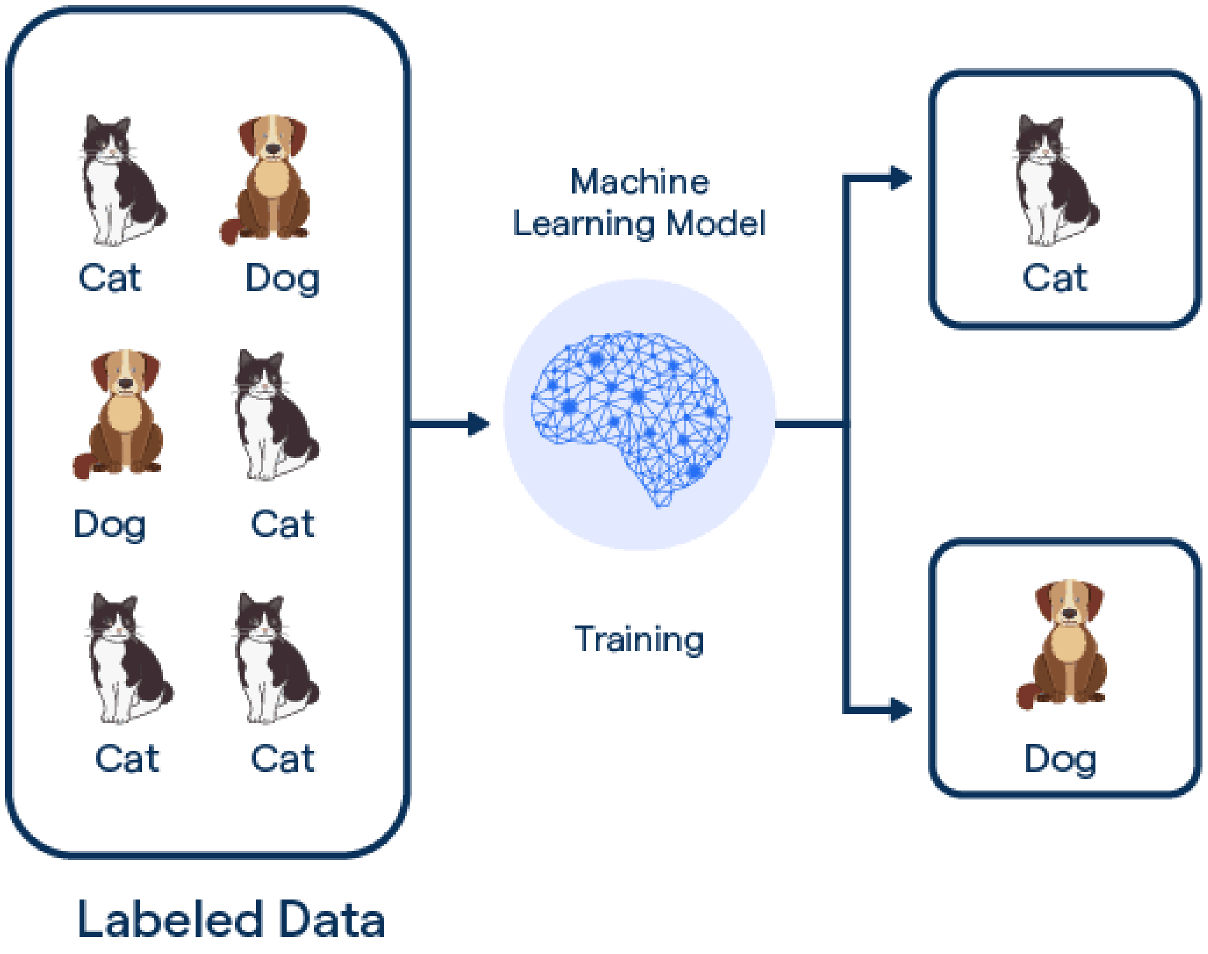
1998 2012 2015 2018 2021 2022 2023

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

A New Training Paradigm

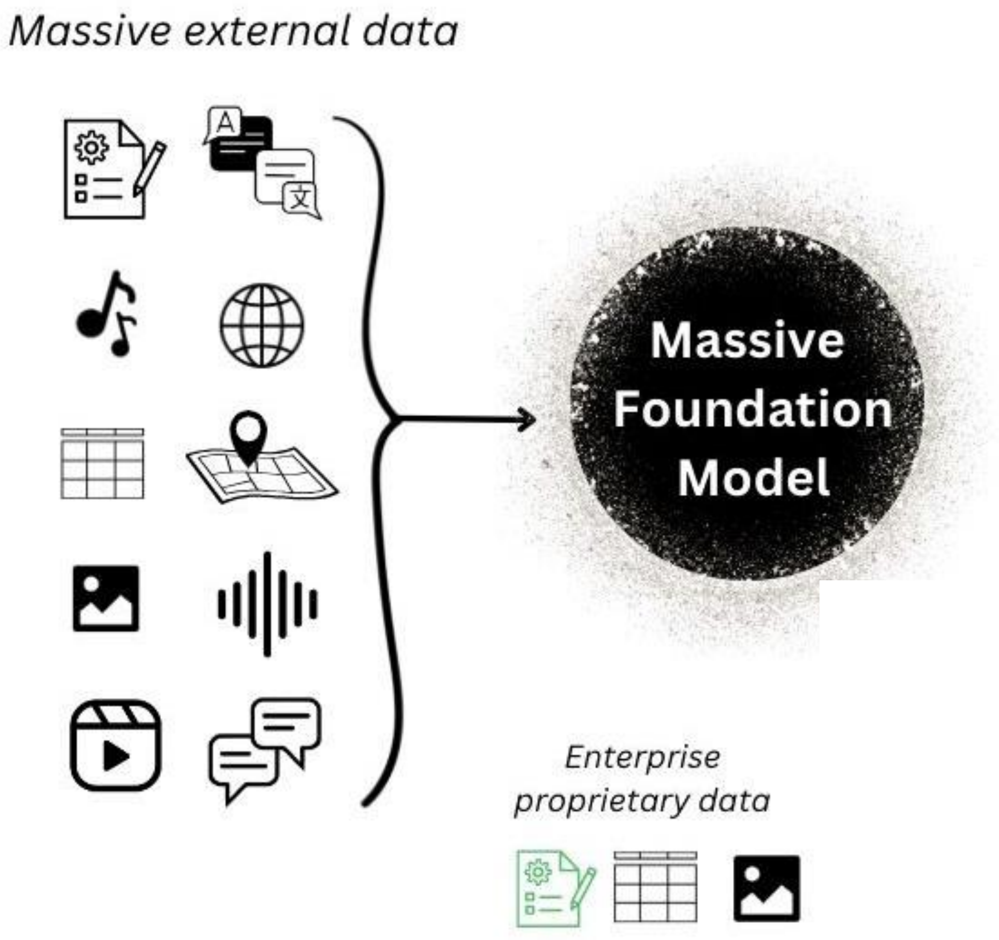
-- Thanks to blogs from BotPenguin and Humanloop

A New Training Paradigm



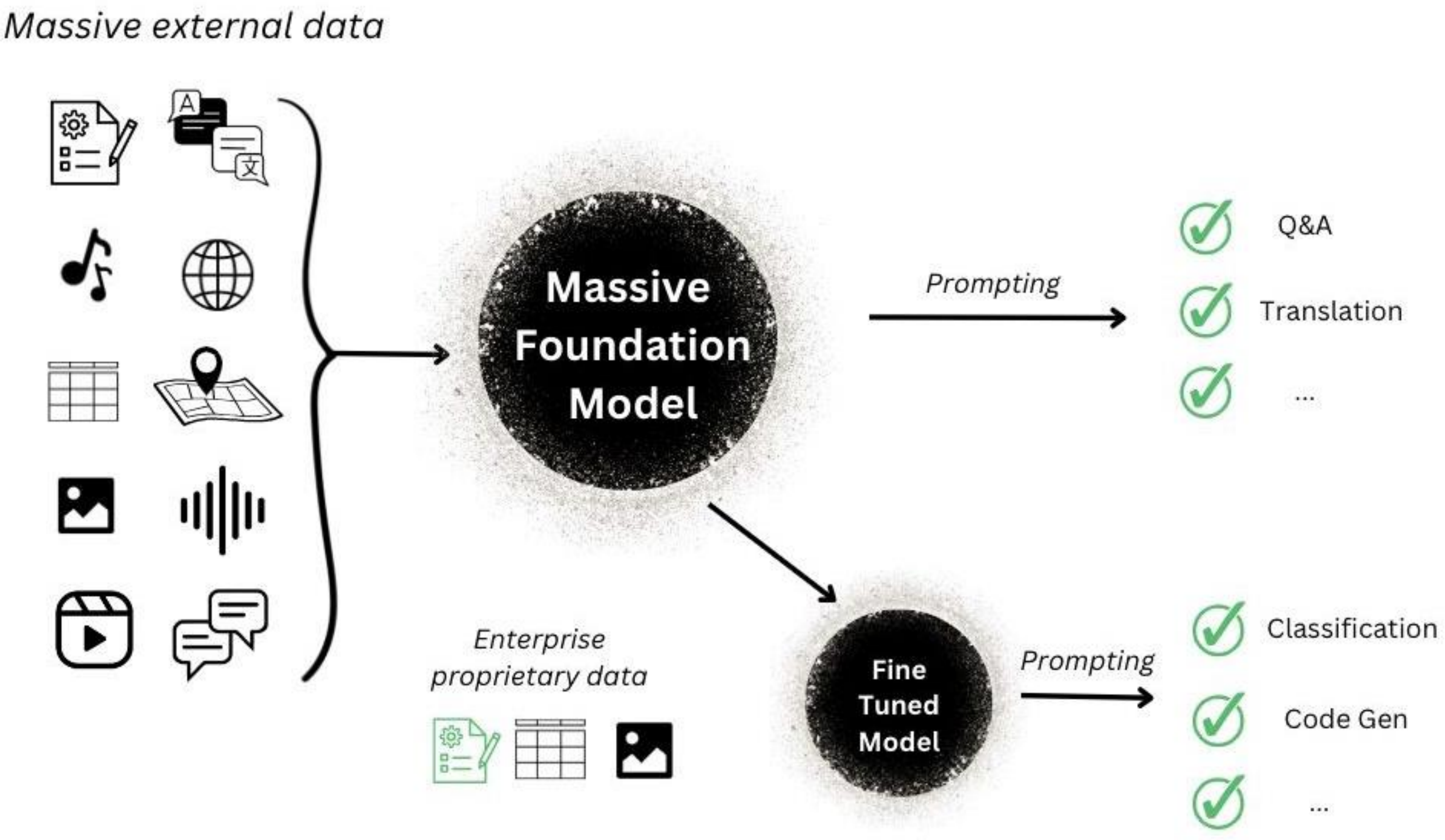
-- Thanks to blogs from BotPenguin and Humanloop

A New Training Paradigm



-- Thanks to blogs from BotPenguin and Humanloop

A New Training Paradigm



-- Thanks to blogs from BotPenguin and Humanloop

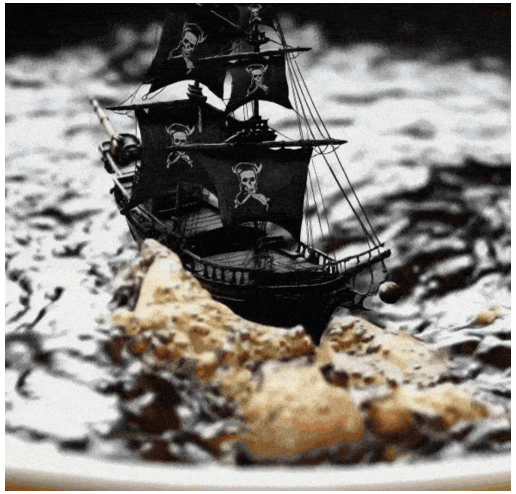
Transformative Power of Fine-Tuning

Transformative Power of Fine-Tuning

ChatGPT

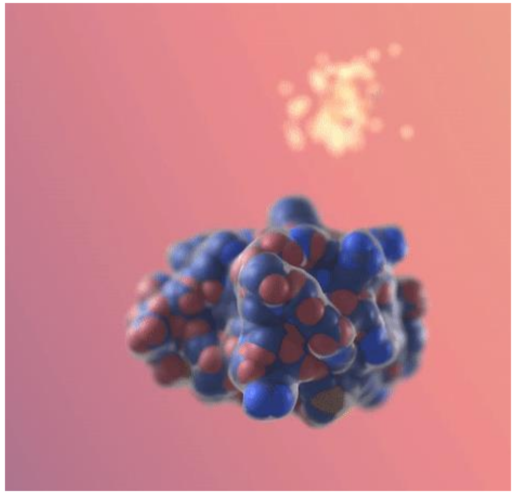
Language

Sora



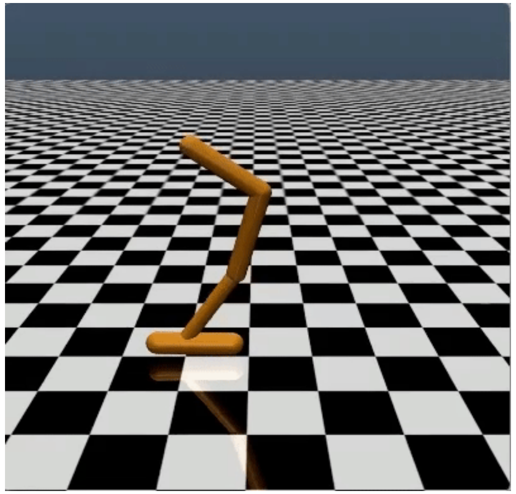
Video

RFDiffusion



Biology

Decision Diffuser



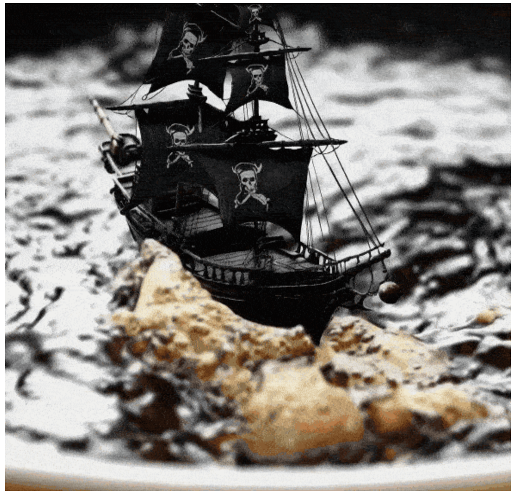
RL/control

Transformative Power of Fine-Tuning

ChatGPT

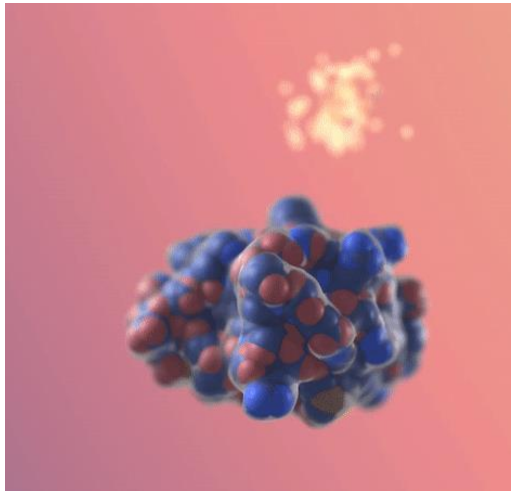
Language

Sora



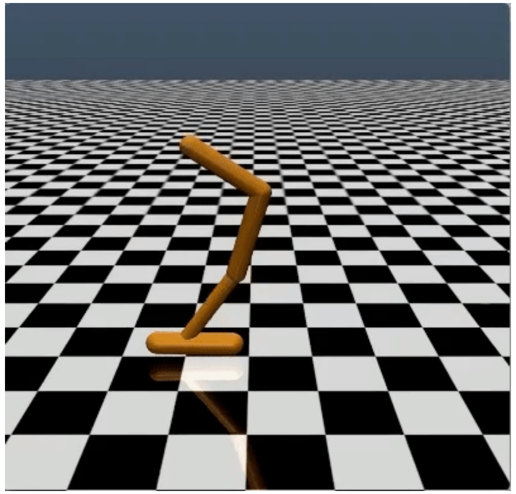
Video

RFDiffusion



Biology

Decision Diffuser



RL/control

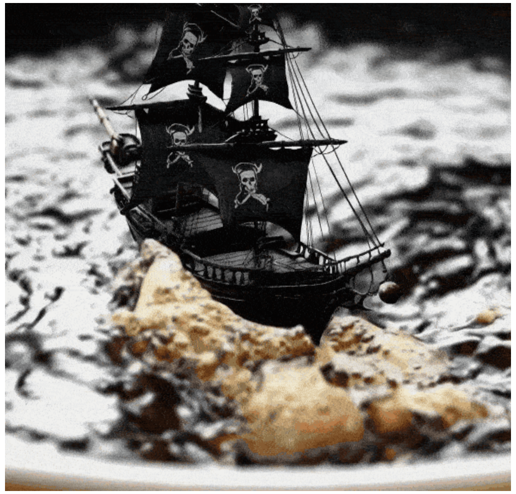
- Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2022), Supervised fine-tuning (Ouyang et al., 2022), RLHF (Ouyang et al., 2022), Distillation (Poole et al., 2022), …

Transformative Power of Fine-Tuning

ChatGPT

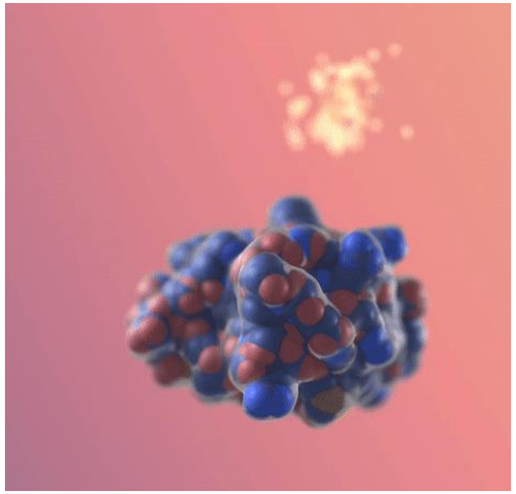
Language

Sora



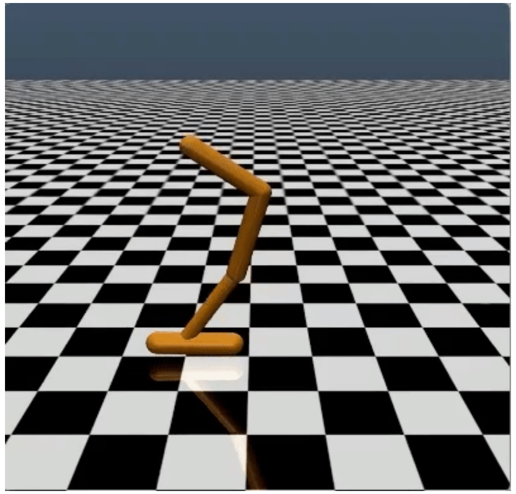
Video

RFDiffusion



Biology

Decision Diffuser

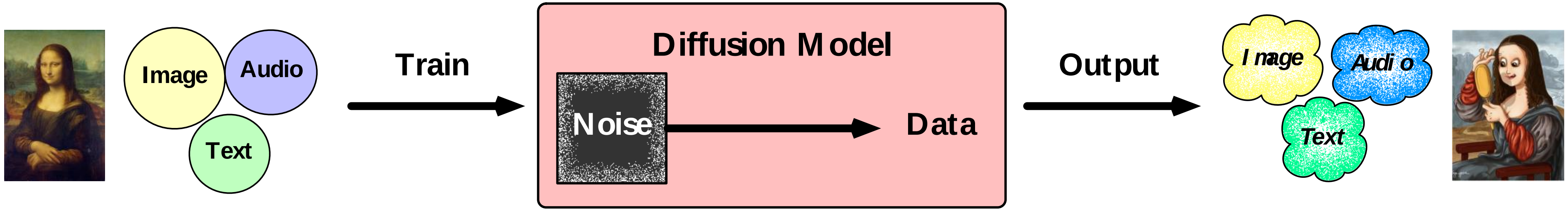


RL/control

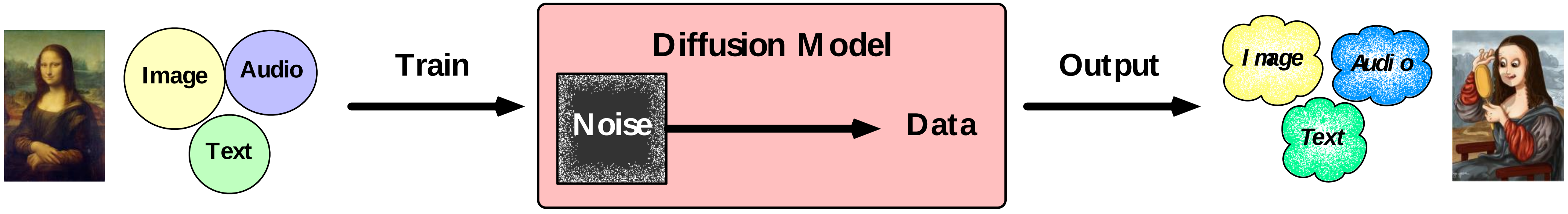
- Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2022), Supervised fine-tuning (Ouyang et al., 2022), RLHF (Ouyang et al., 2022), Distillation (Poole et al., 2022), …

❖ Gap: Methodology focused; limited theoretical guarantees

Our Focus: Diffusion Models



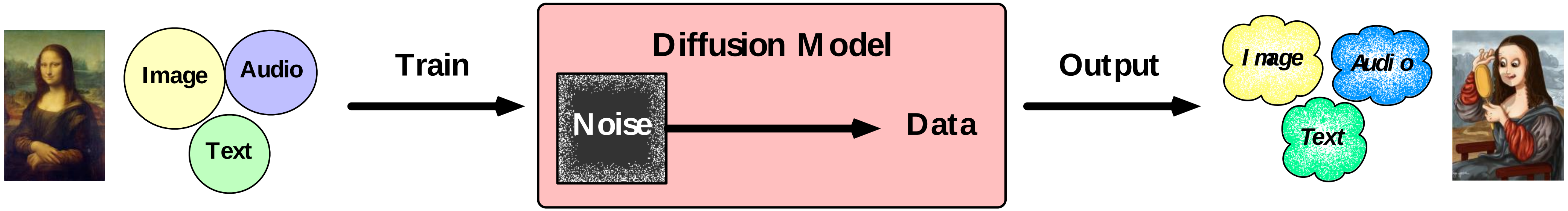
Our Focus: Diffusion Models



Modeling

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

Our Focus: Diffusion Models



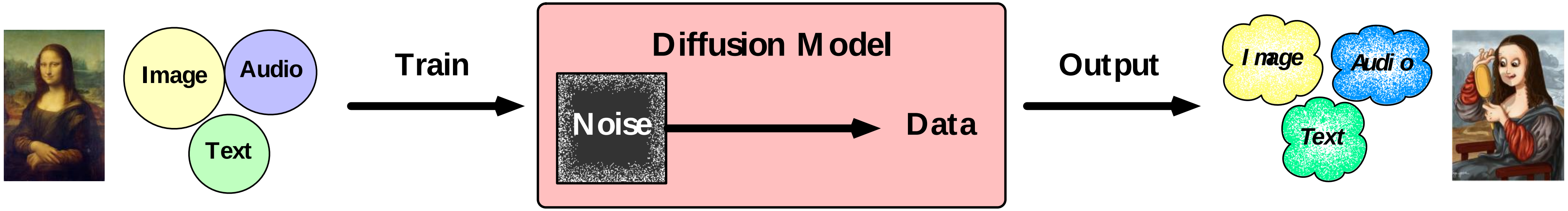
Modeling

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

Sampling

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

Our Focus: Diffusion Models



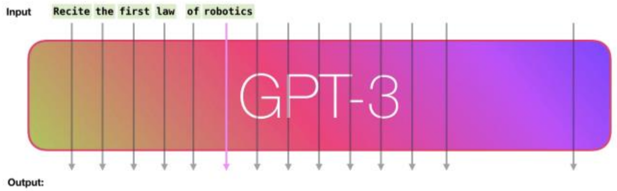
Modeling

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

Sampling

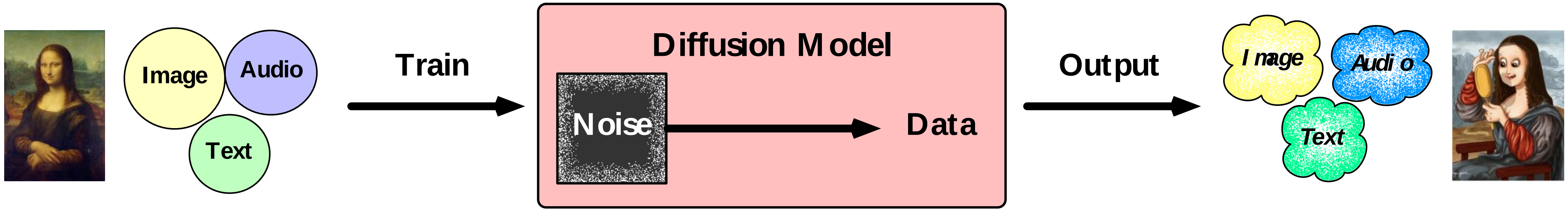
$$x \sim P(\cdot), P(\cdot|y)$$

Autoregressive



Autoregressive

Our Focus: Diffusion Models



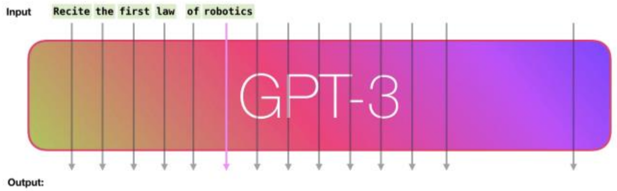
Modeling

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

Sampling

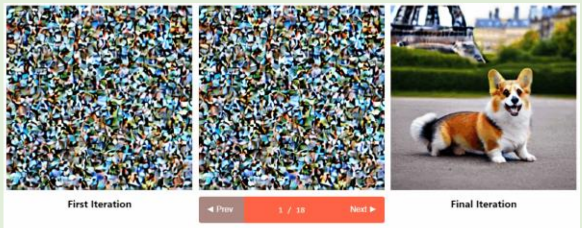
$$x \sim P(\cdot), P(\cdot|y)$$

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*
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

New Promises of Diffusion Models

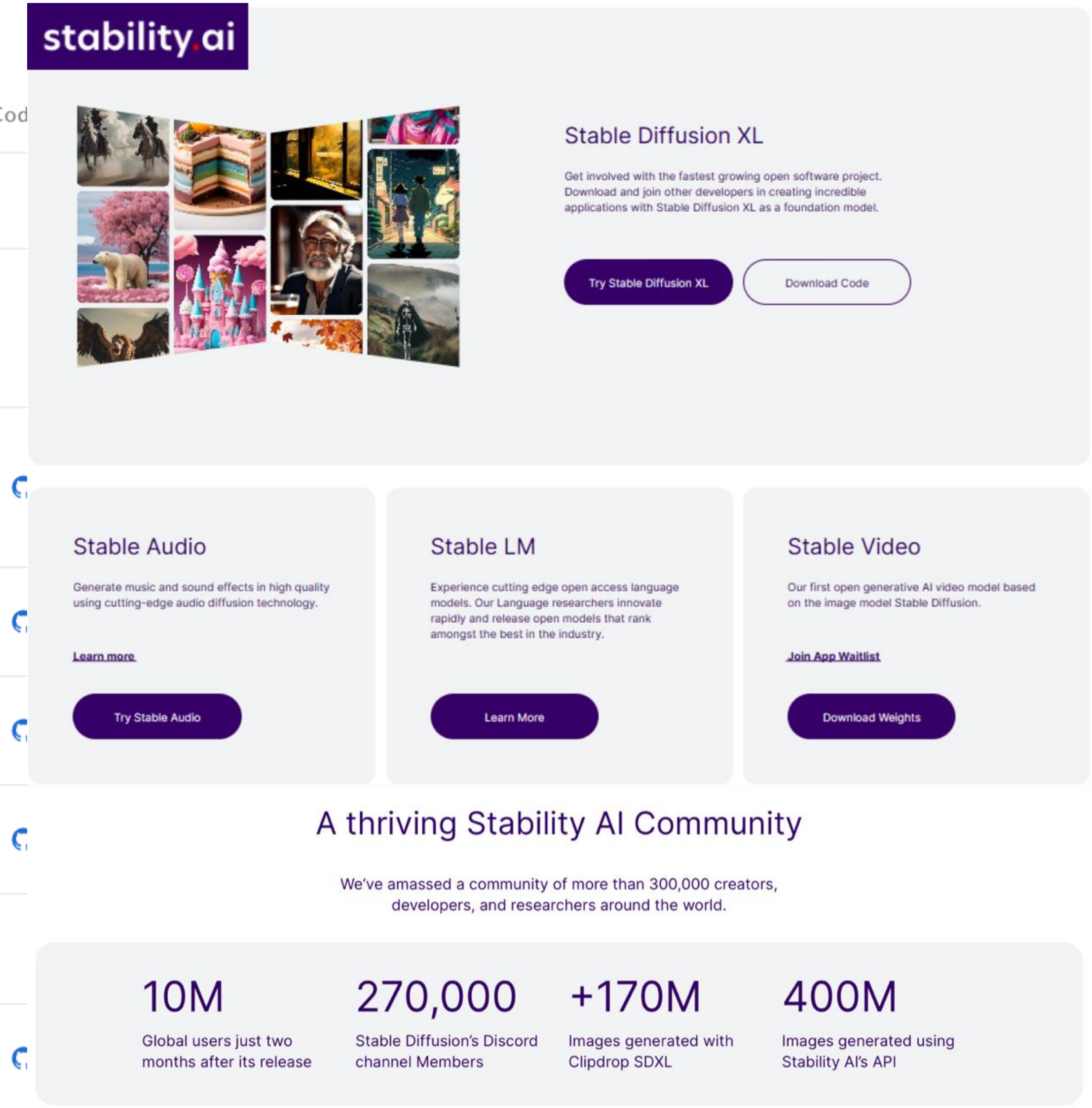
ImageNet Benchmark (Image Generation)

Rank	Model	FID ↓	Paper	Code	Result	Year	Tags
1	DiT-XL/2 with CADS	1.70	CADS: Unleashing the Diversity of Diffusion Models through Condition-Annealed Sampling			2023	Diffusion
2	MAGVIT-v2	1.78	Language Model Beats Diffusion -- Tokenizer is Key to Visual Generation			2023	VAE/VQ-VAE, Mask Prediction
3	MDT-XL2	1.79	Masked Diffusion Transformer is a Strong Image Synthesizer			2023	VAE/VQ-VAE, Diffusion
4	Discriminator Guidance	1.83	Refining Generative Process with Discriminator Guidance in Score-based Diffusion Models			2022	Diffusion
5	RDM	1.87	Relay Diffusion: Unifying diffusion process across resolutions for image synthesis			2023	Diffusion
6	ViT-XL	2.06	Efficient Diffusion Training via Min-SNR Weighting Strategy			2023	Diffusion
7	VDM++	2.12					Diffusion
8	StyleSAN-XL	2.14	SAN: Inducing Metrizability of GAN with Discriminative Normalized Linear Layer			2023	GAN

New Promises of Diffusion Models

ImageNet Benchmark (Image Generation)

Rank	Model	FID ↓	Paper	Cod
1	**DiT-XL/2 with CADS**	1.70	CADS: Unleashing the Diversity of Diffusion Models through Condition-Annealed Sampling	
2	**MAGVIT-v2**	1.78	Language Model Beats Diffusion -- Tokenizer is Key to Visual Generation	
3	**MDT-XL2**	1.79	Masked Diffusion Transformer is a Strong Image Synthesizer	
4	**Discriminator Guidance**	1.83	Refining Generative Process with Discriminator Guidance in Score-based Diffusion Models	
5	**RDM**	1.87	Relay Diffusion: Unifying diffusion process across resolutions for image synthesis	
6	**ViT-XL**	2.06	Efficient Diffusion Training via Min-SNR Weighting Strategy	
7	**VDM++**	2.12		
8	**StyleSAN-XL**	2.14	SAN: Inducing Metrizability of GAN with Discriminative Normalized Linear Layer	

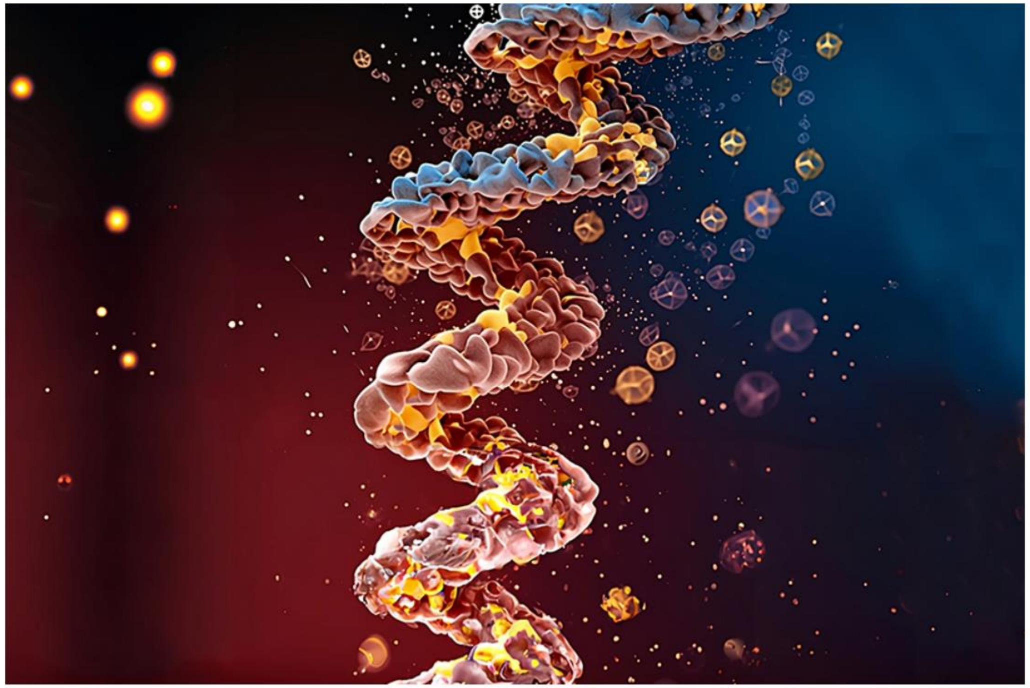


New Promises of Diffusion Models

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.

n)
Cod
Models
r is Key to
age
tor
cross
ghting

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.

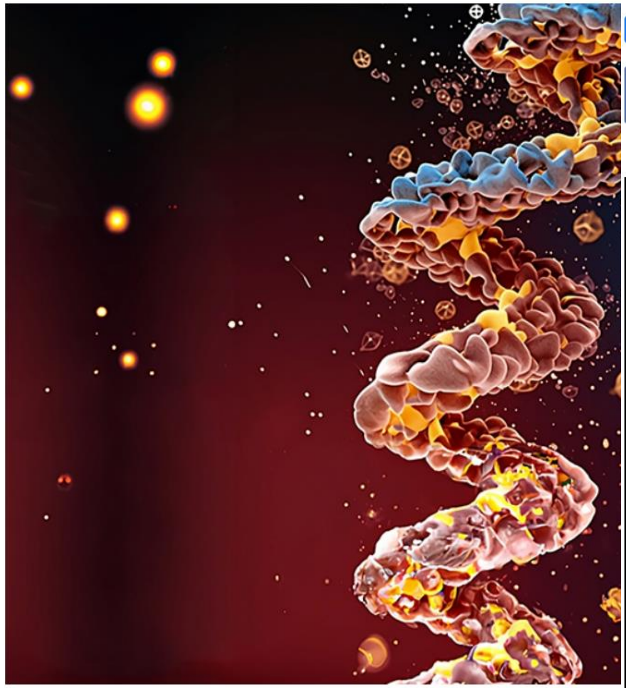
10M	270,000	+170M	400M
Global users just two months after its release	Stable Diffusion's Discord channel Members	Images generated with Clipdrop SDXL	Images generated using Stability AI's API

New Promises of Diffusion Models

Generative AI imagines new pr

“FrameDiff” is a computational tool that uses g
protein structures, with the aim of accelerating
improving gene therapy.

Rachel Gordon | MIT CSAIL
July 12, 2023



Biology is a wondrous yet delicate tapestry. At the heart is
encodes proteins, responsible for orchestrating the many
within the human body. However, our body is akin to a fine
losing its harmony. After all, we’re faced with an ever-chan
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
Our 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.

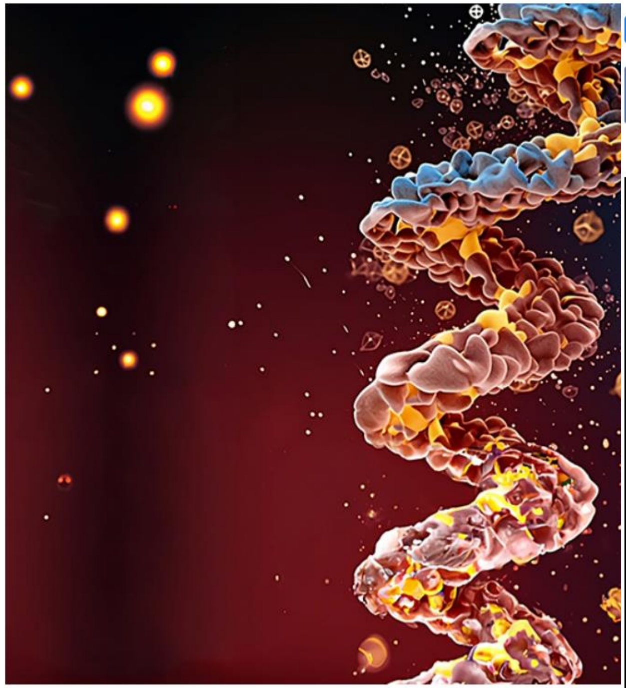
)0	+170M	400M
Discord	Images generated with Clipdrop SDXL	Images generated using Stability AI's API

New Promises of Diffusion Models

Generative AI imagines new pr

“FrameDiff” is a computational tool that uses g
protein structures, with the aim of accelerating
improving gene therapy.

Rachel Gordon | MIT CSAIL
July 12, 2023



Biology is a wondrous yet delicate tapestry. At the heart is
encodes proteins, responsible for orchestrating the many
within the human body. However, our body is akin to a fine
losing its harmony. After all, we’re faced with an ever-chan
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

Gen

"Fram
protei
impro

Rachel C
July 12,

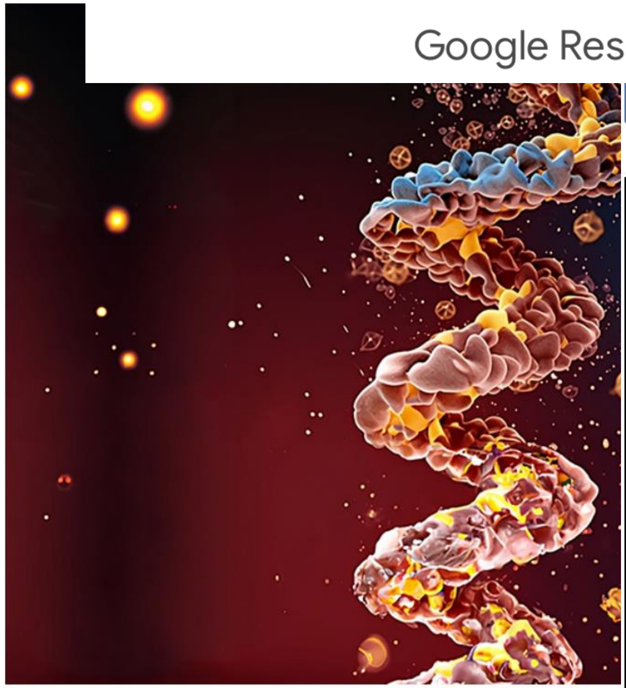
Diffusion Models Are Real-Time Game Engines

s of

Dani Valevski* Yaniv Leviathan* Moab Arar*† Shlomi Fruchter*

Google Research Google Research Tel Aviv University Google DeepMind

Facebook Twitter RedditIt



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

Image generated with Bing Image Creator

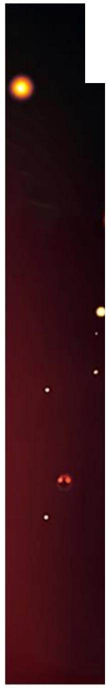
This article is part of our coverage of the late

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

Gen
“Fram
protei
impro

Rachel C
July 12,



Diffusion Models Are Real-Time Game Engines

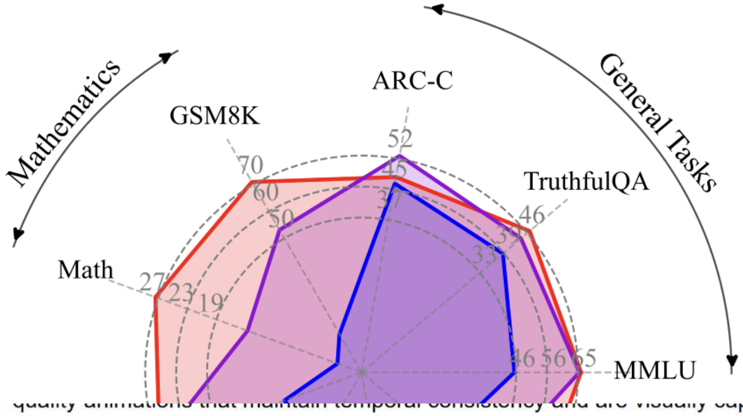
s of

Large Language Diffusion Models

Shen Nie [1 * †] **Fengqi Zhu** [1 * †] **Zebin You** [1 †] **Xiaolu Zhang** [2 ‡] **Jingyang Ou** [1] **Jun Hu** [2 ‡] **Jun Zhou** [2]
Yankai Lin [1 ‡] **Ji-Rong Wen** [1] **Chongxuan Li** [1 ‡ ¶]

Abstract

Autoregressive models (ARMs) are widely regarded as the cornerstone of large language models (LLMs). We challenge this notion by introducing **LLaDA**, a diffusion model trained from scratch under the pre-training and supervised fine-tuning (SFT) paradigm. LLaDA models distributions through a forward data masking process



Biology is
encodes
within the human body. However, our body is akin to a fine
losing its harmony. After all, we’re faced with an ever-chan
pathogens, viruses, diseases, and cancer.

This article is part of our coverage of the late

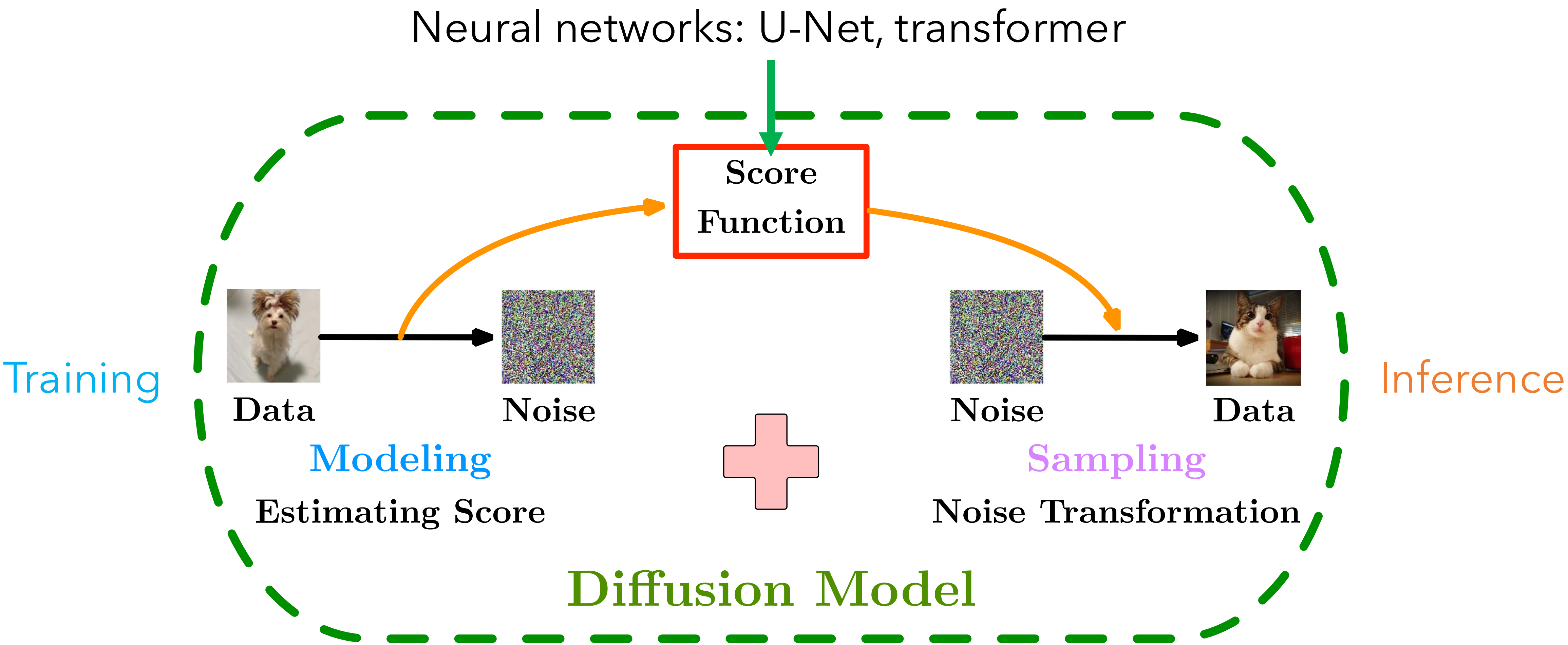
and audio
Despite
ting high-
quality animations that maintain temporal consistency and are visually captivating. A major cause for the
complexity is the need for intricate coordination of lip movements, head positions, and facial expressions
to craft a visually compelling effect.

Outline

- Introduction to diffusion models
- Conditional models and guidance
- Guiding diffusion models in offline and online settings
- Future directions

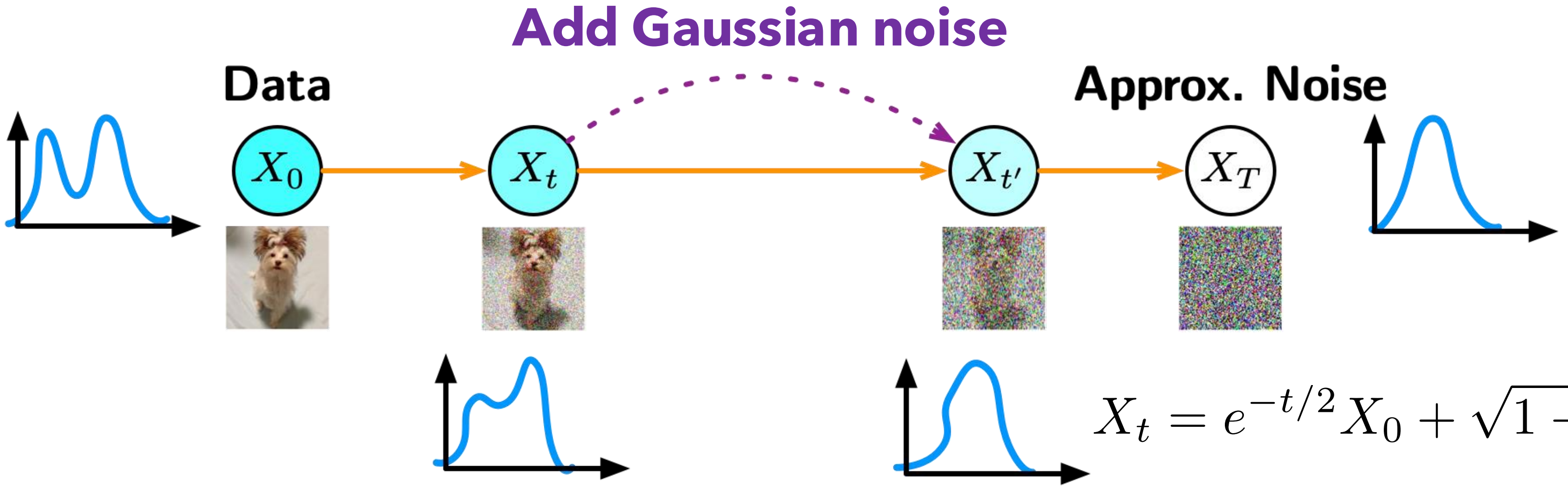
Diffusion Model and Guidance

Overview of Diffusion Models



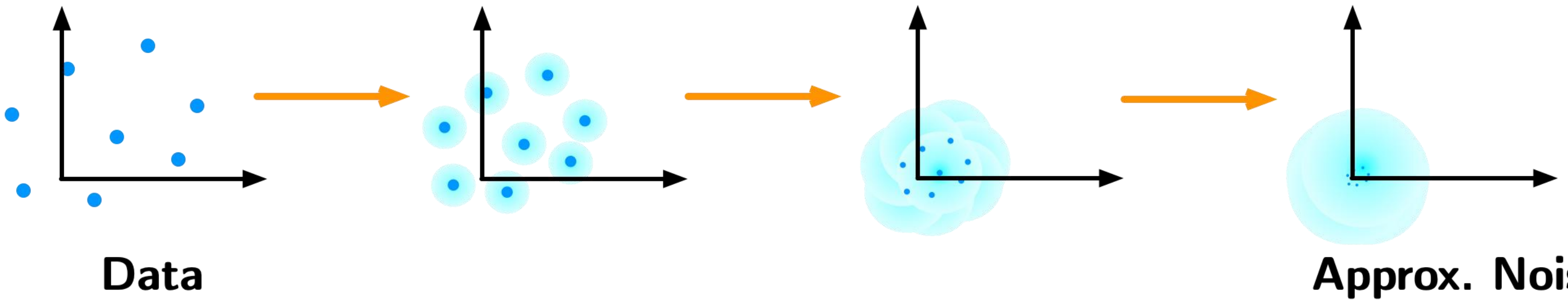
Forward Process – Noise Corruption

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



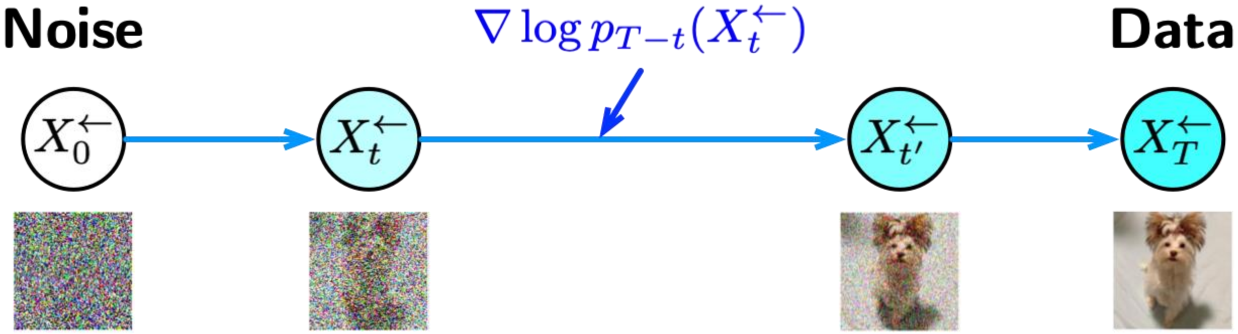
$$X_t = e^{-t/2}X_0 + \sqrt{1 - e^{-t}} \cdot \mathsf{N}(0, I)$$

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

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.*

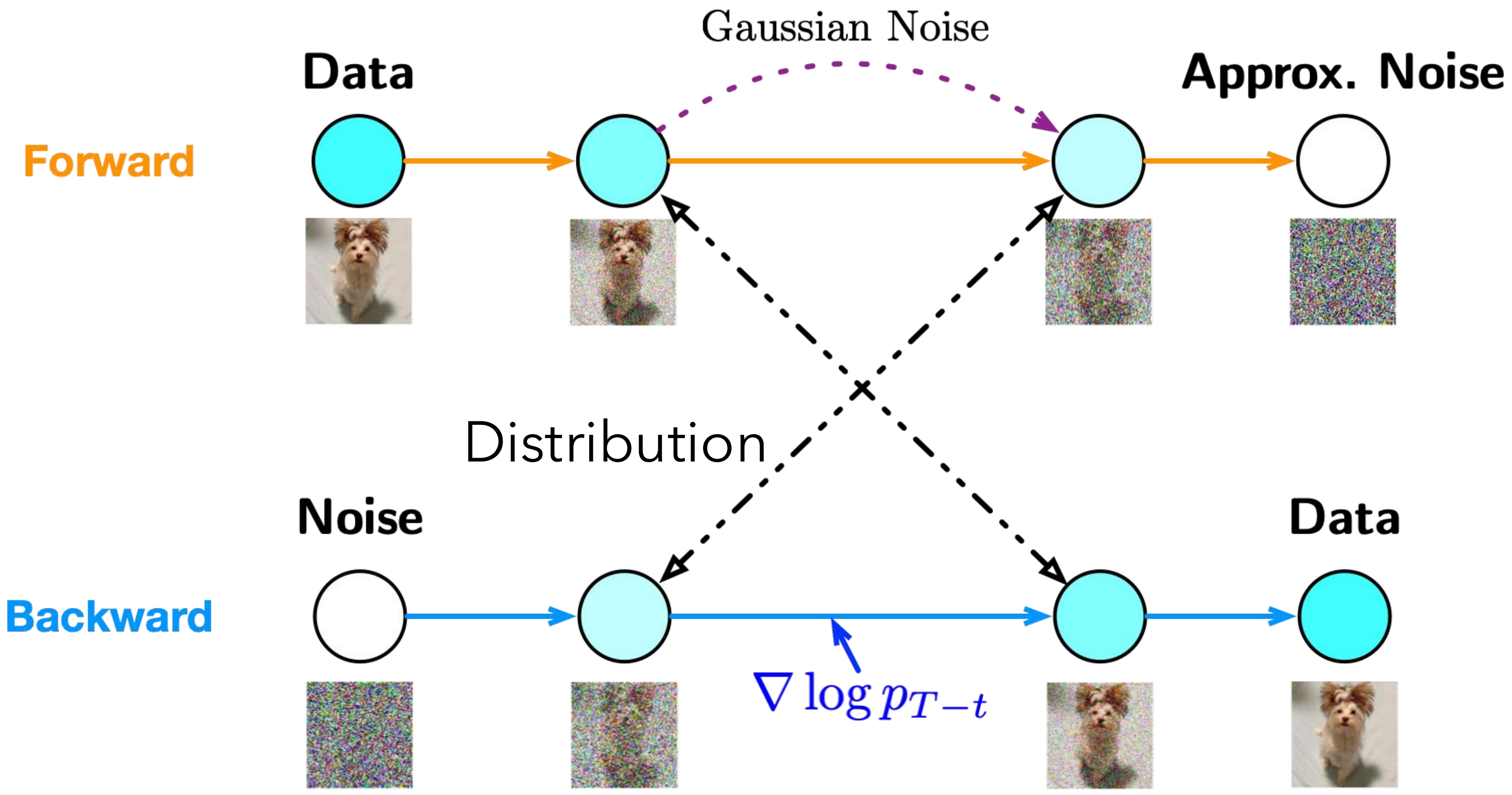
(iii) *A reverse time model for x_t is defined by*

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

where

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

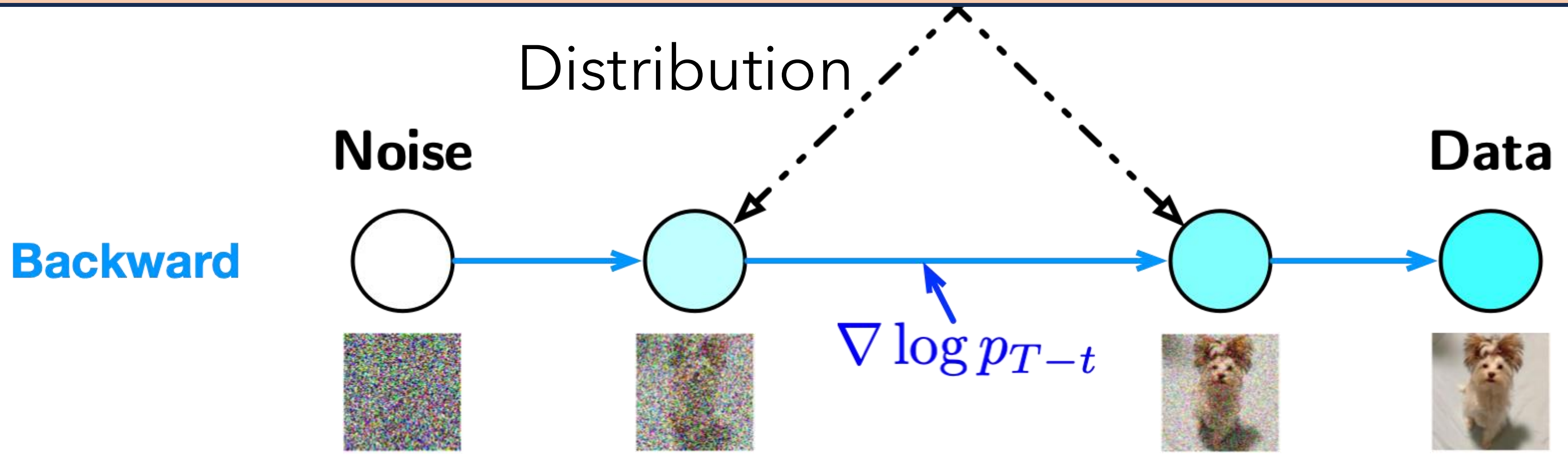
Forward and Backward Coupling



Forward and Backward Coupling

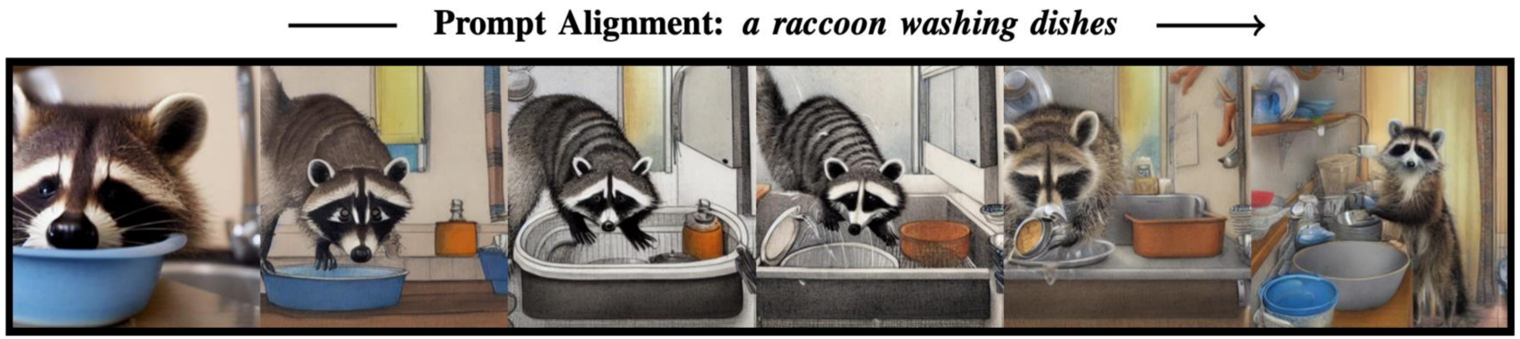
Training $\int_0^T \mathbb{E}_{x_t}\left[\|\nabla \log p_t(x_t) - s(x_t, t)\|_2^2\right] \mathrm{d}t$

where s is parameterized by neural networks

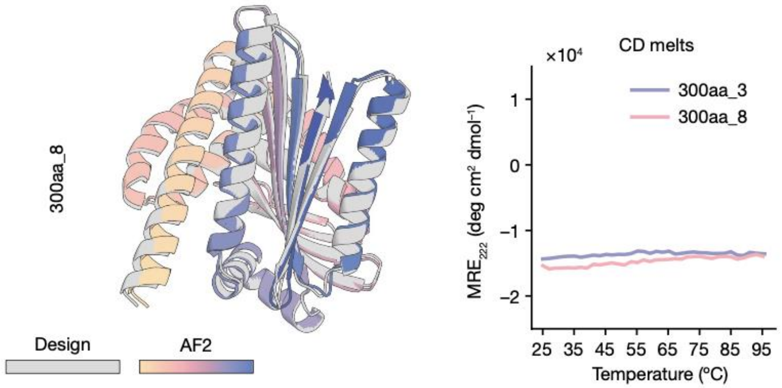


From $P(x)$ to $P(x|y)$

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



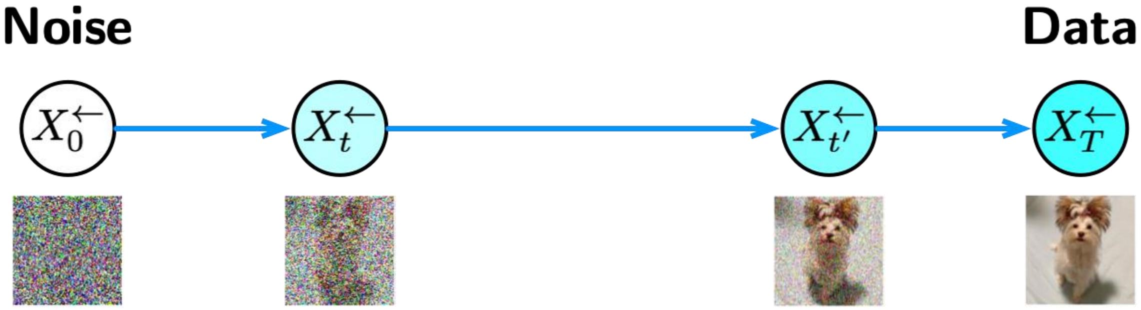
- Protein generation with biochemical properties (Watson et al., 2023; Gruver et al., 2023)



Conditional Diffusion Models

- Conditioned sample generation for a given label

$y = \mathrm{dog}$



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

Conditional Score **Brownian**

-- More in the survey: M. Chen, S. Mei, J. Fan, and M. Wang. "Challenges and Opportunities of Diffusion Models for Generative AI". National Science Review 2024

Conditional Diffusion Models

- Conditioned sample generation for a given label

$y = \mathrm{dog}$

Label

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

-- More in the survey: M. Chen, S. Mei, J. Fan, and M. Wang. "Challenges and Opportunities of Diffusion Models for Generative AI". National Science Review 2024

Conditional Diffusion Models

- Conditioned sample generation for a given label

$y = \text{dog}$

Label

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

Conditional Score **Brownian**

- Diffusion models can handle diverse conditional information, e.g., text prompts, partial images, etc.
- The key is to scalably estimate the conditional score

-- More in the survey: M. Chen, S. Mei, J. Fan, and M. Wang. "Challenges and Opportunities of Diffusion Models for Generative AI". National Science Review 2024

Conditional Score and Guidance

Conditional Score Revisited

- Bayes' rule defines **guidance**

$$\begin{aligned}\nabla_x \log p_t(x_t|y) &= \nabla_x \log \frac{p_t(x_t, y)}{p_t(y)} \\ &= \nabla_x \log p_t(x_t) p_t(y|x_t) - \cancel{\nabla_x \log p_t(y)} \\ &= \nabla_x \log p_t(x_t) + \color{blue}{\nabla_x \log p_t(y|x_t)}\end{aligned}$$

Conditional Score Revisited

- Bayes' rule defines **guidance**

$$
\begin{aligned}
\nabla_x \log p_t(x_t|y) &= \nabla_x \log \frac{p_t(x_t, y)}{p_t(y)} \\
&= \nabla_x \log p_t(x_t) p_t(y|x_t) - \cancel{\nabla_x \log p_t(y)} \\
&= \nabla_x \log p_t(x_t) + \boxed{\nabla_x \log p_t(y|x_t)}
\end{aligned}
$$

Conditional Score Revisited

- Bayes' rule defines **guidance**

$$
\begin{aligned}
\nabla_x \log p_t(x_t|y) &= \nabla_x \log \frac{p_t(x_t, y)}{p_t(y)} \\
&= \nabla_x \log p_t(x_t) p_t(y|x_t) - \cancel{\nabla_x \log p_t(y)} \\
&= \nabla_x \log p_t(x_t) + \boxed{\nabla_x \log p_t(y|x_t)}
\end{aligned}
$$

Guidance

Conditional Score Revisited

- Bayes' rule defines **guidance**

$$
\begin{aligned}
\nabla_x \log p_t(x_t|y) &= \nabla_x \log \frac{p_t(x_t, y)}{p_t(y)} \\
&= \nabla_x \log p_t(x_t) p_t(y|x_t) - \cancel{\nabla_x \log p_t(y)} \\
&= \nabla_x \log p_t(x_t) + \underbrace{\boxed{\nabla_x \log p_t(y|x_t)}}_{\textbf{Guidance}}
\end{aligned}
$$

- When *y* is a label, guidance is gradient of classification logit; known as the *classifier guidance* (Dhariwal & Nichol, 2021)

Conditional Score Revisited

- Bayes' rule defines **guidance**

$$\nabla_x \log p_t(x_t|y) = \nabla_x \log \frac{p_t(x_t, y)}{p_t(y)}$$
$$= \nabla_x \log p_t(x_t)p_t(y|x_t) - \cancel{\nabla_x \log p_t(y)}$$
$$= \nabla_x \log p_t(x_t) + \underbrace{\boxed{\nabla_x \log p_t(y|x_t)}}_{\text{Guidance}}$$

- When *y* is a label, guidance is gradient of classification logit; known as the *classifier guidance* (Dhariwal & Nichol, 2021)

- Beyond labels? (Ho & Salimans, 2022; Bansal et al., 2023)

Conditional Score Revisited

- Bayes' rule defines **guidance**

$$
\begin{aligned}
\nabla_x \log p_t(x_t|y) &= \nabla_x \log \frac{p_t(x_t, y)}{p_t(y)} \\
&= \nabla_x \log p_t(x_t) p_t(y|x_t) - \cancel{\nabla_x \log p_t(y)} \\
&= \nabla_x \log p_t(x_t) + \underbrace{\boxed{\nabla_x \log p_t(y|x_t)}}_{\textbf{Guidance}}
\end{aligned}
$$

- When *y* is a label, guidance is gradient of classification logit; known as the *classifier guidance* (Dhariwal & Nichol, 2021)

- Beyond labels? 🤔 (Ho & Salimans, 2022; Bansal et al., 2023)

Role of Guidance I --- To Optimize

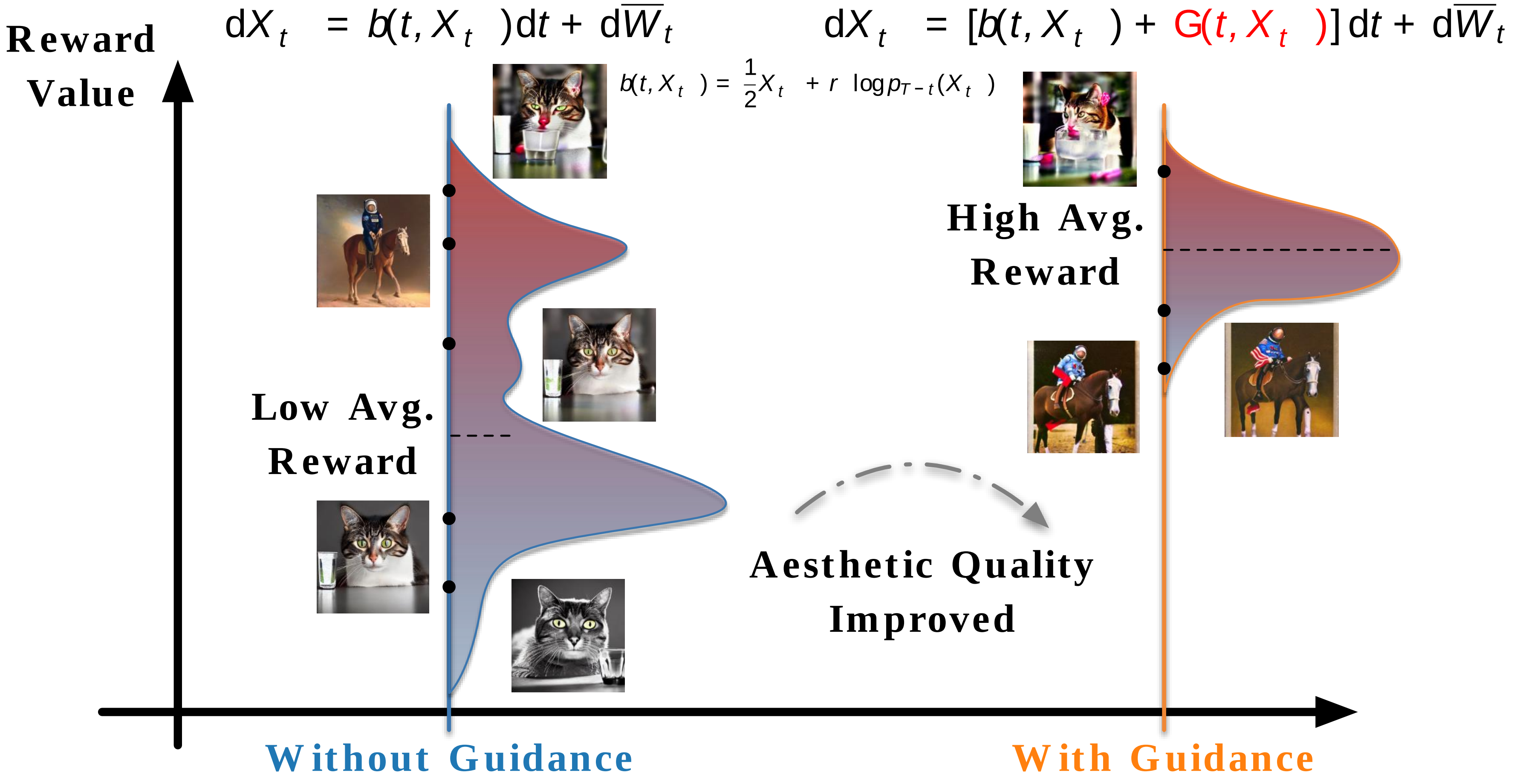
- Abstract the task-specific objective as a reward function f

Role of Guidance I --- To Optimize

- Abstract the task-specific objective as a reward function f
- Guidance is to optimize reward by **generating solutions**

Role of Guidance I --- To Optimize

- Abstract the task-specific objective as a reward function f
- Guidance is to optimize reward by **generating solutions**



Role of Guidance II --- Preserve Fidelity

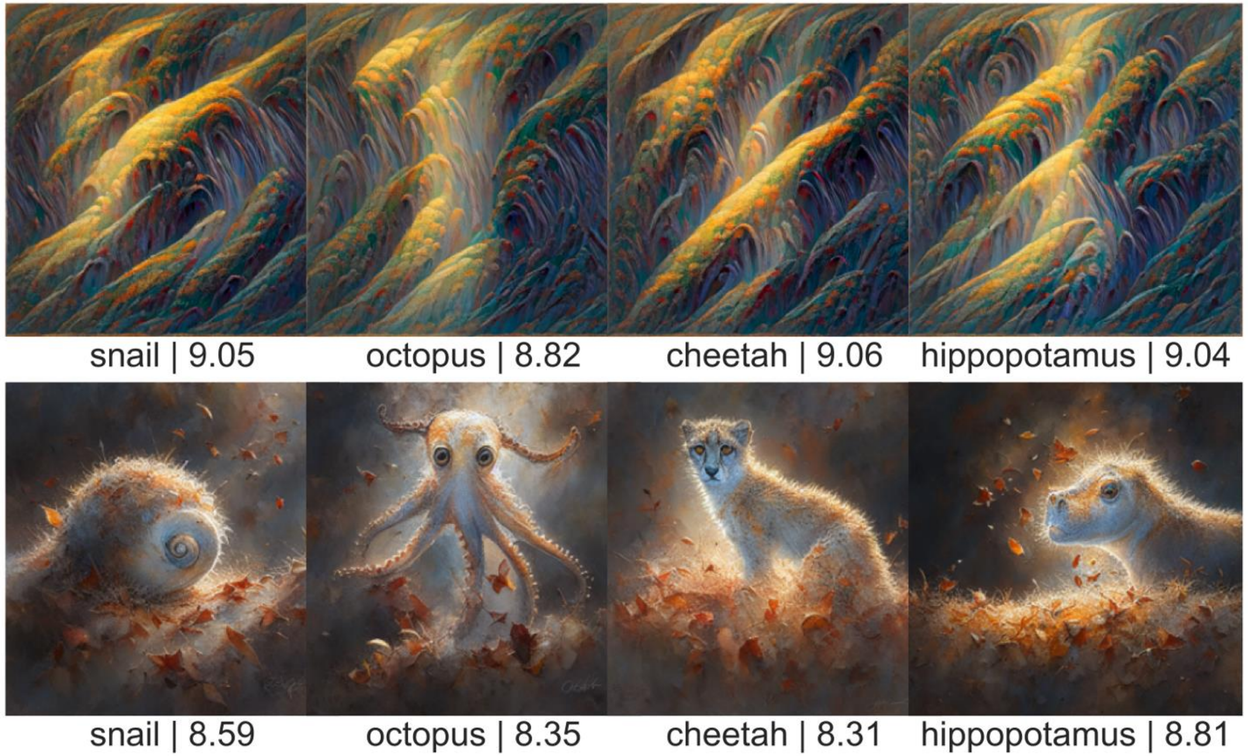
- Reward collapse is a common challenge (Ouyang et al., 2022; Song et al., 2023)

Role of Guidance II --- Preserve Fidelity

- Reward collapse is a common challenge (Ouyang et al., 2022; Song et al., 2023)
- Guidance also needs to preserve data fidelity

Role of Guidance II --- Preserve Fidelity

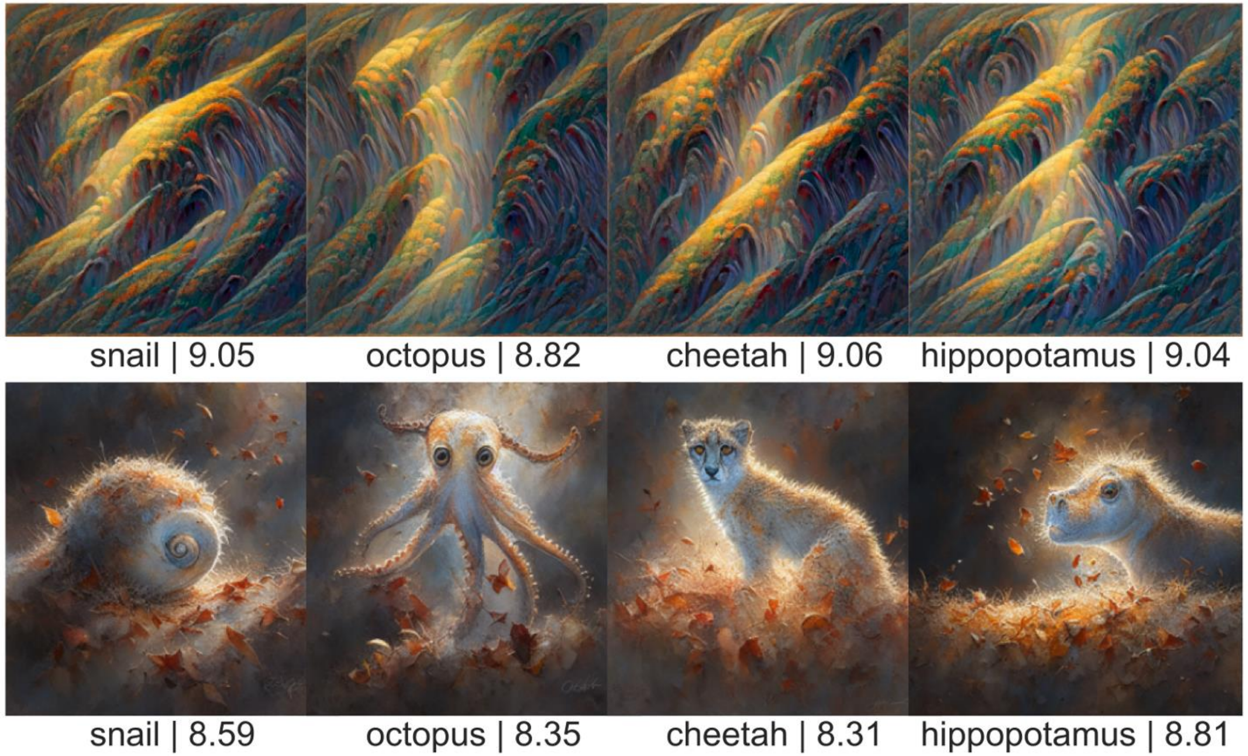
- Reward collapse is a common challenge (Ouyang et al., 2022; Song et al., 2023)
- Guidance also needs to preserve data fidelity



-- Figure from Uehara et al., (2024)

Role of Guidance II --- Preserve Fidelity

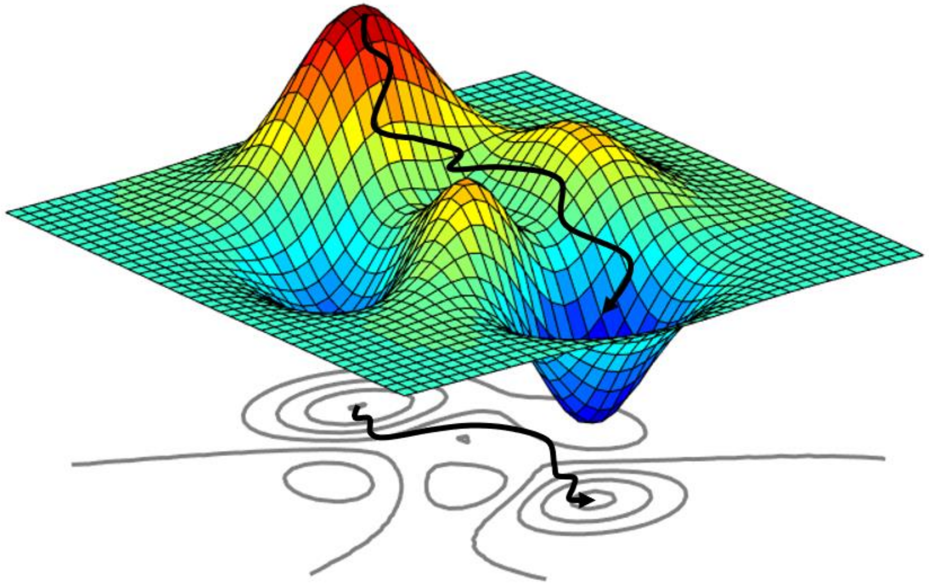
- Reward collapse is a common challenge (Ouyang et al., 2022; Song et al., 2023)
- Guidance also needs to preserve data fidelity



-- Figure from Uehara et al., (2024)

A Generative Optimization Perspective

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



Complex landscape

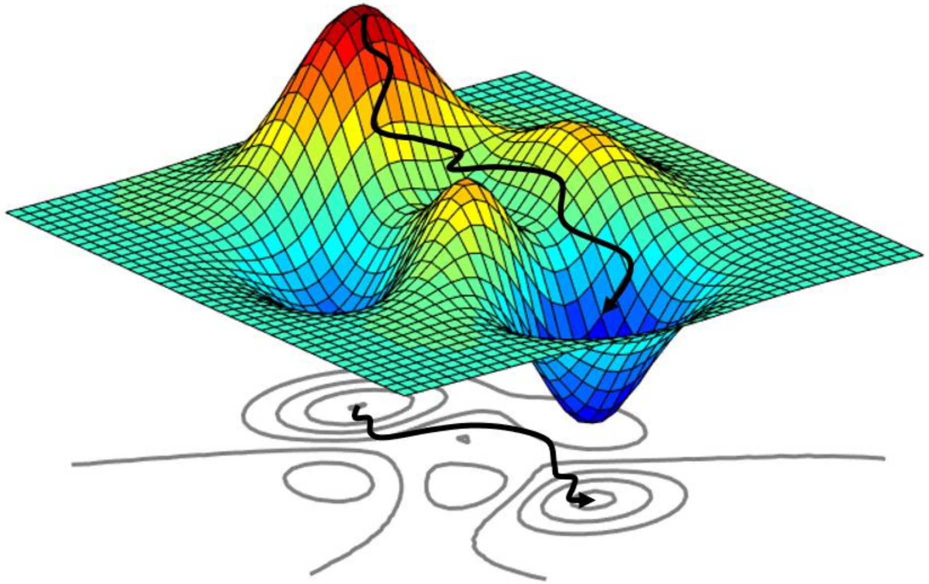
Data fidelity

Generate solution

$$x \sim \mathbb{P}(\cdot \mid f(\cdot) = a)$$

A Generative Optimization Perspective

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



Complex landscape

Data fidelity

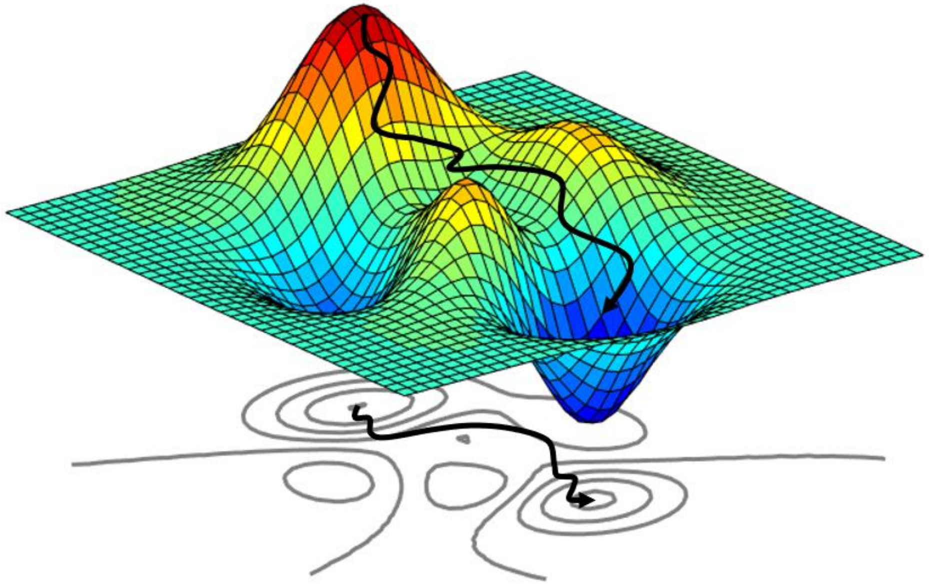
Generative Optimization

Generate solution

$$x \sim \mathbb{P}(\cdot \mid f(\cdot) = a)$$

A Generative Optimization Perspective

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



Complex landscape

Data fidelity

Generative Optimization

Generate solution

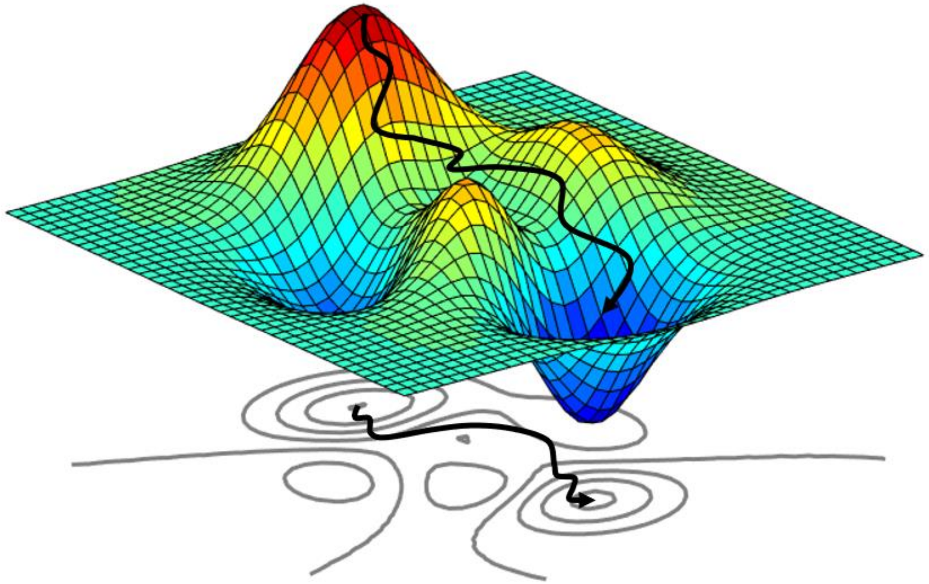
$$x \sim \mathbb{P}(\cdot \mid f(\cdot) = a)$$

Conditional distribution

Guidance design

A Generative Optimization Perspective

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



Complex landscape

Data fidelity

Generative Optimization

Generate solution

$$x \sim \mathbb{P}(\cdot \mid f(\cdot) = a)$$

Guidance

Conditional distribution

Guidance design

Offline with Logged Data:

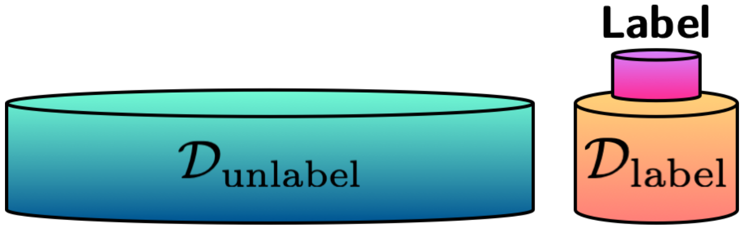
Can we learn a **conditional** diffusion model to generate **high-reward** high-fidelity data?

Problem Setup: Offline Reward Maximization

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

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

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



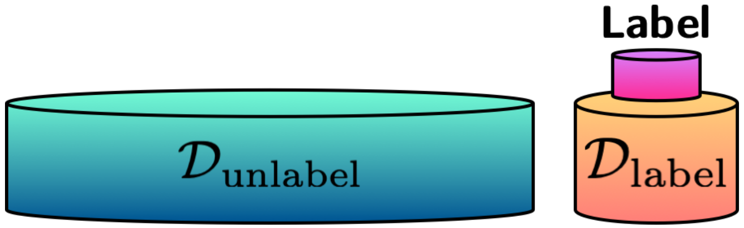
- ➢ ϵ_i is observation noise
- ➢ f^* is reward function
- ➢ x is in a linear subspace

$$x = Az \quad \text{with} \quad A \in \mathbb{R}^{D \times d} \quad z \in \mathbb{R}^d$$

Problem Setup: Offline Reward Maximization

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

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



- ➢ ϵ_i is observation noise
- ➢ f^* is reward function
- ➢ x is in a linear subspace

$$x = Az \quad \text{with} \quad A \in \mathbb{R}^{D \times d} \quad z \in \mathbb{R}^d$$

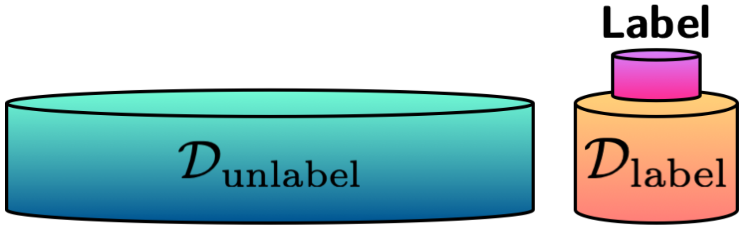
❑ Example: a large collection of unlabeled protein structures; only a few has measured properties.

Problem Setup: Offline Reward Maximization

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

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

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



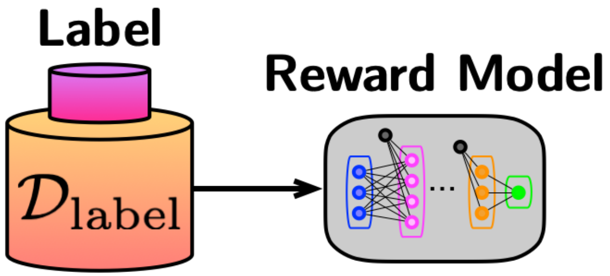
- ➢ ϵ_i is observation noise
- ➢ f^* is reward function
- ➢ x is in a linear subspace

$$x = Az \quad \text{with} \quad A \in \mathbb{R}^{D \times d} \quad z \in \mathbb{R}^d$$

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

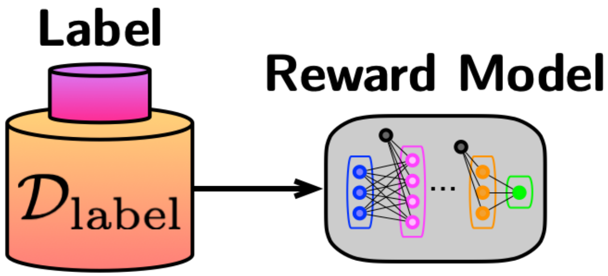
❑ Example: a large collection of unlabeled protein structures; only a few has measured properties.

Meta Algorithm

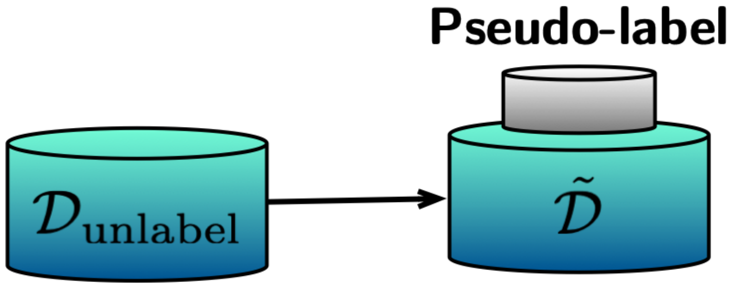


Step 1: Reward Learning

Meta Algorithm

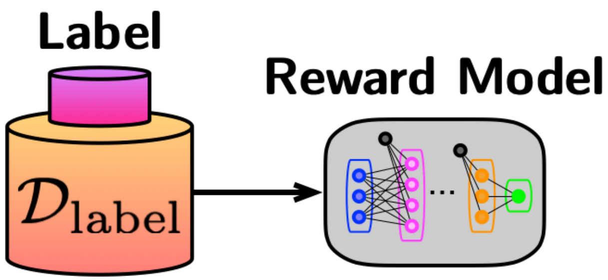


Step 1: Reward Learning

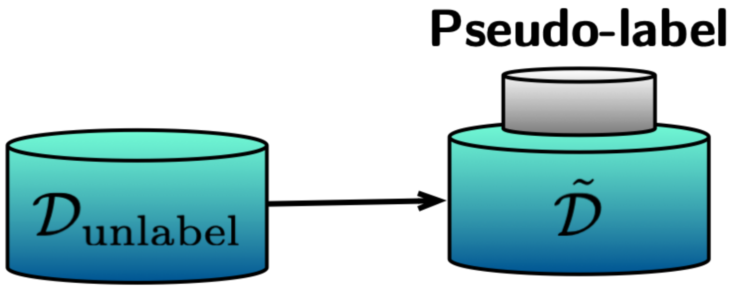


Step 2: Pseudo Labeling

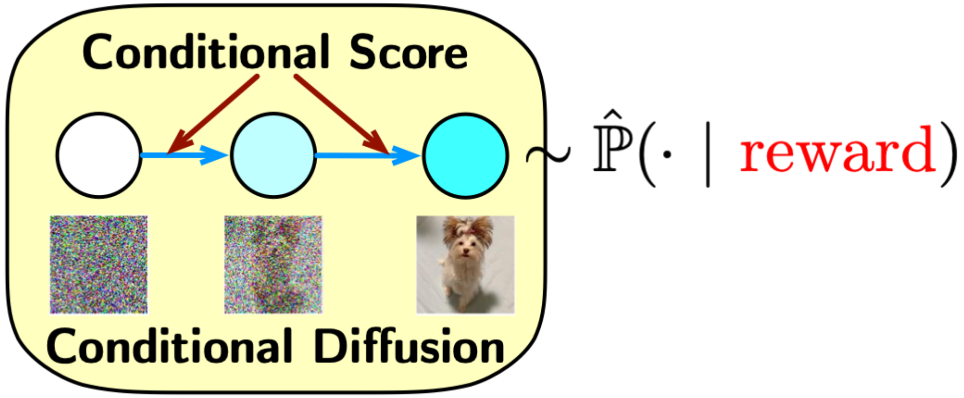
Meta Algorithm



Step 1: Reward Learning

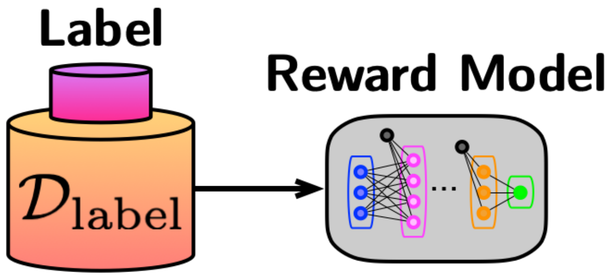


Step 2: Pseudo Labeling

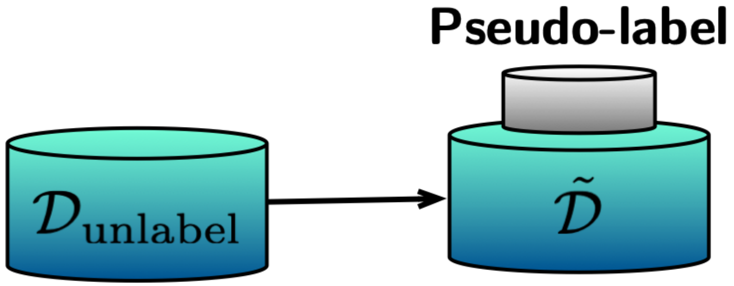


Step 3: Conditional Diffusion Training

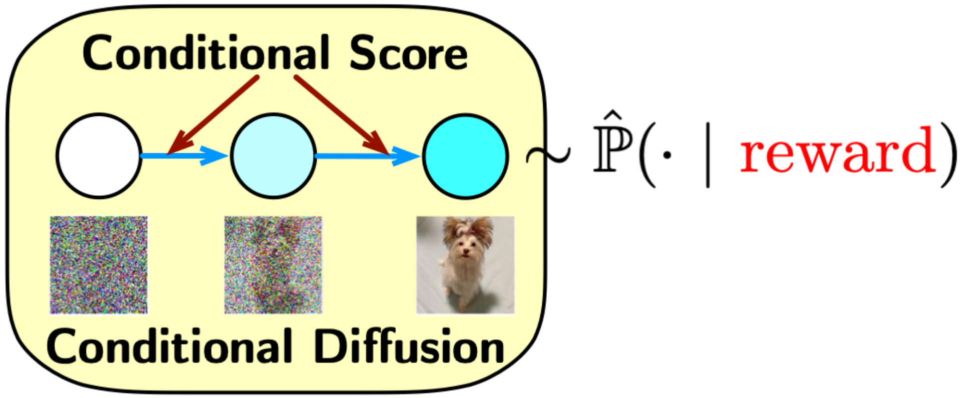
Meta Algorithm



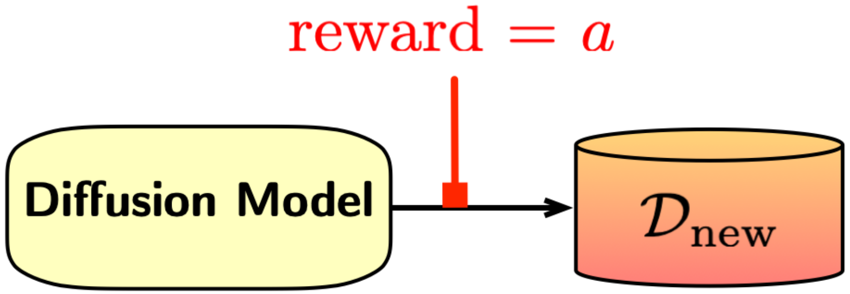
Step 1: Reward Learning



Step 2: Pseudo Labeling



Step 3: Conditional Diffusion Training



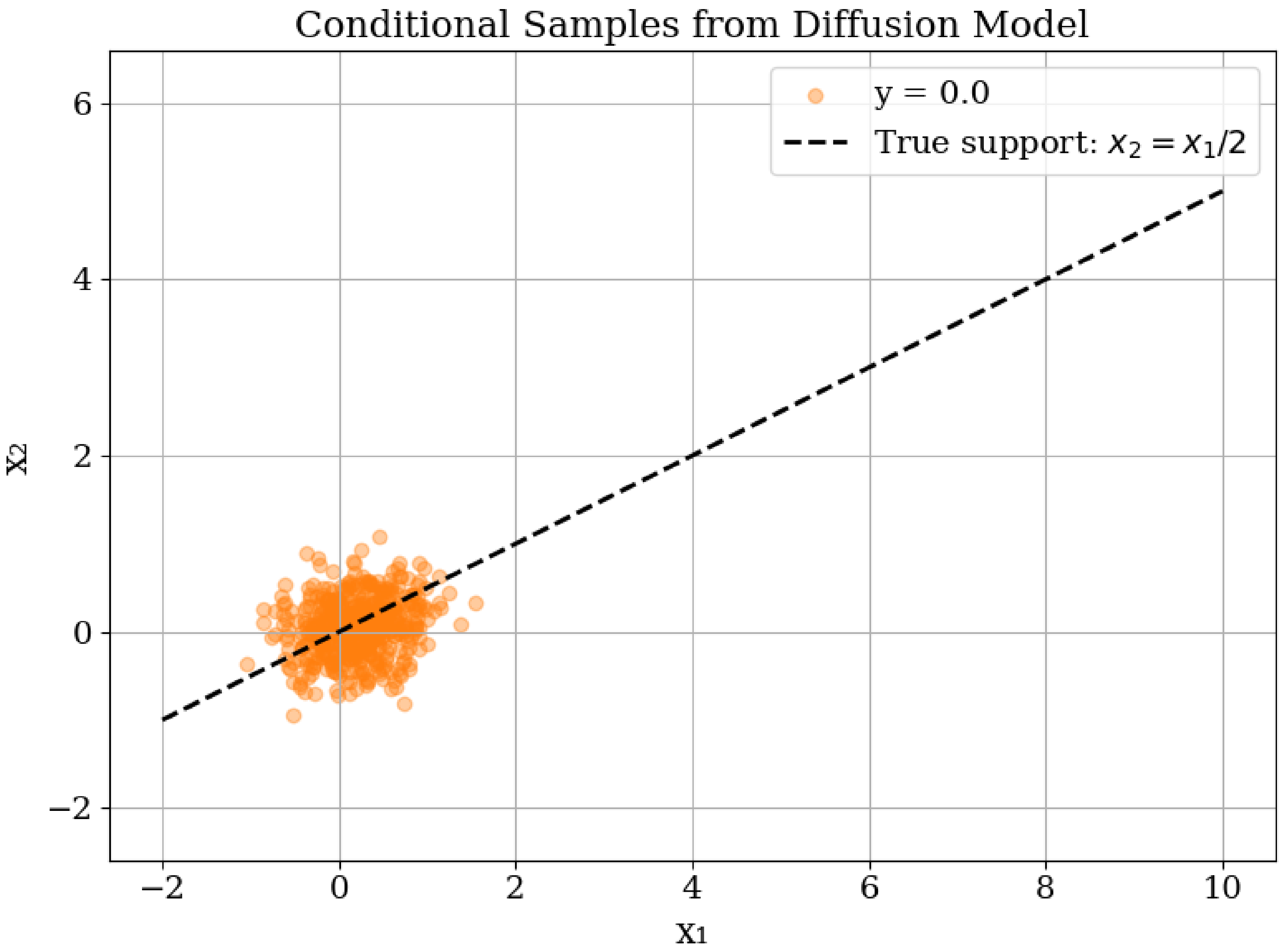
Step 4: Guided Generation

A Toy Example

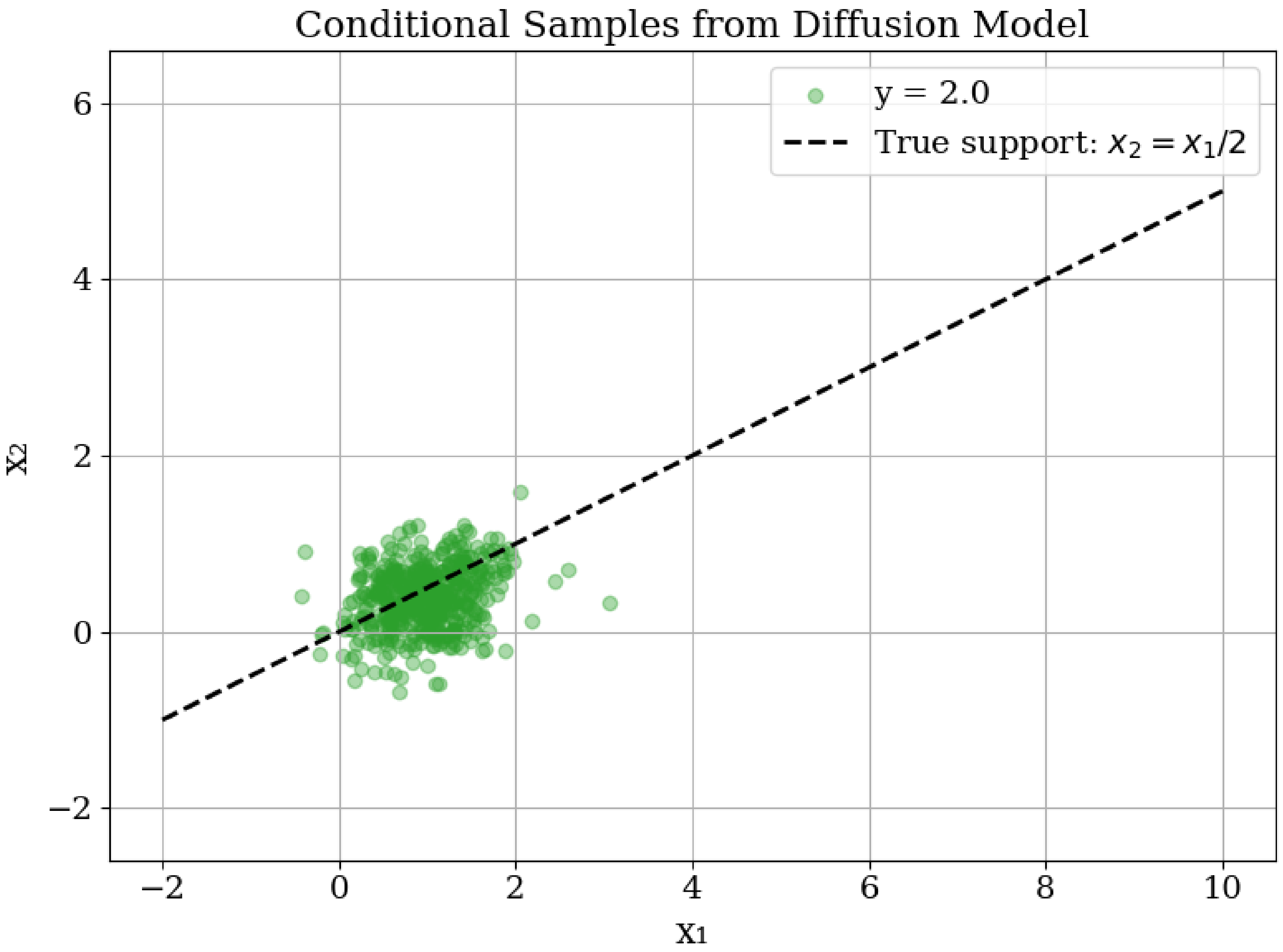
- 2D data *X* with $X = [1, 0.5]^\top z$ for $z \sim \mathrm{N}(0, 1)$
- Linear reward $Y = [1, 1]^\top x + \epsilon$

A Toy Example Cont'd: Good and Bad

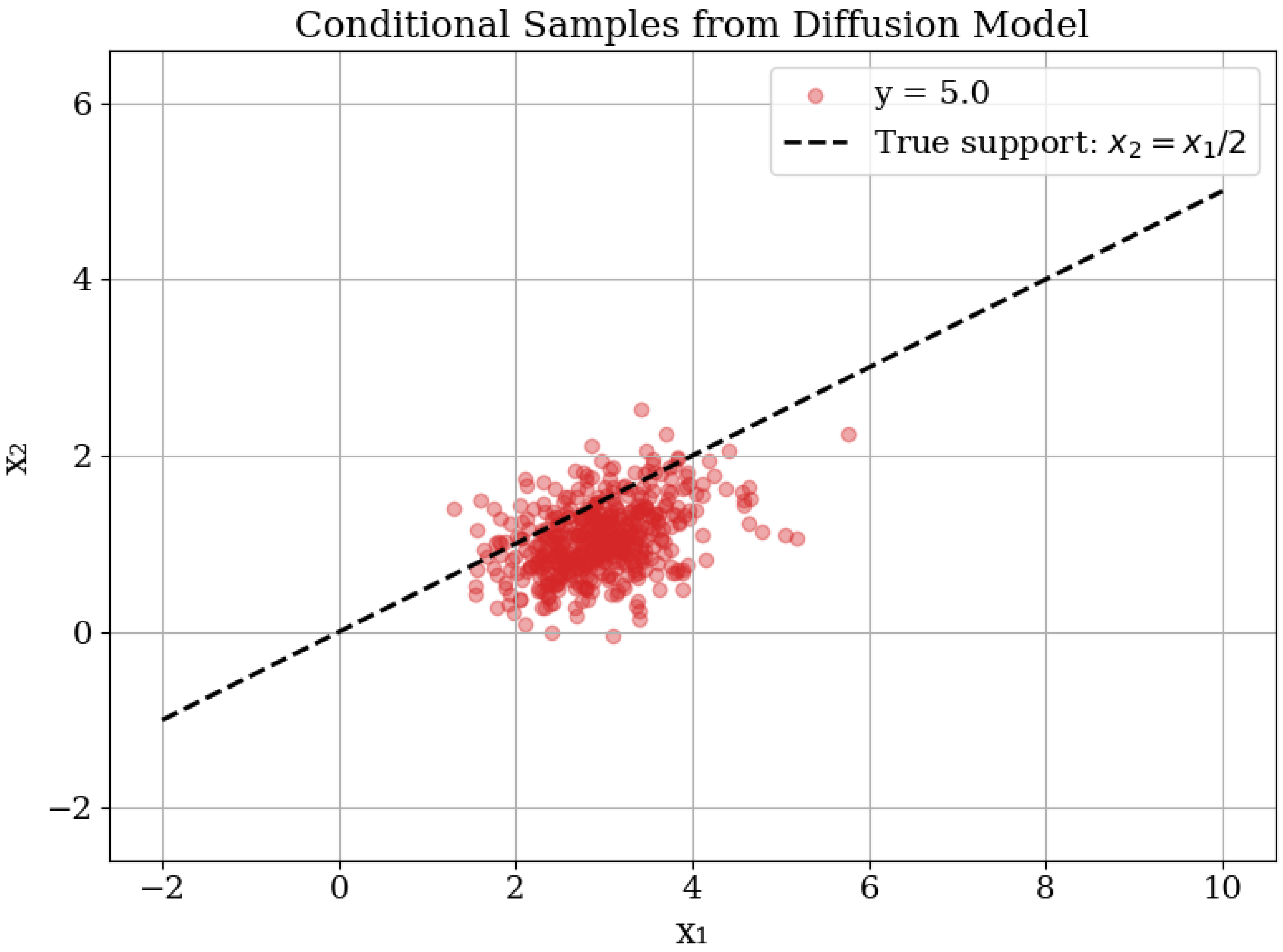
A Toy Example Cont'd: Good and Bad



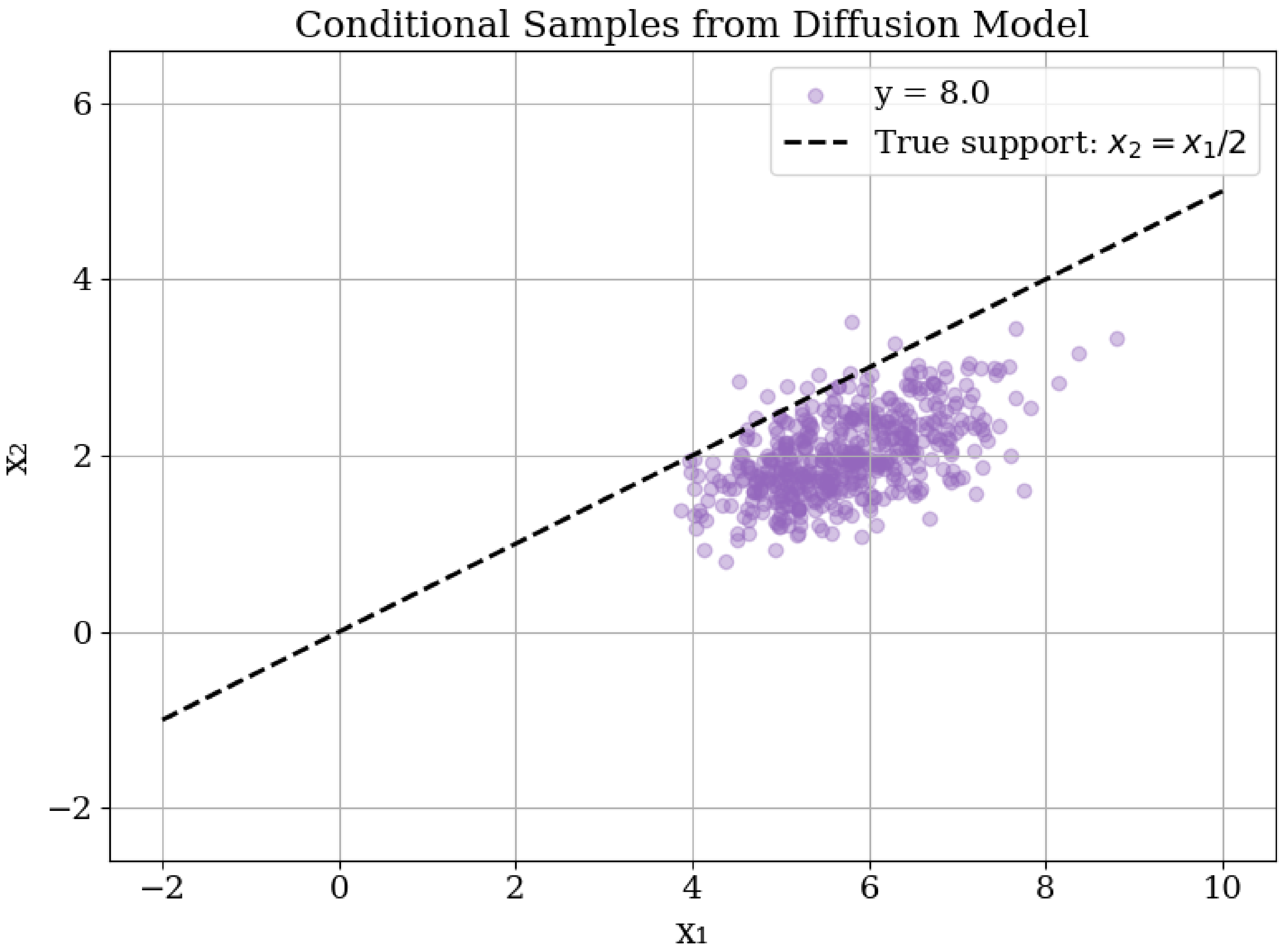
A Toy Example Cont'd: Good and Bad



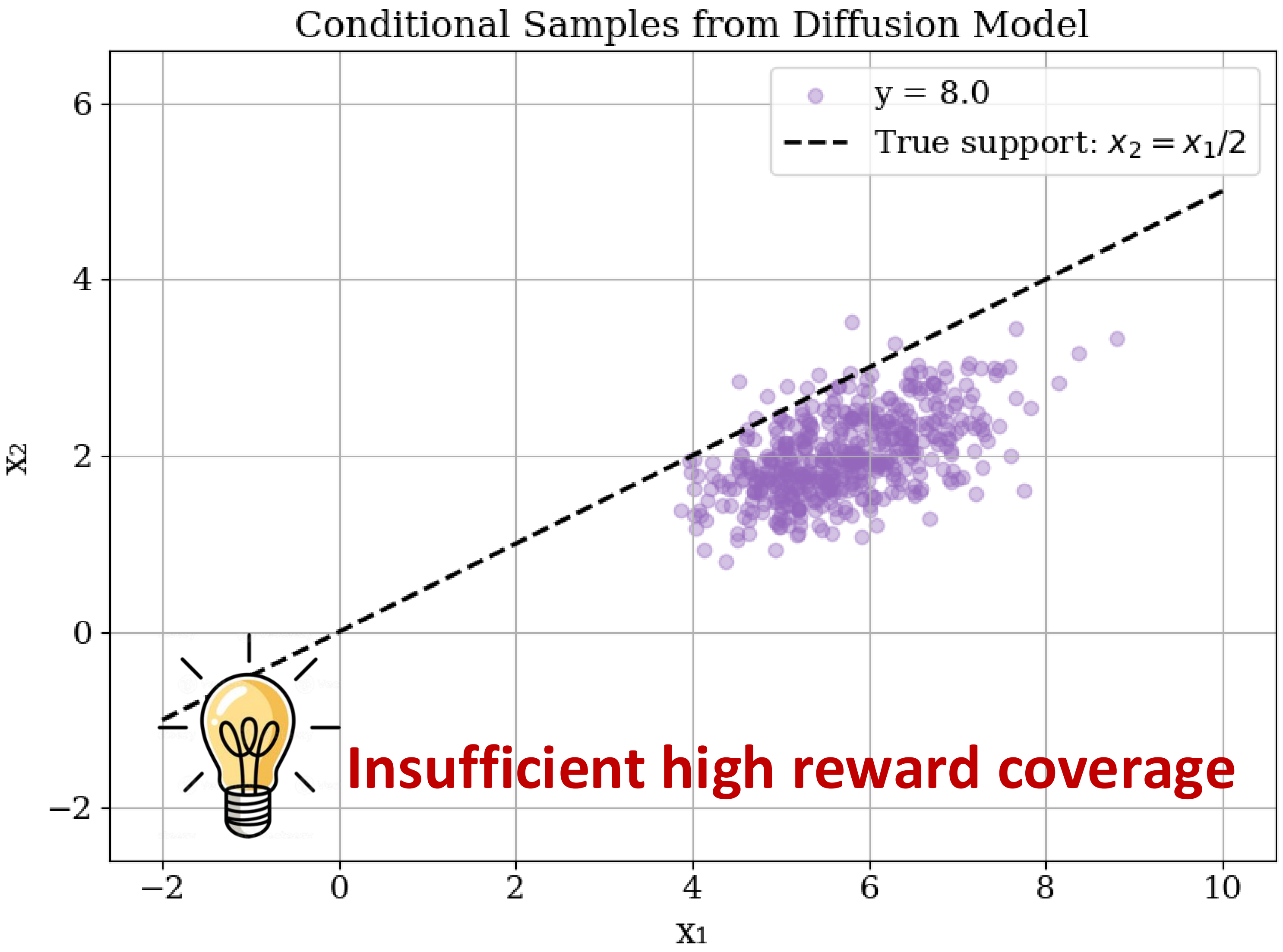
A Toy Example Cont'd: Good and Bad



A Toy Example Cont'd: Good and Bad



A Toy Example Cont'd: Good and Bad



How Far Are We from The Target Reward

- Let a be the target reward of generation

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

How Far Are We from The Target Reward

- Let a be the target reward of generation

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

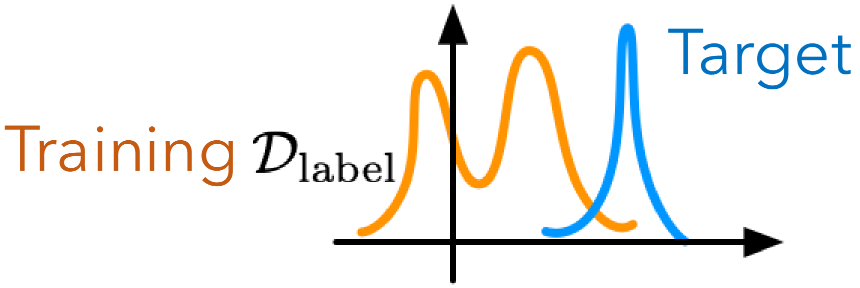
(Reward estimation error)

How Far Are We from The Target Reward

- Let a be the target reward of generation

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

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



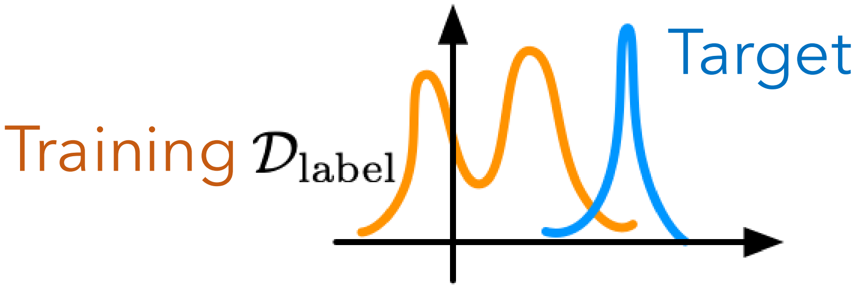
How Far Are We from The Target Reward

- 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)



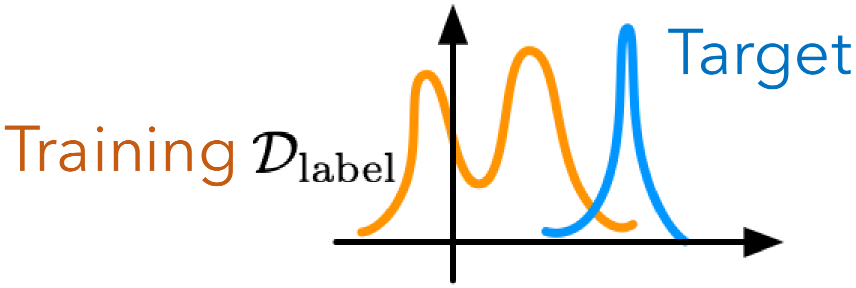
How Far Are We from The Target Reward

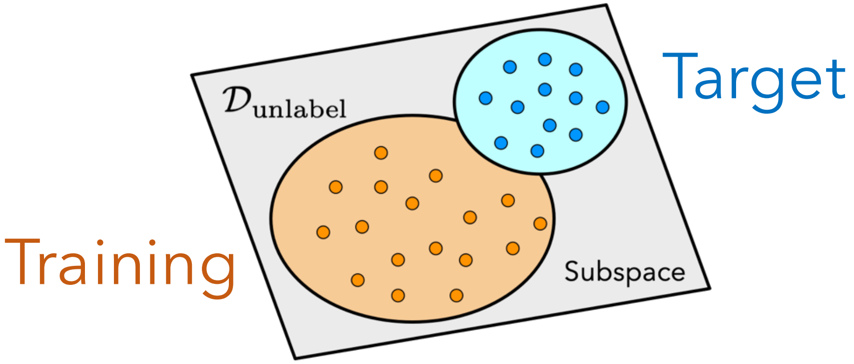
- Let a be the target reward of generation

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

(Reward estimation error) ∘ (Reward distribution shift)

(Conditional diffusion error) ∘ (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 $P_a(\cdot \mid \mathrm{reward} = a)$.

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

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 $P_a(\cdot \mid \mathrm{reward} = a)$.

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

Case Study: Subspace Data + Linear Reward

Theorem

✓ The sub-optimality satisfies

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

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

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

Case Study: Subspace Data + Linear Reward

Theorem

✓ The sub-optimality satisfies

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

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 $P_a(\cdot \mid \mathrm{reward} = a)$.

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

Case Study: Subspace Data + Linear Reward

Theorem

✓ The sub-optimality satisfies

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

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)

-- Z. Li, H. Yuan, 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 Offline Generative Opt.

✓ 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 subspace structures

✓ **Efficiency**: no curse of dimensionality

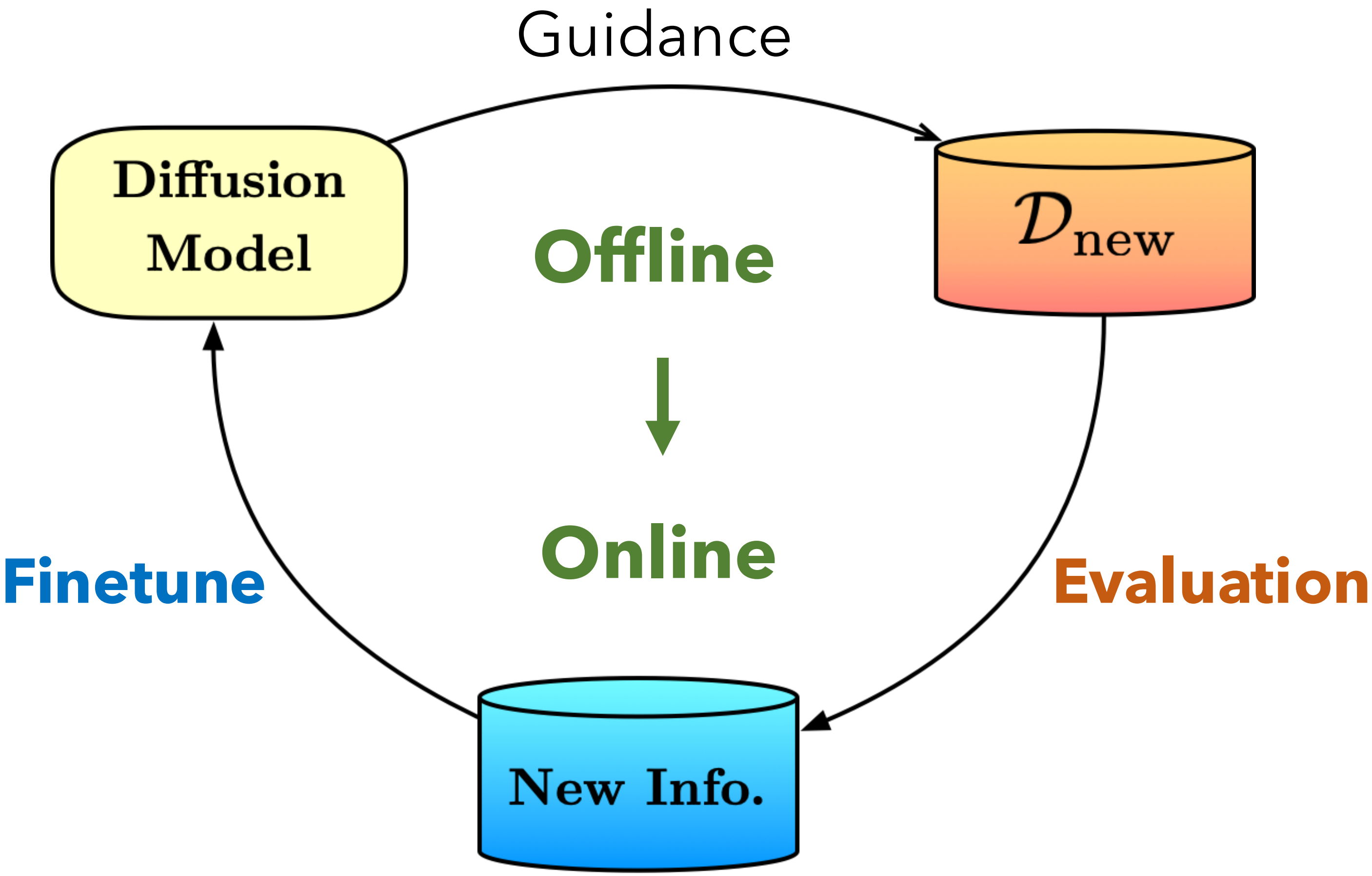
-- Z. Li, H. Yuan, K. Huang, C. Ni, Y. Ye, M. Chen, M. Wang. "Diffusion Model for Data-Driven Black-Box Optimization", Major revision at Management Science

Online with Real-Time Feedback:

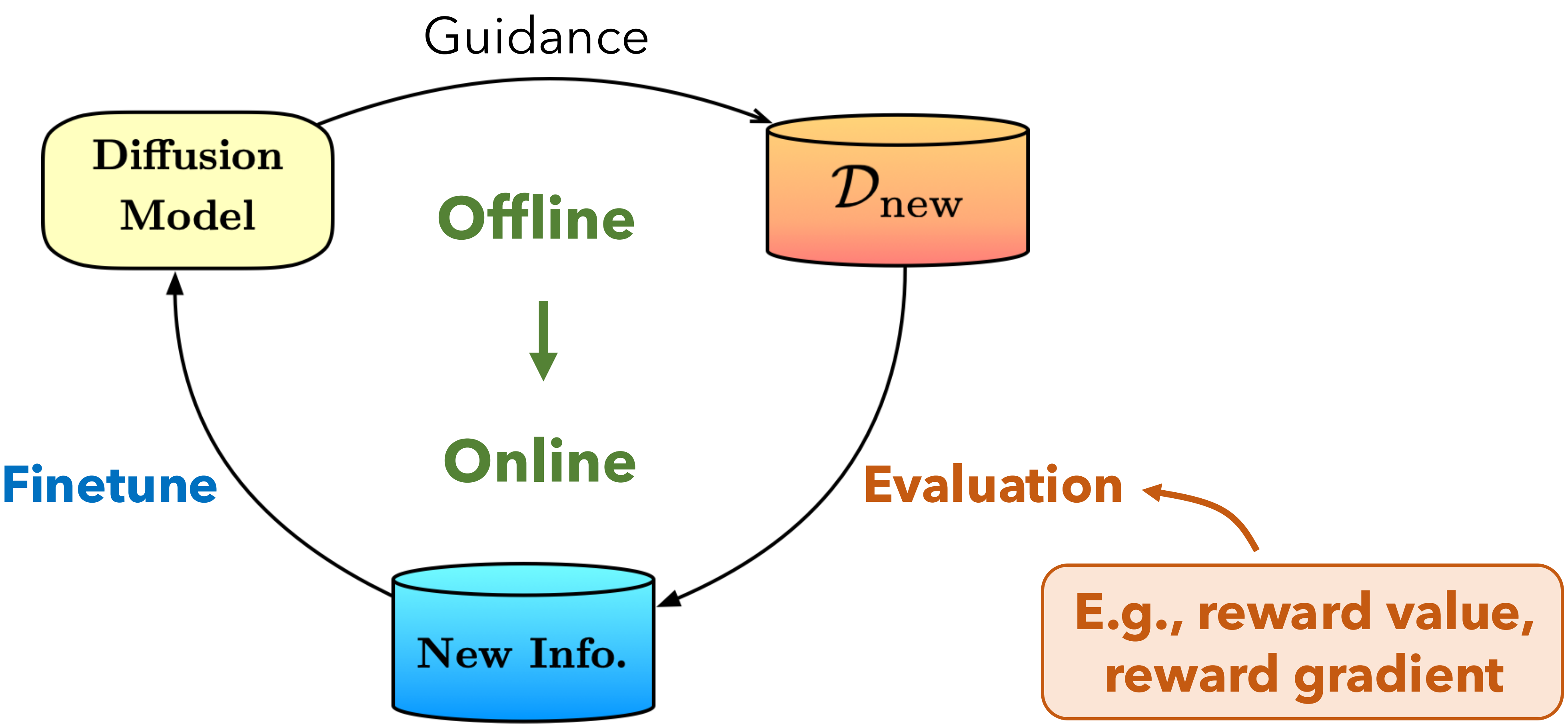
How can we **progressively** fine-tune a diffusion model to generate **max-reward high-fidelity** data?

From Offline to Online

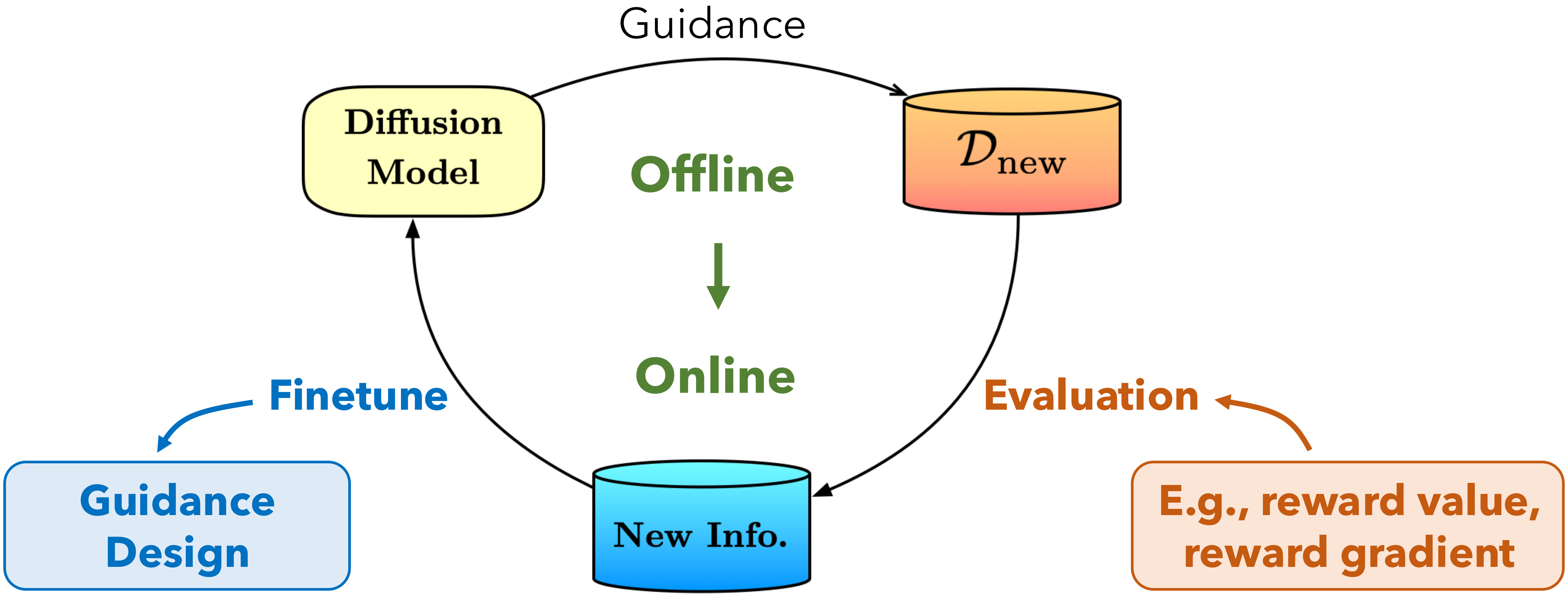
From Offline to Online



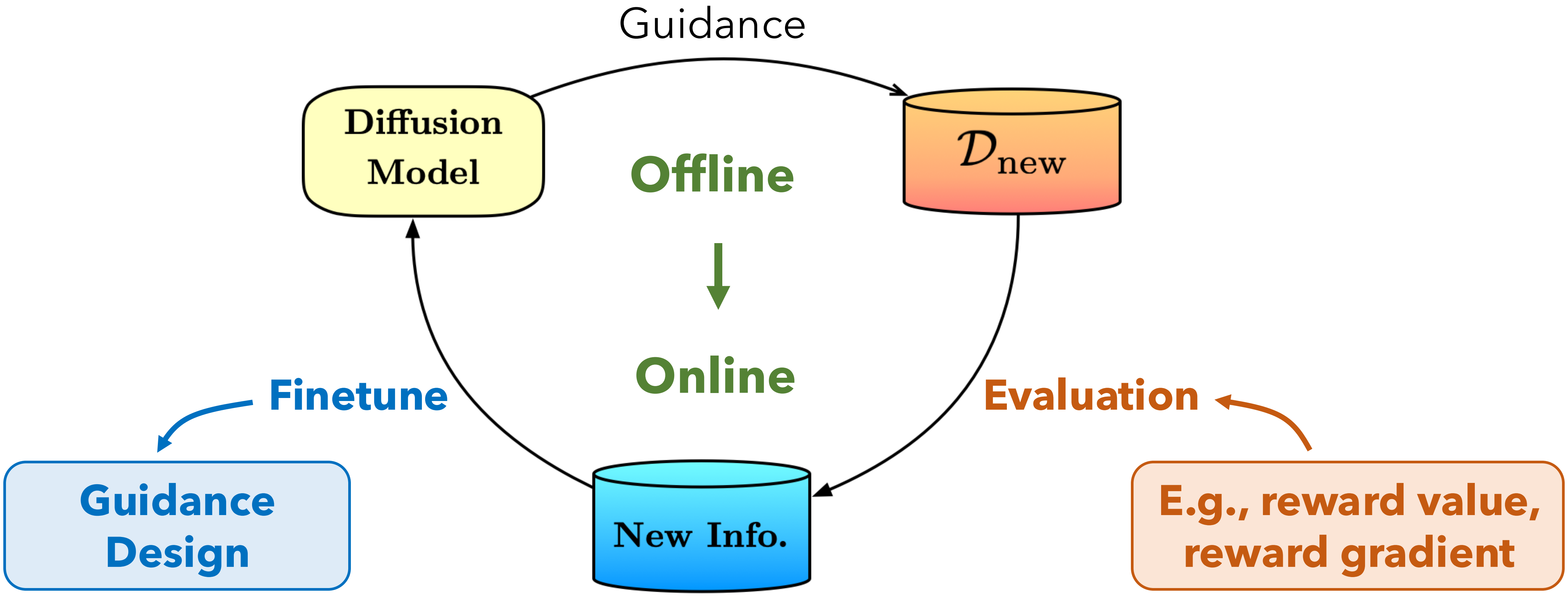
From Offline to Online



From Offline to Online



From Offline to Online

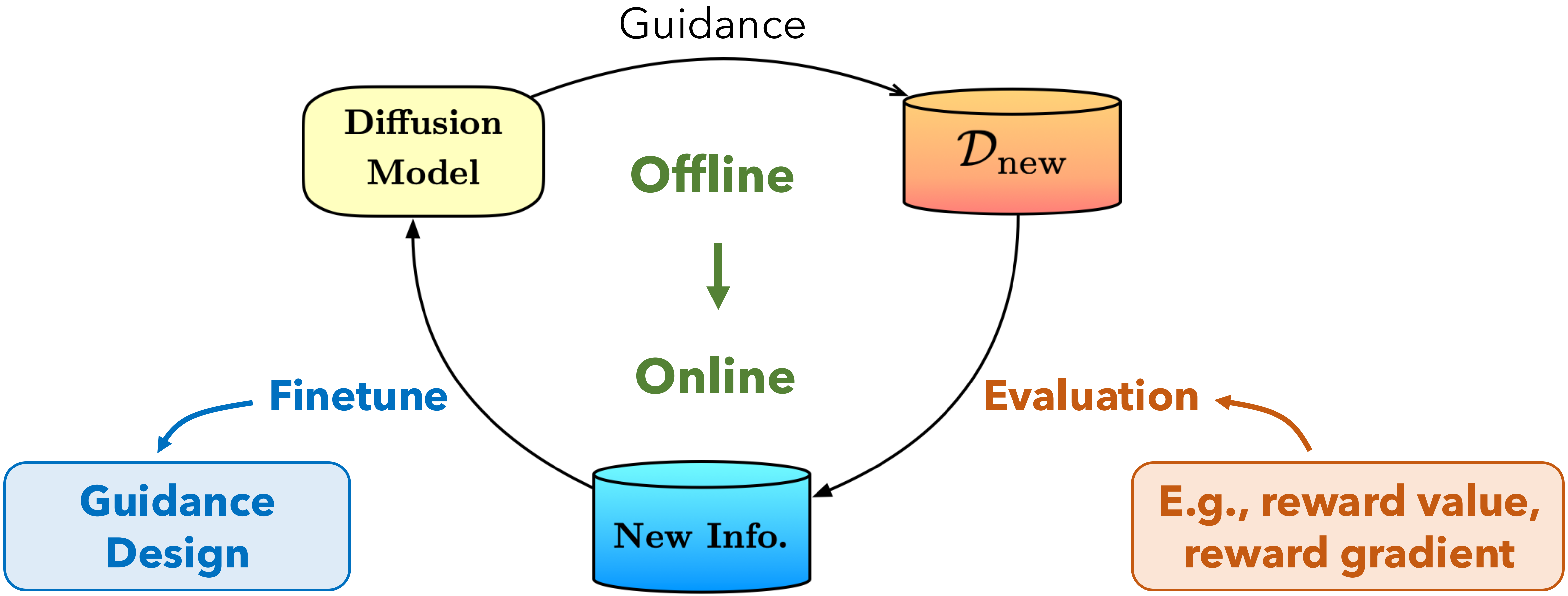


Form of guidance, computation, theoretical guarantees, …

From Offline to Online

Stochastic control method

(Uehara et al., 2024; Han et al., 2024; Tang, 2024)



🤔 Form of guidance, computation, theoretical guarantees, …

Gradient Guidance

Definition

In *general* settings, given a gradient vector *g*, define **gradient guidance** as

$$\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$$

where $\beta(t)$ is some coefficient.

Gradient Guidance

Definition

In *general* settings, given a gradient vector *g*, define **gradient guidance** as

$$\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$$

where $\beta(t)$ is some coefficient.

- Data *X* is Gaussian. Reward function is linear $f(x) = g^\top x$.

Gradient Guidance

Definition
In *general* settings, given a gradient vector *g*, define **gradient guidance** as

$$\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$$

where $\beta(t)$ is some coefficient.

- Data *X* is Gaussian. Reward function is linear $f(x) = g^\top x$.
- Reward evaluation is noisy

$$Y = f(x) + \epsilon \qquad \text{with} \qquad \epsilon \sim \mathsf{N}(0, \sigma^2)$$

Gradient Guidance

Definition

In *general* settings, given a gradient vector *g*, define **gradient guidance** as

$$\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$$

where $\beta(t)$ is some coefficient.

- Data *X* is Gaussian. Reward function is linear $f(x) = g^\top x.$
- Reward evaluation is noisy

$$Y = f(x) + \epsilon \qquad \text{with} \qquad \epsilon \sim \mathsf{N}(0, \sigma^2)$$

- Guidance calculation

$$\nabla \log p_t(x_t | y) = \nabla \log p_t(x_t) + \nabla \log p_t(y | x_t)$$

Gradient Guidance

Definition
In *general* settings, given a gradient vector *g*, define **gradient guidance** as

$$\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$$

where $\beta(t)$ is some coefficient.

- Data *X* is Gaussian. Reward function is linear $f(x) = g^\top x$.
- Reward evaluation is noisy

$$Y = f(x) + \epsilon \qquad \text{with} \qquad \epsilon \sim \mathsf{N}(0, \sigma^2)$$

- Guidance calculation

Gradient Guidance

Definition
In *general* settings, given a gradient vector *g*, define **gradient guidance** as

$$\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$$

where $\beta(t)$ is some coefficient.

- Data *X* is Gaussian. Reward function is linear $f(x) = g^\top x$.
- Reward evaluation is noisy

$$Y = f(x) + \epsilon \qquad \text{with} \qquad \epsilon \sim \mathsf{N}(0, \sigma^2)$$

- Guidance calculation

$$\nabla \log p_t(y | x_t) = -\text{variance}_t^{-1} \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$$

Gradient Guidance

Definition

In *general* settings, given a gradient vector ***g***, define **gradient guidance** as

$$\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$$

where $\beta(t)$ is some coefficient.

- Data *X* is Gaussian. Reward function is linear $f(x) = g^\top x$.
- Reward evaluation is noisy

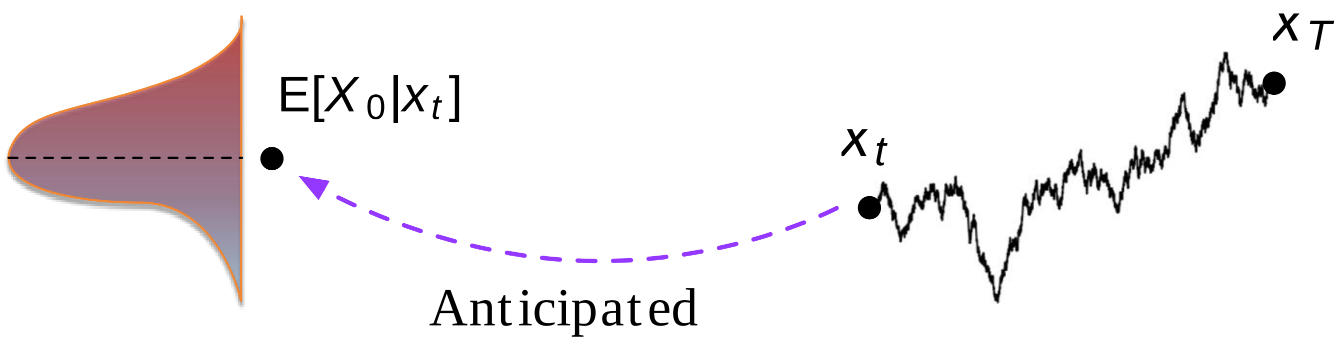
$$Y = f(x) + \epsilon \qquad \text{with} \qquad \epsilon \sim \mathsf{N}(0, \sigma^2)$$

- Guidance calculation

$$\nabla \log p_t(y | x_t) = -\mathsf{variance}_t^{-1} \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$$

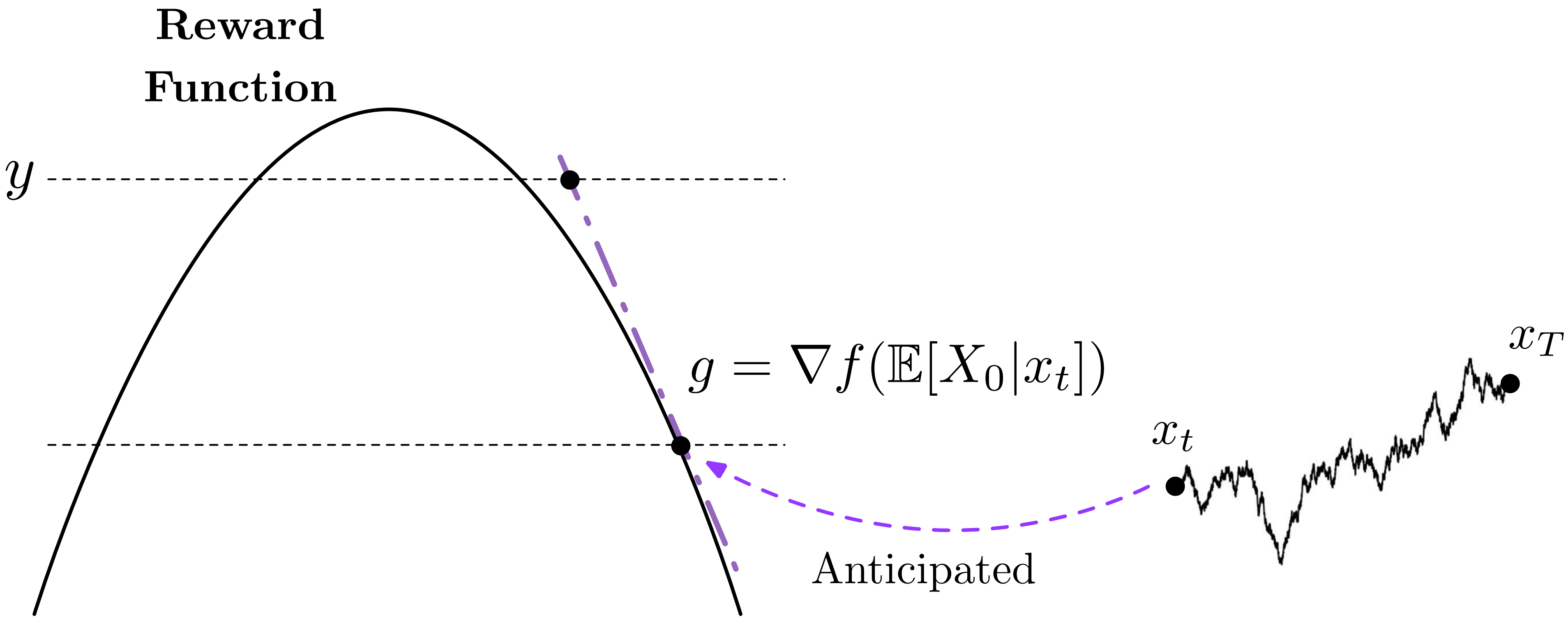
Interpretation of Gradient Guidance

- Gradient guidance $\color{red}{\mathsf{G}(x_t, t)} = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$ steers diffusion model to close a "*look-ahead*" gap



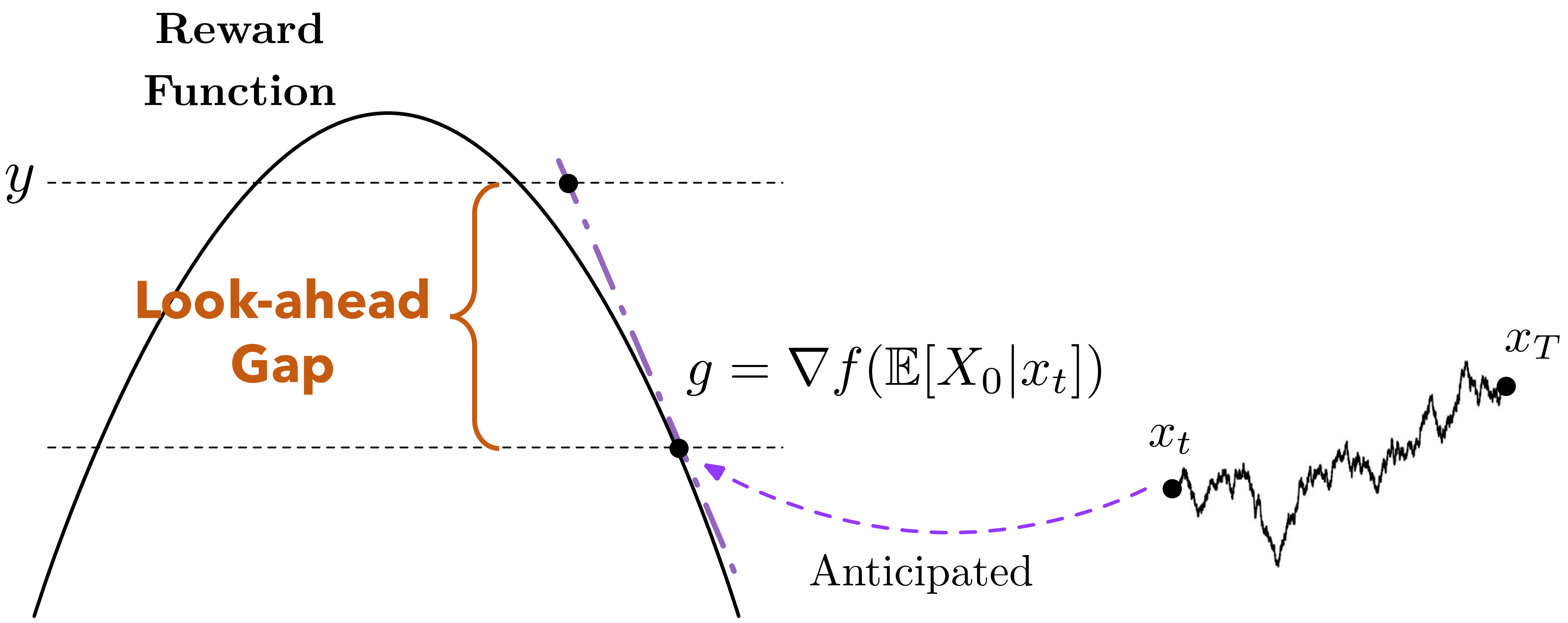
Interpretation of Gradient Guidance

- Gradient guidance $\color{red}{\mathtt{G}(x_t, t)} = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$ steers diffusion model to close a "*look-ahead*" gap



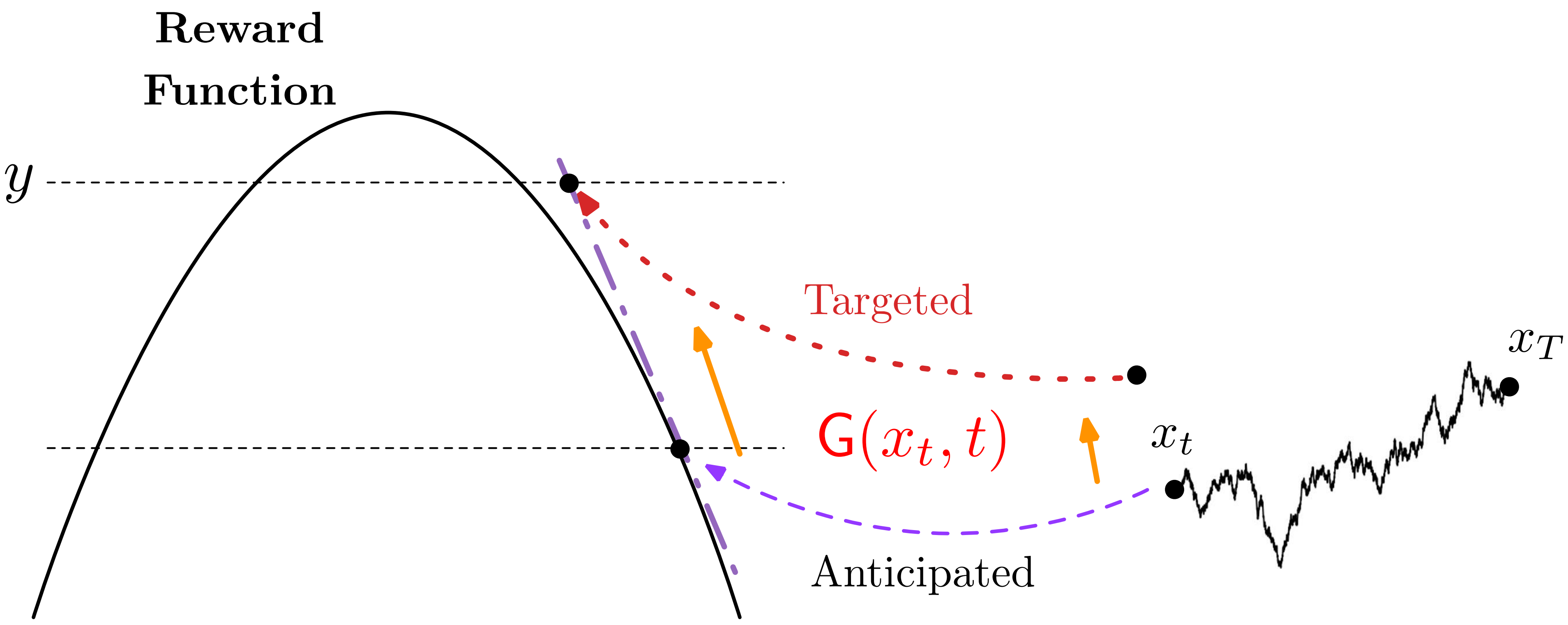
Interpretation of Gradient Guidance

- Gradient guidance $\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$ steers diffusion model to close a "*look-ahead*" gap



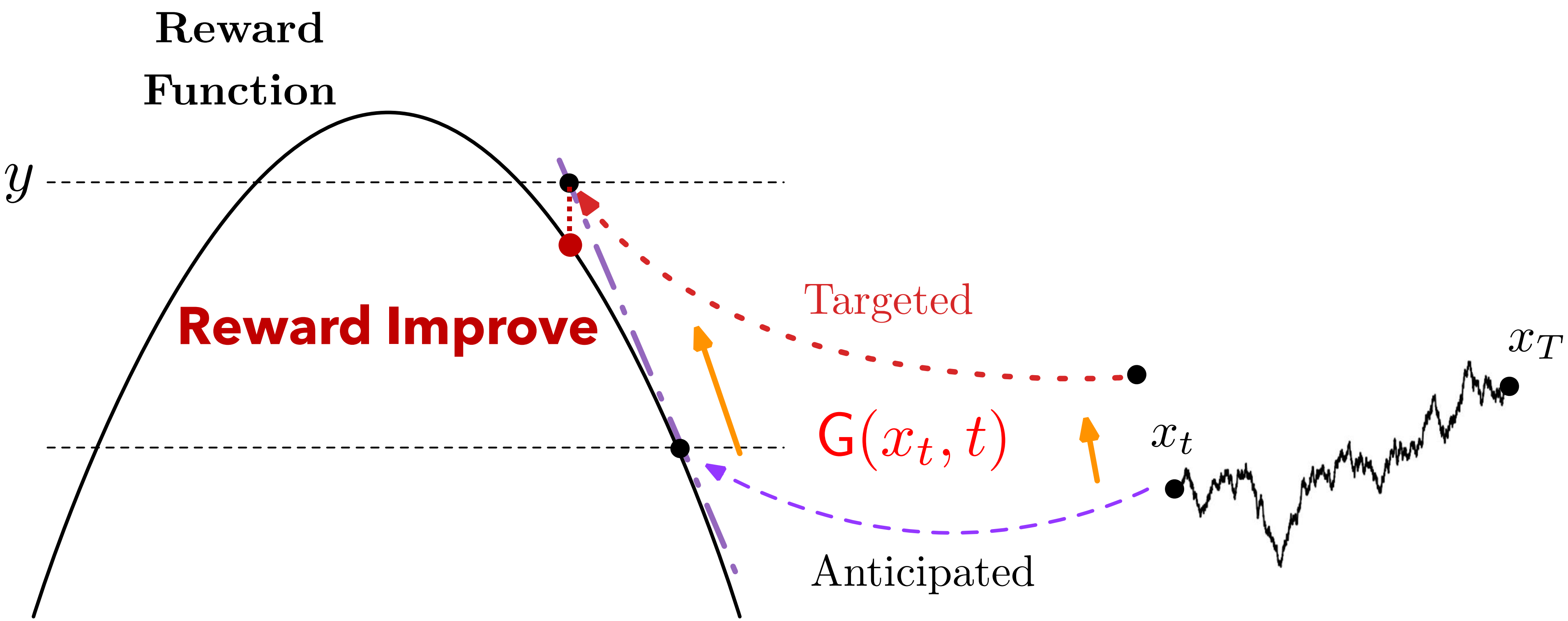
Interpretation of Gradient Guidance

- Gradient guidance $\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0 | x_t]\right)^2$ steers diffusion model to close a "*look-ahead*" gap



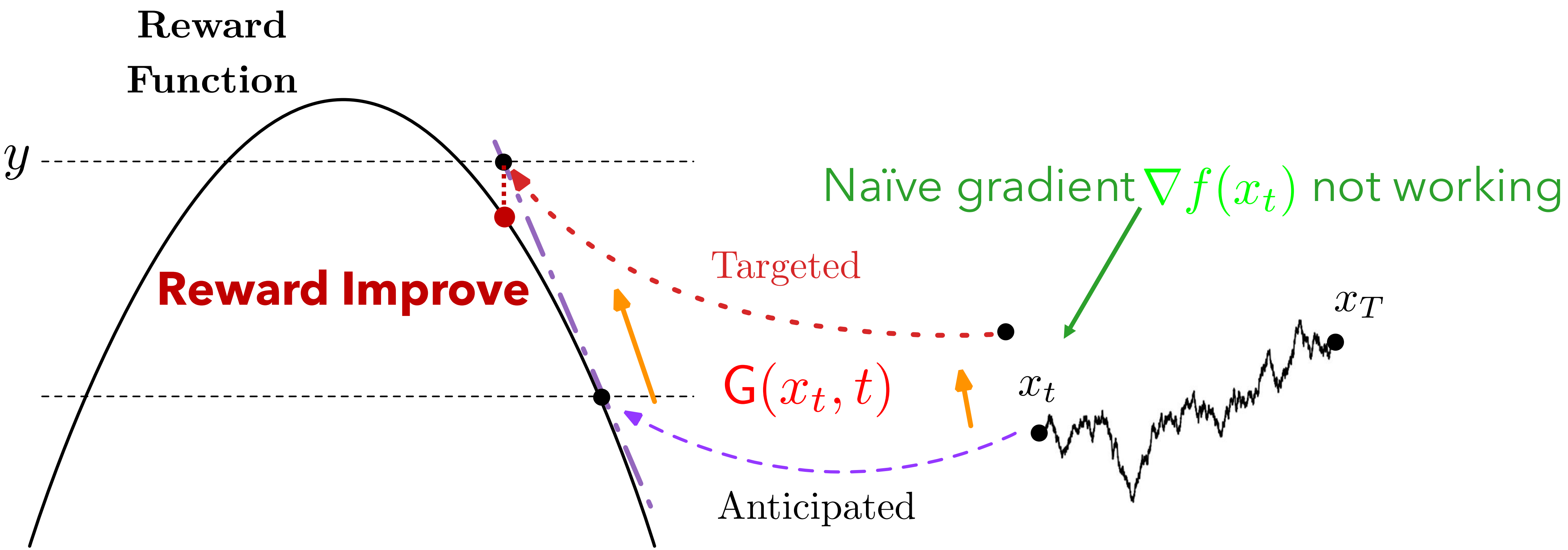
Interpretation of Gradient Guidance

- Gradient guidance $\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0|x_t]\right)^2$ steers diffusion model to close a "*look-ahead*" gap



Interpretation of Gradient Guidance

- Gradient guidance $\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left(y - g^\top \mathbb{E}[X_0|x_t]\right)^2$ steers diffusion model to close a "*look-ahead*" gap



Gradient Guidance Preserves Structure

- Gradient guidance **preserves** subspace structures, but naïve gradient deviates from the subspace

$$\mathsf{G}(x_t, t) = -2\beta(t)(y - g^\top \mathbb{E}[X_0|x_t]) \cdot (\nabla_{x_t}\mathbb{E}[X_0|x_t]) \cdot g$$

Gradient Guidance Preserves Structure

- Gradient guidance **preserves** subspace structures, but naïve gradient deviates from the subspace

$$\mathsf{G}(x_t, t) = -2\beta(t)(y - g^\top \mathbb{E}[X_0|x_t]) \cdot (\nabla_{x_t} \mathbb{E}[X_0|x_t]) \cdot g$$

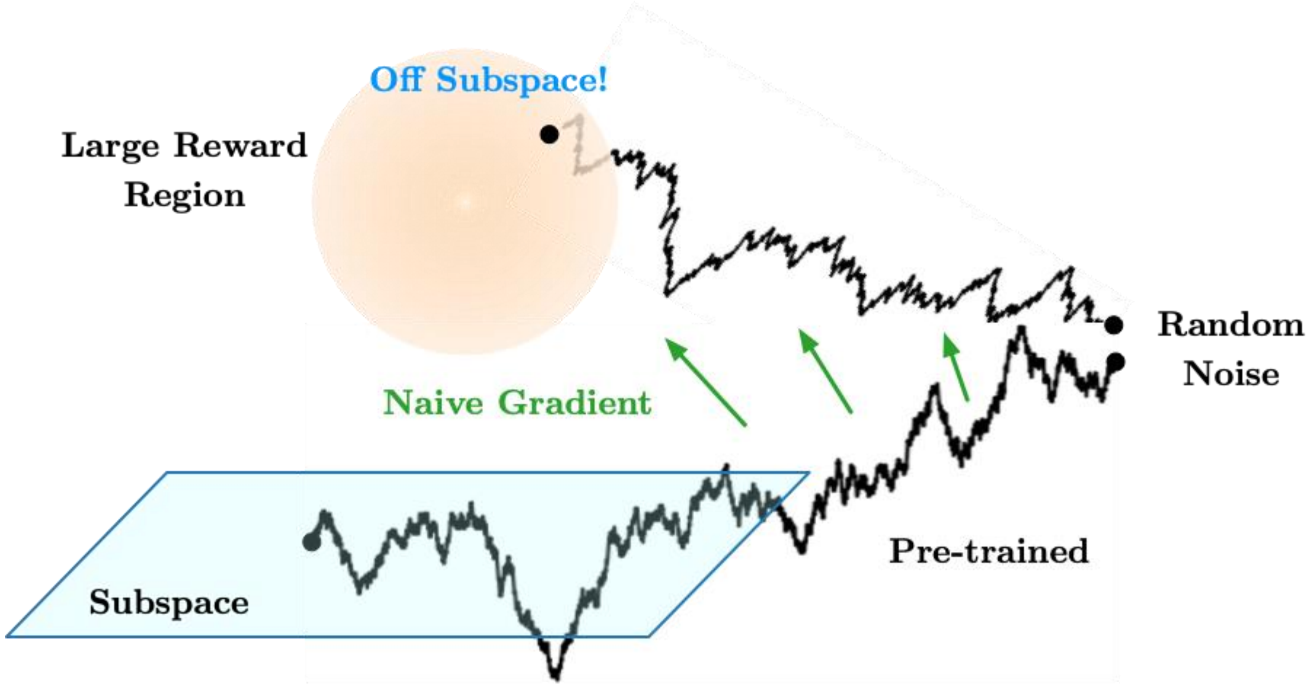
Pre-training induces projection

Gradient Guidance Preserves Structure

- Gradient guidance **preserves** subspace structures, but naïve gradient deviates from the subspace

$$\mathsf{G}(x_t, t) = -2\beta(t)(y - g^\top \mathbb{E}[X_0|x_t]) \cdot (\nabla_{x_t}\mathbb{E}[X_0|x_t]) \cdot g$$

Pre-training induces projection

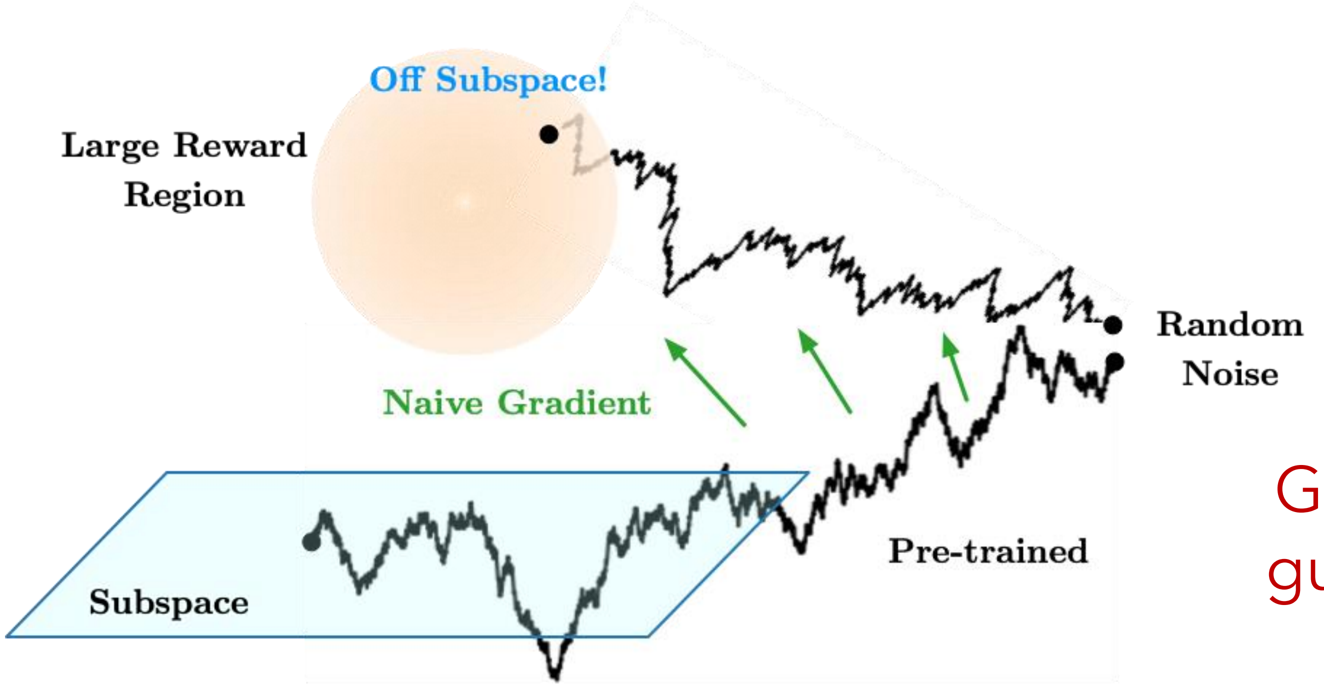


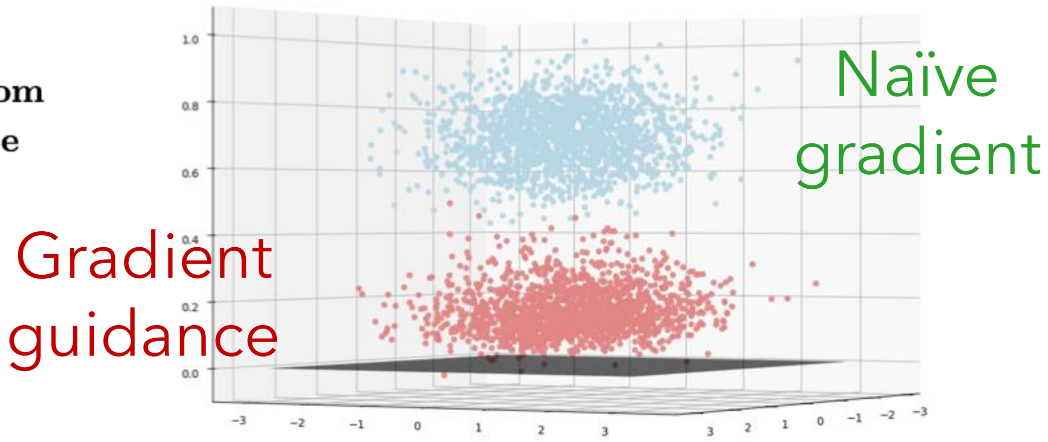
Gradient Guidance Preserves Structure

- Gradient guidance **preserves** subspace structures, but naïve gradient deviates from the subspace

$$\mathsf{G}(x_t, t) = -2\beta(t)(y - g^\top \mathbb{E}[X_0|x_t]) \cdot (\nabla_{x_t}\mathbb{E}[X_0|x_t]) \cdot g$$

Pre-training induces projection





Implementing Gradient Guidance

- Gradient guidance involves an unknown expectation $\mathbb{E}[X_0|x_t]$

Implementing Gradient Guidance

- Gradient guidance involves an unknown expectation $\mathbb{E}[X_0|x_t]$
- Tweedie's formula (Efron, 2011)

Implementing Gradient Guidance

- Gradient guidance involves an unknown expectation $\mathbb{E}[X_0|x_t]$
- Tweedie's formula (Efron, 2011)

$$\mathbb{E}[X_0|x_t] = e^{t/2}(x_t + (1 - e^{-t}) \cdot \texttt{score-function}(x_t, t))$$

Implementing Gradient Guidance

- Gradient guidance involves an unknown expectation $\mathbb{E}[X_0|x_t]$
- Tweedie's formula (Efron, 2011)

$$\mathbb{E}[X_0|x_t] = e^{t/2}(x_t + (1 - e^{-t}) \cdot \texttt{score-function}(x_t, t))$$

- Implementable gradient guidance

Implementing Gradient Guidance

- Gradient guidance involves an unknown expectation $\mathbb{E}[X_0|x_t]$
- Tweedie's formula (Efron, 2011)

$$\mathbb{E}[X_0|x_t] = e^{t/2}(x_t + (1 - e^{-t}) \cdot \texttt{score-function}(x_t, t))$$

- Implementable gradient guidance

$$\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left[y - e^{t/2} g^\top (x_t + (1 - e^{-t}) \cdot \texttt{pretrain-score}(x_t, t)) \right]^2$$

Implementing Gradient Guidance

- Gradient guidance involves an unknown expectation $\mathbb{E}[X_0|x_t]$
- Tweedie's formula (Efron, 2011)

$$\mathbb{E}[X_0|x_t] = e^{t/2}(x_t + (1-e^{-t}) \cdot \texttt{score-function}(x_t, t))$$

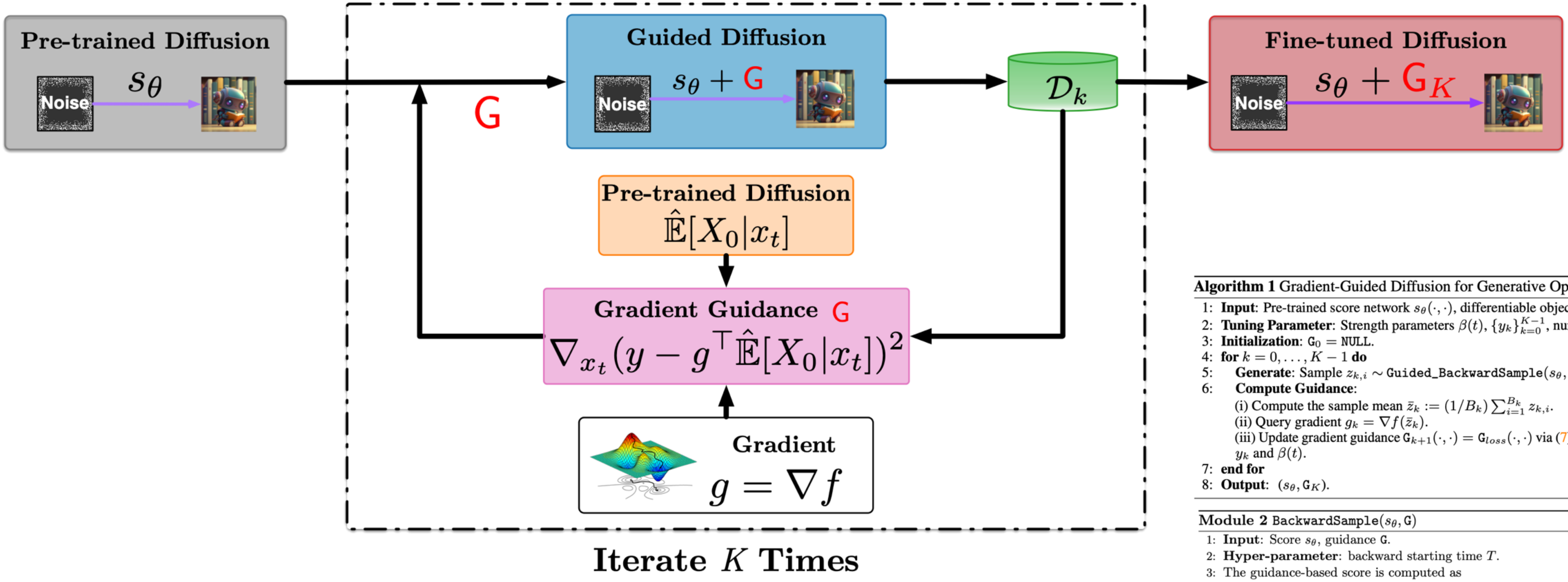
- Implementable gradient guidance

$$\mathsf{G}(x_t, t) = -\beta(t) \cdot \nabla_{x_t} \left[y - e^{t/2} g^\top (x_t + (1-e^{-t}) \cdot \texttt{pretrain-score}(x_t, t)) \right]^2$$

- The gradient can be found by auto-differentiation
- Coefficient $\beta(t)$ is a tuning parameter, akin to a step size

Gradient Guidance Algorithm

$$\mathrm{d}X_t^{\leftarrow} = \left[\frac{1}{2}X_t^{\leftarrow} + s_\theta\left(X_t^{\leftarrow}, T-t\right) + \mathtt{G}\left(X_t^{\leftarrow}, T-t\right)\right]\mathrm{d}t + \mathrm{d}\overline{W}_t$$



Algorithm 1 Gradient-Guided Diffusion for Generative Optimization

Input: Pre-trained score network $s_\theta(\cdot,\cdot)$, differentiable objective function f.
Tuning Parameter: Strength parameters $\beta(t)$, $\{y_k\}_{k=0}^{K-1}$, number of iterations K, batch sizes $\{B_k\}$.
Initialization: $\mathtt{G}_0 = \mathtt{NULL}$.
for $k = 0, \ldots, K-1$ **do**
  Generate: Sample $z_{k,i} \sim \mathtt{Guided\_BackwardSample}(s_\theta, \mathtt{G}_k)$ using Module 1, for $i \in [B_k]$.
  Compute Guidance:
  (i) Compute the sample mean $\bar{z}_k := (1/B_k)\sum_{i=1}^{B_k} z_{k,i}$.
  (ii) Query gradient $g_k = \nabla f(\bar{z}_k)$.
  (iii) Update gradient guidance $\mathtt{G}_{k+1}(\cdot,\cdot) = \mathtt{G}_{loss}(\cdot,\cdot)$ via (7), using s_θ, gradient vector g_k, and parameters y_k and $\beta(t)$.
end for
Output: $(s_\theta, \mathtt{G}_K)$.

Module 2 $\mathtt{BackwardSample}(s_\theta, \mathtt{G})$

Input: Score s_θ, guidance $\mathtt{G}$.
Hyper-parameter: backward starting time T.
The guidance-based score is computed as

$$s(x_t, t) = s_\theta(x_t, t) + \mathtt{G}(x_t, t),$$

Sample from backward process:

$$\mathrm{d}X_t^{\leftarrow} = \left[\frac{1}{2}X_t^{\leftarrow} + s_\theta\left(X_t^{\leftarrow}, T-t\right) + \mathtt{G}\left(X_t^{\leftarrow}, T-t\right)\right]\mathrm{d}t + \mathrm{d}\overline{W}_t.$$

Output: $z = X_T^{\leftarrow}$.

Convergence to Regularized Optima

Theorem

Suppose the reward function is concave and *L*-smooth. Consider linear pre-trained score. With high probability, it holds

$$f(x^*_{A,\lambda}) - f(\mu_K) = \lambda(L/\lambda)^K \cdot \tilde{\mathcal{O}}(d)$$

where $\lambda = \mathcal{O}(L)$, μ_K is the mean of generated samples, and $x^*_{A,\lambda}$ is the maximizer of

$$x^*_{A,\lambda} = \arg\max_{x \in \text{subspace}} f(x) - \frac{\lambda}{2}\|x - \bar{\mu}_0\|^2_{\bar{\Sigma}_0^{-1}}$$

with $\bar{\mu}_0$ and $\bar{\Sigma}_0$ the mean and covariance of the pre-training data.

Convergence to Regularized Optima

Theorem
Suppose the reward function is concave and *L*-smooth. Consider linear pre-trained score. With high probability, it holds

$$f(x^*_{A,\lambda}) - f(\mu_K) = \lambda (L/\lambda)^K \cdot \tilde{\mathcal{O}}(d)$$

where $\lambda = \mathcal{O}(L)$, μ_K is the mean of generated samples, and $x^*_{A,\lambda}$ is the maximizer of

$$x^*_{A,\lambda} = \arg \max_{x \in \text{subspace}} f(x) - \frac{\lambda}{2} \|x - \bar{\mu}_0\|^2_{\bar{\Sigma}_0^{-1}}$$

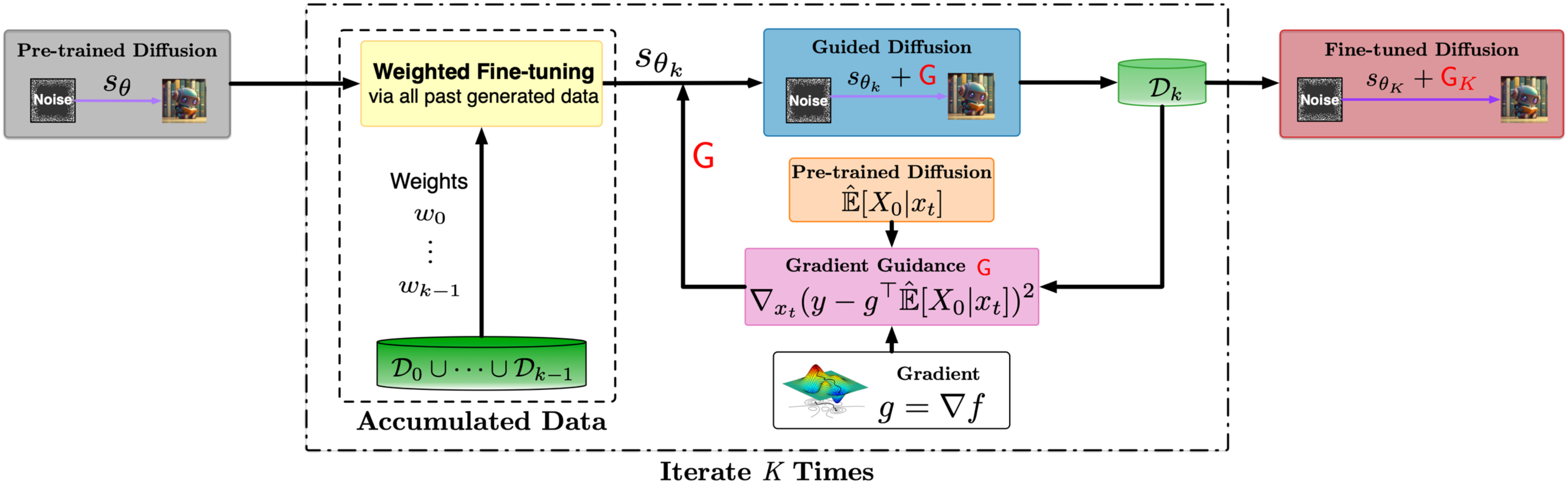
with $\bar{\mu}_0$ and $\bar{\Sigma}_0$ the mean and covariance of the pre-training data.

Effective gradient-guided diffusion for optimization:

✓ **Linear** convergence and **intrinsic** dimension dependence

✓ Pre-training induces **regularization**

Algorithm with Adaptive Pre-trained Score



Algorithm 2 Gradient-Guided Diffusion with **Adaptive Fine-tuning**

Input: Pre-trained score $s_\theta(\cdot,\cdot)$, differentiable objective function f.
Tuning Parameter: strength parameter $\beta(t)$, $\{y_k\}_{k=0}^{K-1}$, weights $\{\{w_{k,i}\}_{i=0}^{k}\}_{k=0}^{K-1}$, number of iterations K, batch sizes $\{B_k\}$.
Initialize: $s_{\theta_0} = s_\theta$, $\mathtt{G}_0 = \mathtt{NULL}$.
for $k = 0, \cdots, K-1$ **do**
  Generate: Sample a batch $\mathcal{D}_k = \{z_{k,i}\}_{i=1}^{B_k}$ from `Guided_BackwardSample`$(s_{\theta_k}, \mathtt{G}_k)$ (Module 1).
  Compute Guidance:
  (i) Compute sample mean $\bar{z}_k = (1/B_k)\sum_{i=1}^{B_k} z_{k,i}$, and query gradient $g_k = \nabla f(\bar{z}_k)$.
  (ii) Update s_{θ_k} to $s_{\theta_{k+1}}$ by minimizing the re-weighted objective (14).
  (iii) Compute $\mathtt{G}_{k+1}(\cdot,\cdot) = \mathtt{G}_{loss}(\cdot,\cdot)$ in (7), using $s_{\theta_{k+1}}$ and g_k, with parameter $y_k, \beta(t)$.
end for
Output: $(s_{\theta_K}, \mathtt{G}_K)$.

Module 2 `BackwardSample`$(s_\theta, \mathtt{G})$

Input: Score s_θ, guidance $\mathtt{G}$.
Hyper-parameter: backward starting time T.
The guidance-based score is computed as

$$s(x_t, t) = s_\theta(x_t, t) + \mathtt{G}(x_t, t),$$

Sample from backward process:

$$\mathrm{d}X_t^{\leftarrow} = \left[\frac{1}{2}X_t^{\leftarrow} + s_\theta\left(X_t^{\leftarrow}, T-t\right) + \mathtt{G}\left(X_t^{\leftarrow}, T-t\right)\right]\mathrm{d}t + \mathrm{d}\overline{W}_t.$$

Output: $z = X_T^{\leftarrow}$.

Global Convergence with Adapted Score

Theorem

Suppose the reward function is concave and L-smooth. Consider adapting a linear pre-trained score. It holds that

$$f_A^* - f(\mu_K) = \tilde{O}\left(dL^2/K\right)$$

where $f_A^* = \max_{x \in \text{subspace}} f(x)$ is the global maximum.

-- Q. Ying, H. Yuan, Y. Yang, M. Chen, M. Wang. "Gradient Guidance for Diffusion Models", NeurIPS 2024

Global Convergence with Adapted Score

Theorem
Suppose the reward function is concave and *L*-smooth. Consider adapting a linear pre-trained score. It holds that

$$f_A^* - f(\mu_K) = \tilde{O}\left(dL^2/K\right)$$

where $f_A^* = \max_{x \in \text{subspace}} f(x)$ is the global maximum.

Effective gradient-guided diffusion for optimization:

✓ *1/K* **global** convergence

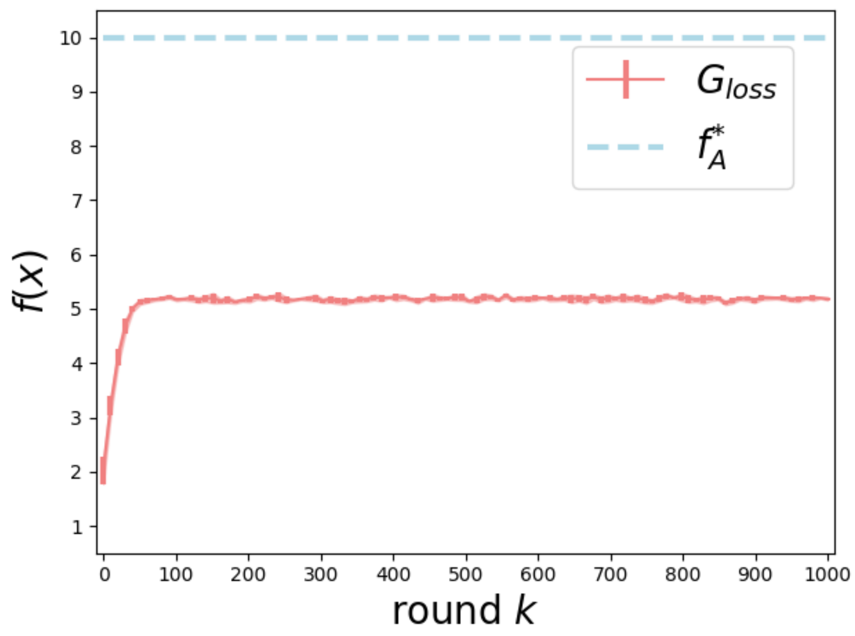
✓ **intrinsic** dimension dependence

✓ Preservation of subspace structure

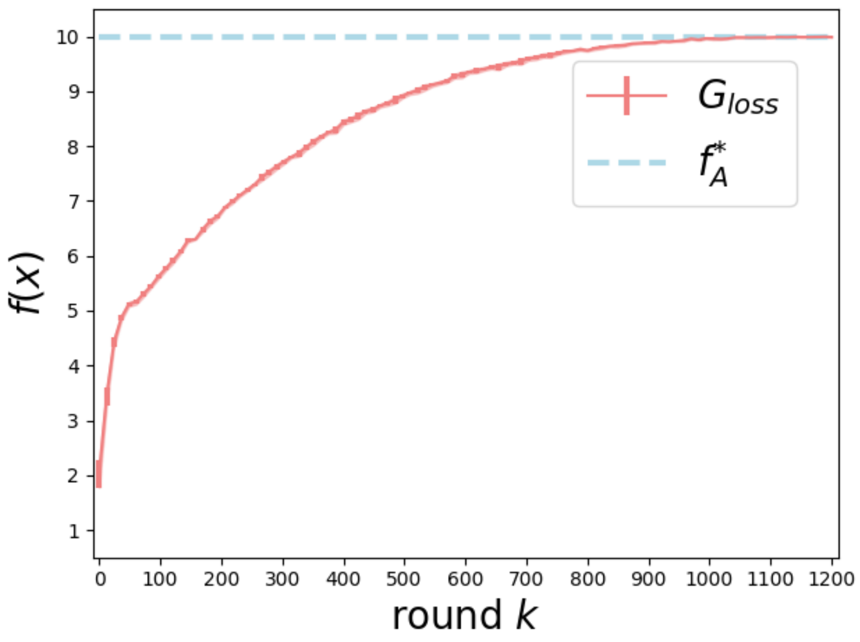
-- Q. Ying, H. Yuan, Y. Yang, M. Chen, M. Wang. "Gradient Guidance for Diffusion Models", NeurIPS 2024

Numerical Results

- Reward function $f(x) = 10 - (\theta^\top x - 3)^2$
- Ambient dimension $D = 64$; subspace dimension $d = 16$



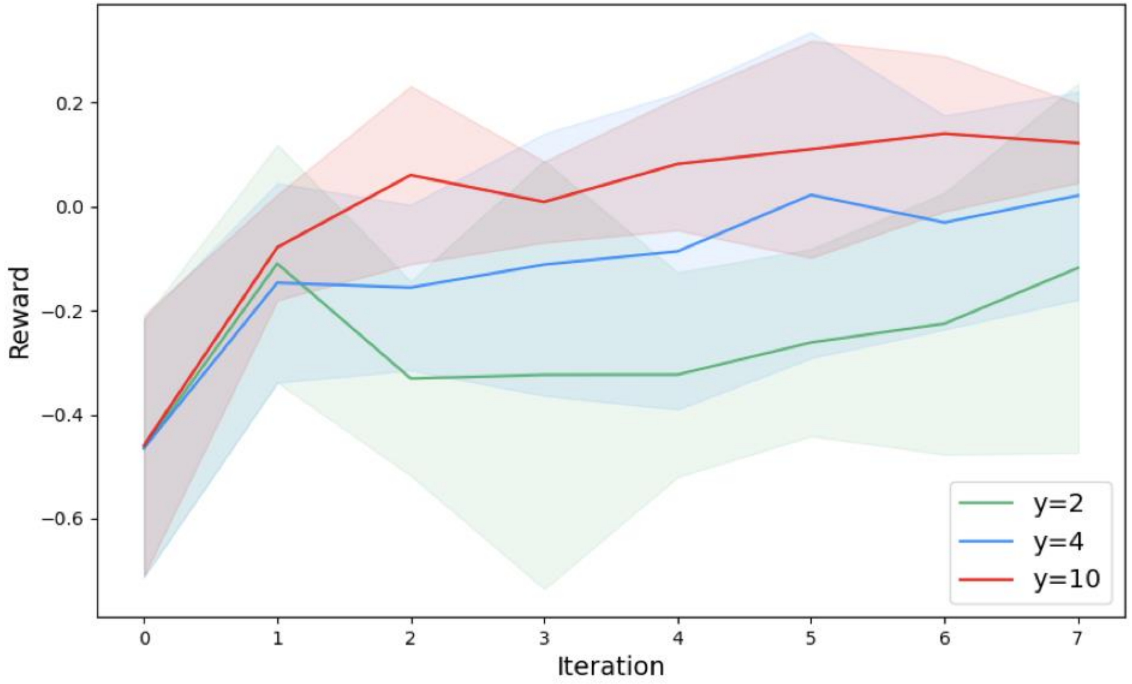
Nonadaptive:
regularized maximum

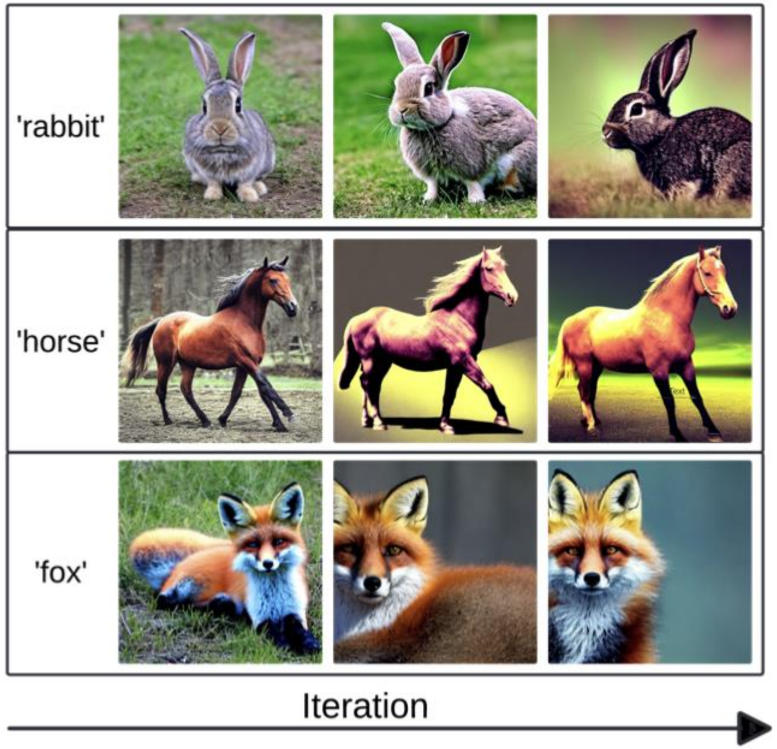


Adaptive:
global maximum

Image Generation

- Finetuing StableDiffusion v1.5 model (Rombach et al., 2022) on ImageNet





Take-Home Message and Future Directions

- We present methods for **adapting** diffusion models to an abstract reward function in both offline and online settings
- In the offline setting, diffusion models enjoy the **optimality** of off-policy bandits
- In the online setting, gradient guidance incorporates real-time feedback and enjoys **convergence** akin to first-order methods

- Beyond real-valued rewards, such as human preferences
- Noisy feedback, such as noisy reward gradients or contaminated gradients
- Nonconvex nonsmooth rewards

Thank You!