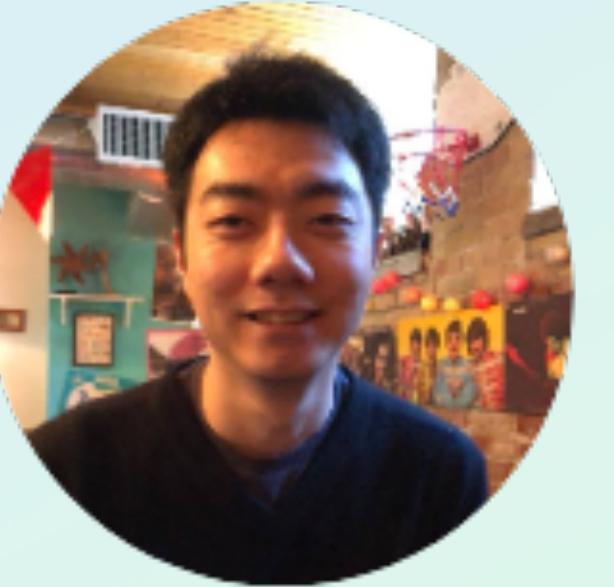


# Stochastic Control for Large Scale Reward-Driven Generative Modeling



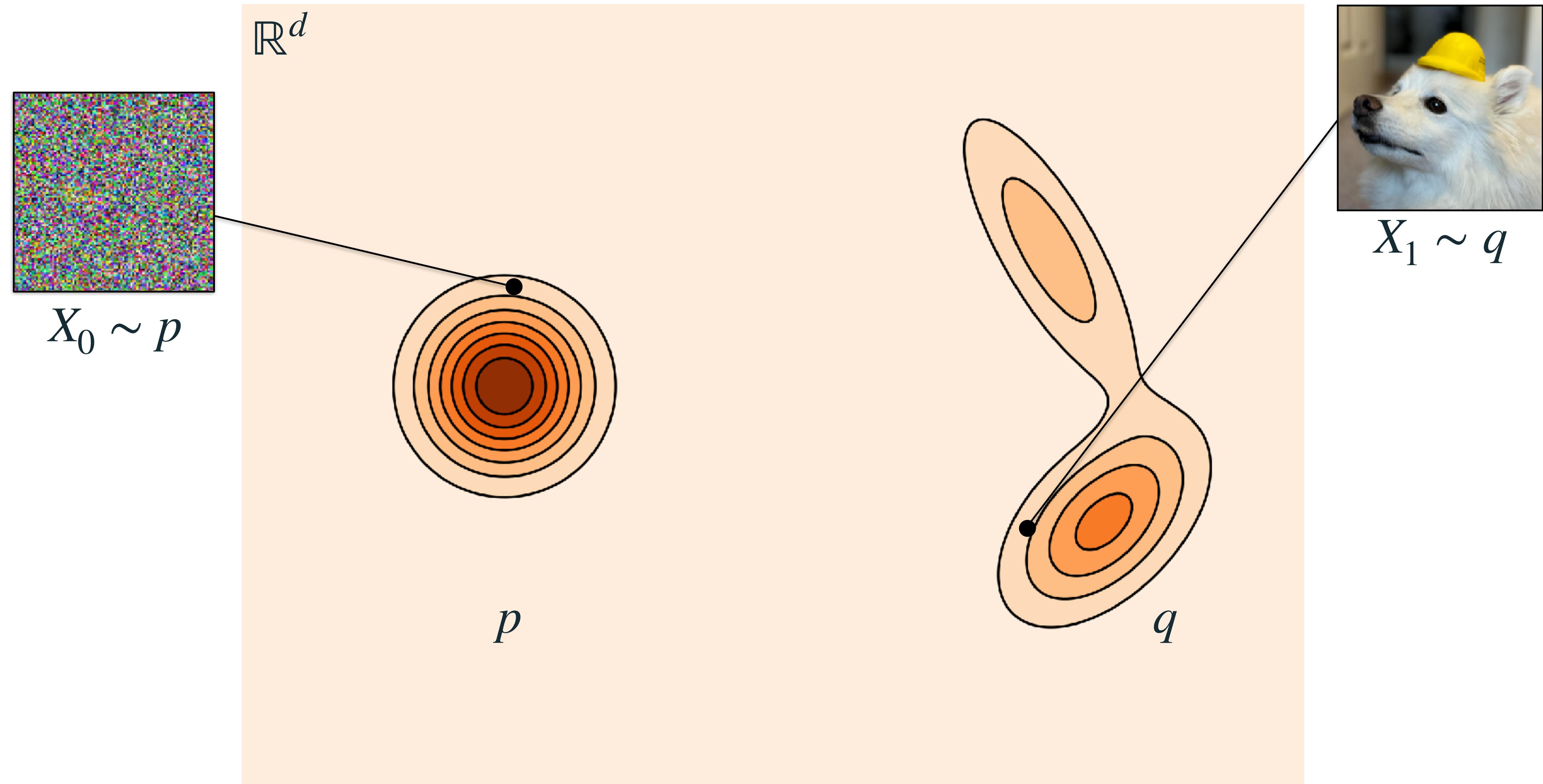
Ricky T. Q. Chen

# The Generative Modeling problem

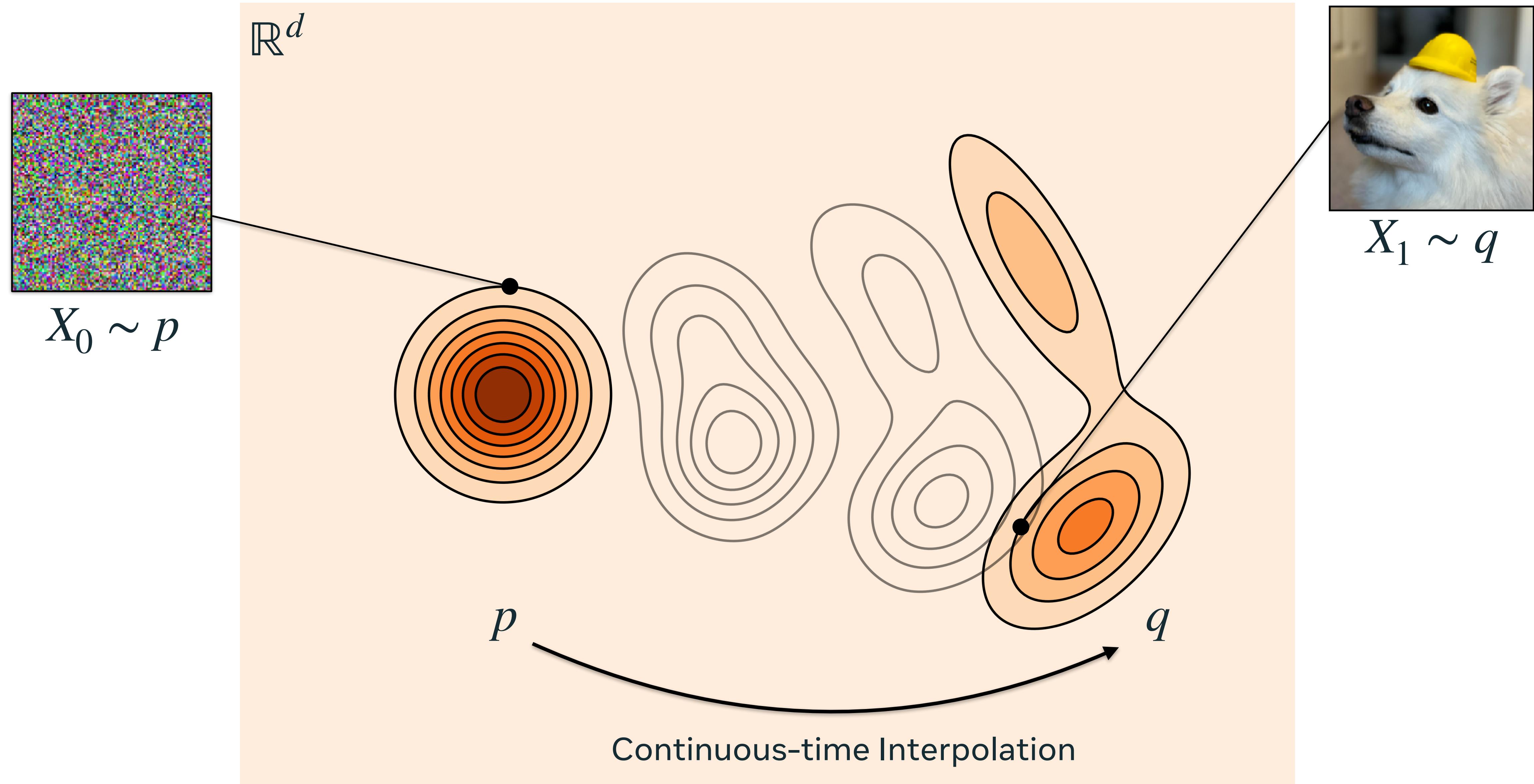
$\mathbb{R}^d$



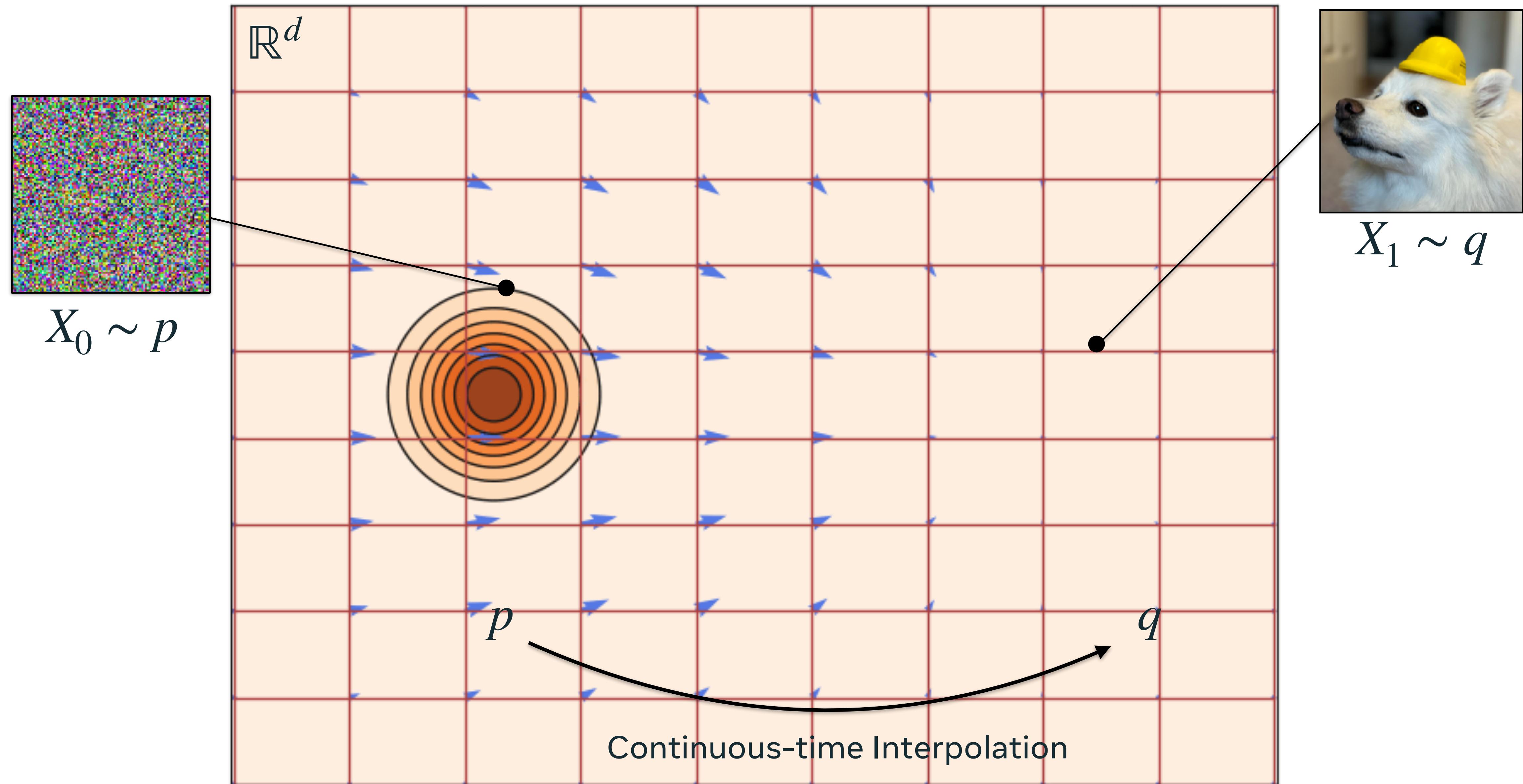
# The Generative Modeling problem



# The Generative Modeling problem



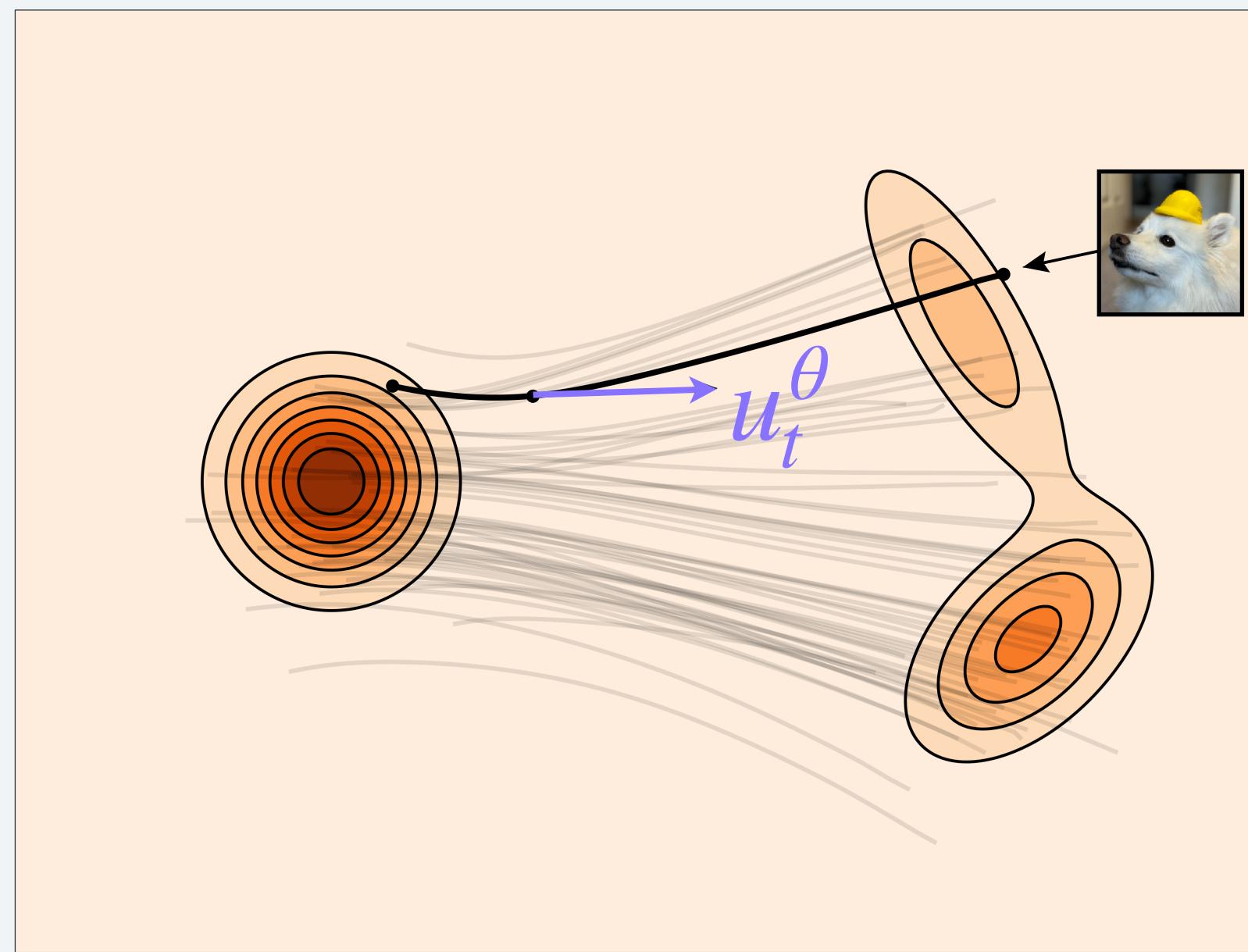
# The Generative Modeling problem



# A family of continuous-time generative models

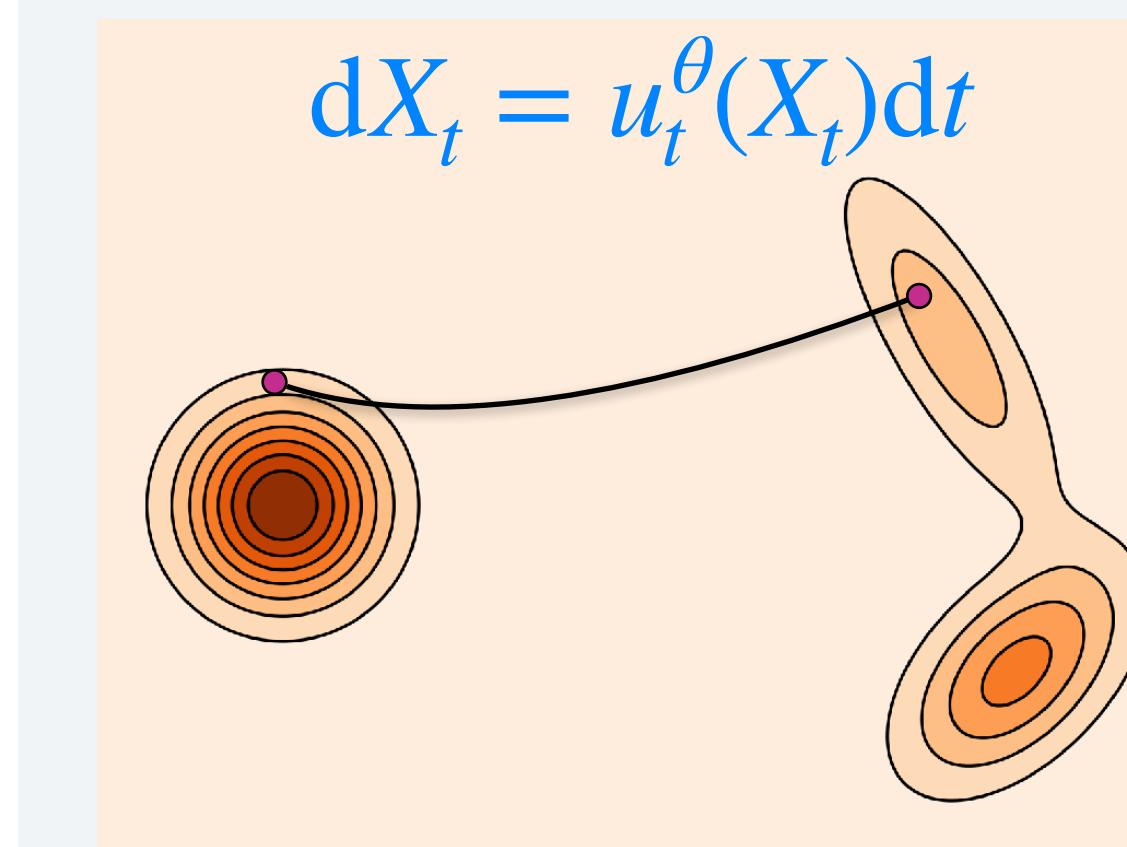
## Parameterization

A **vector field**  $u_t^\theta$  that points towards the data

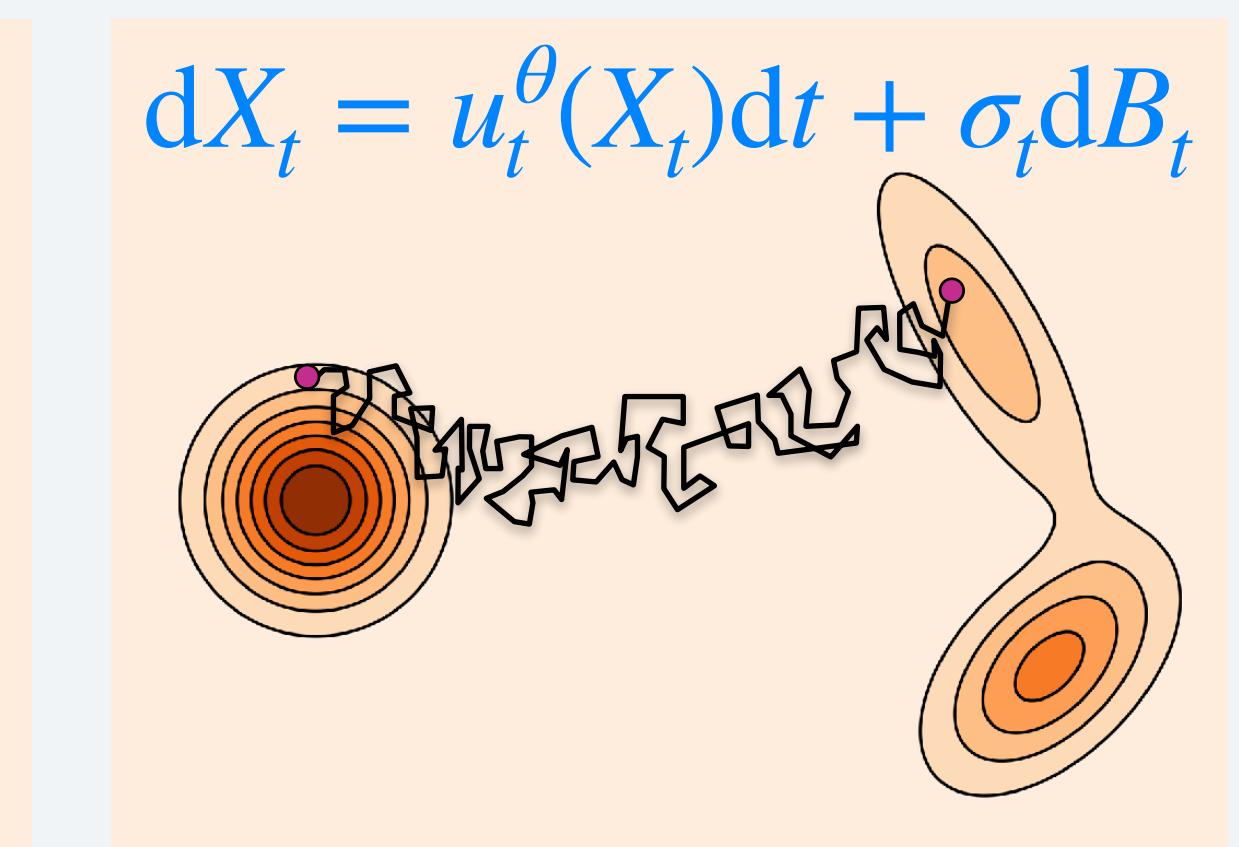


## Sampling

A **diffusion process**  $X_t$  with (optional) diffusion coefficient  $\sigma_t$



Ordinary Differential Equation



Stochastic Differential Equation

# Data-driven vs. reward-driven learning problems

## Data-driven

- fit to data
- simple & scalable

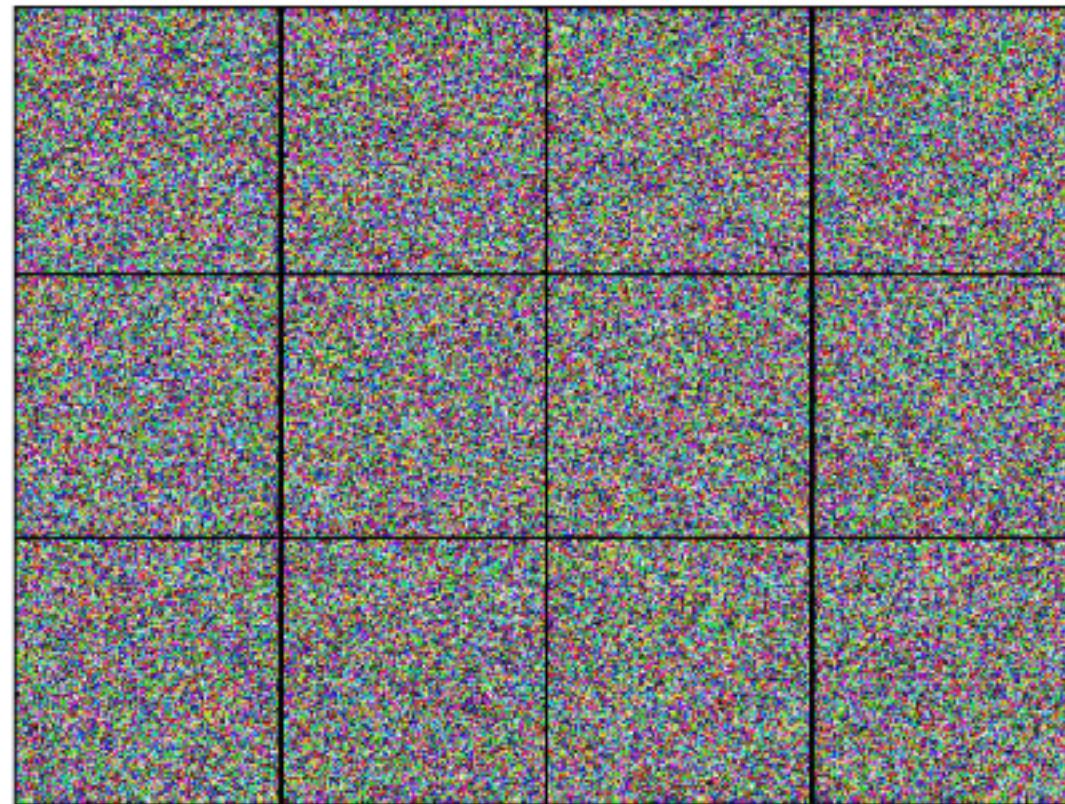


Image Generation



Meta MovieGen

# Data-driven vs. reward-driven learning problems

## Data-driven

- fit to data
- simple & scalable

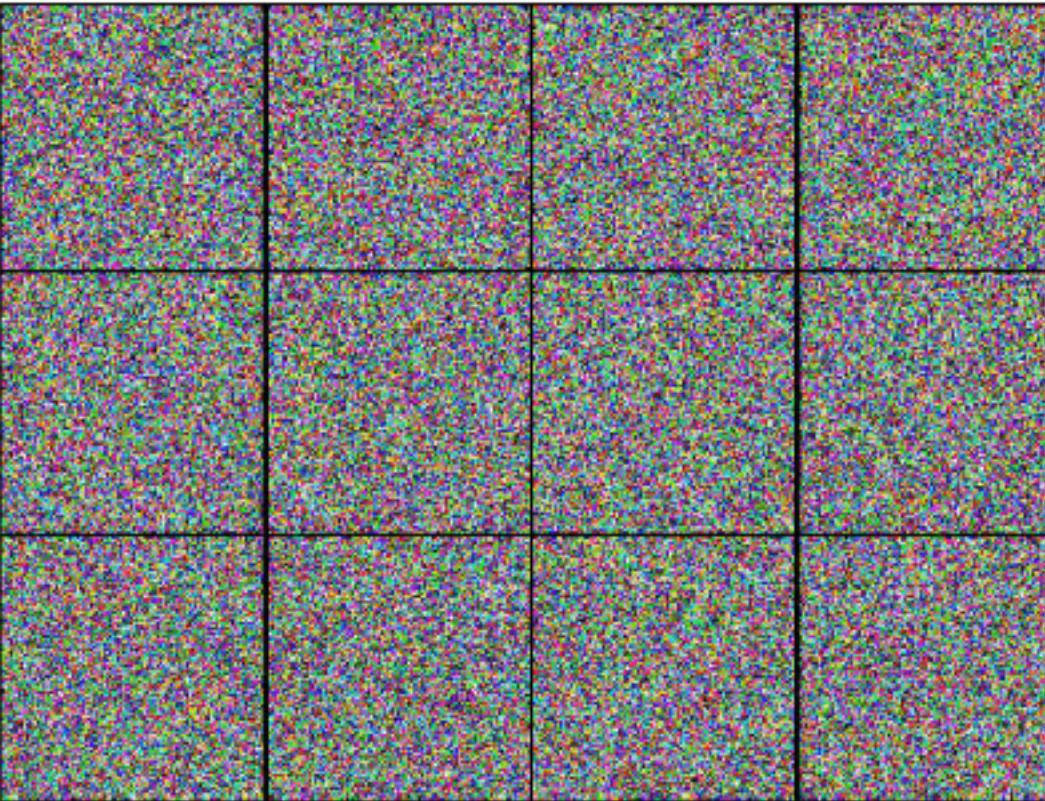


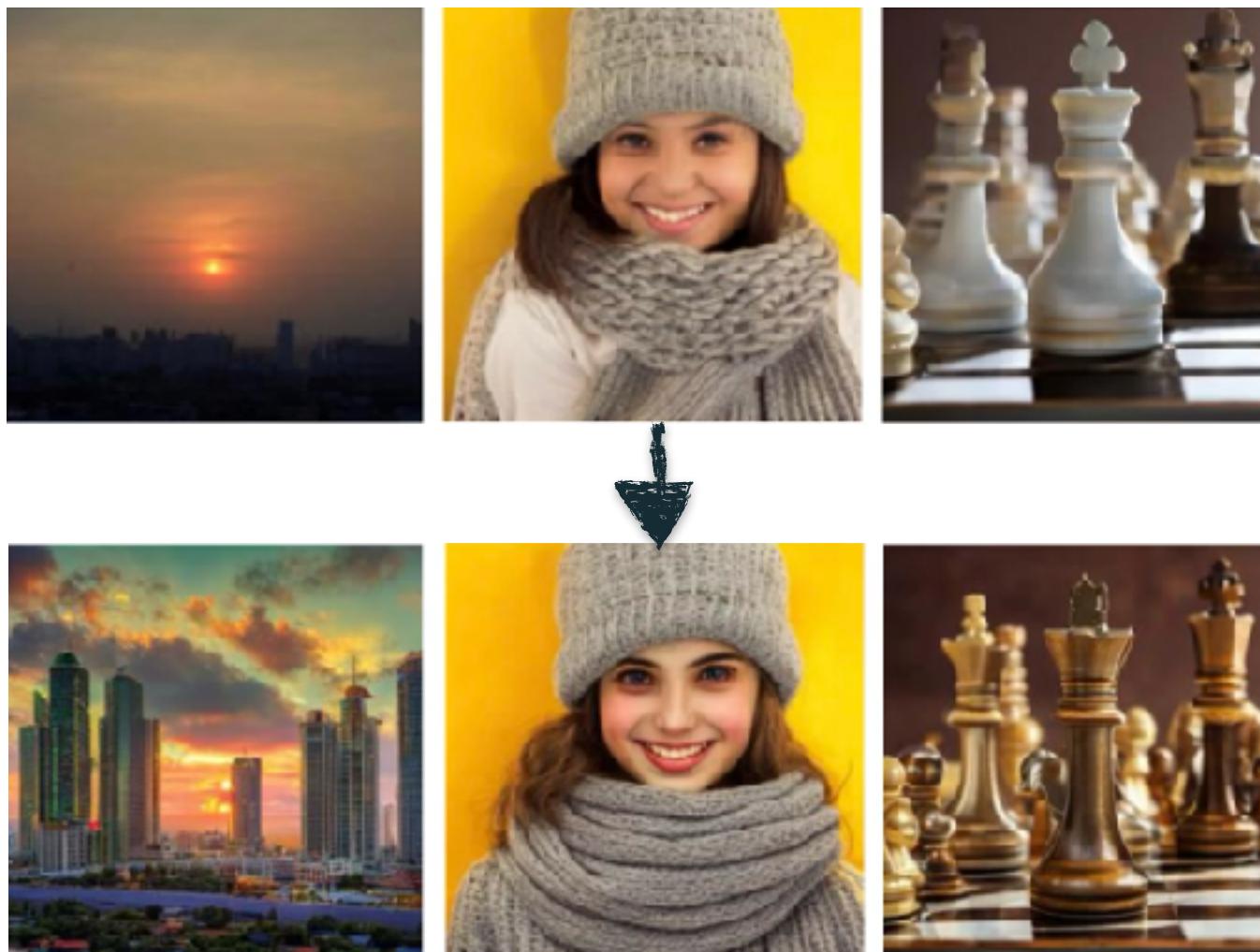
Image Generation



Meta MovieGen

## Reward-driven

- maximize reward
- no data available



Reward Fine-tuning



Low Energy Generation

I. How to formulate the problem?

II. How to solve the problem?

III. How to scale the method?

## I. How to formulate the problem?

**Stochastic Optimal Control** formulations  
for reward-driven generative modeling

# Reward-driven generative modeling

Basic setup

**Sampling from unnormalized distribution**

$$p^*(x) \propto \exp\{-E(x)\} = \exp\{r(x)\}$$

(differentiable) energy model

or

(differentiable) reward model

More generally,

**Sampling from tilted distribution**

$$p^*(x) \propto p^{\text{base}}(x) \exp\{r(x)\}$$

(sample-able) base generative model

(differentiable) reward model

# SDEs as generative models

Target distribution

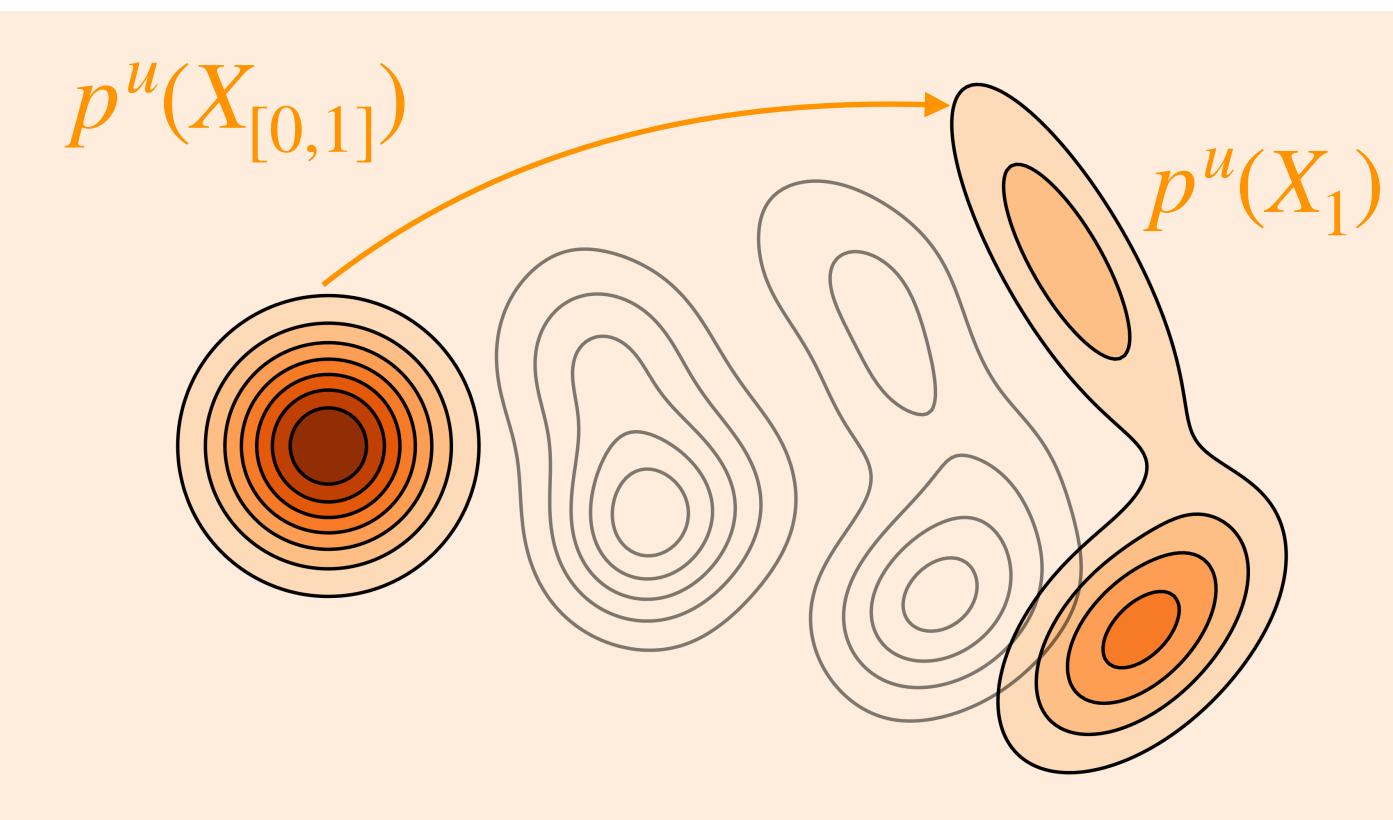
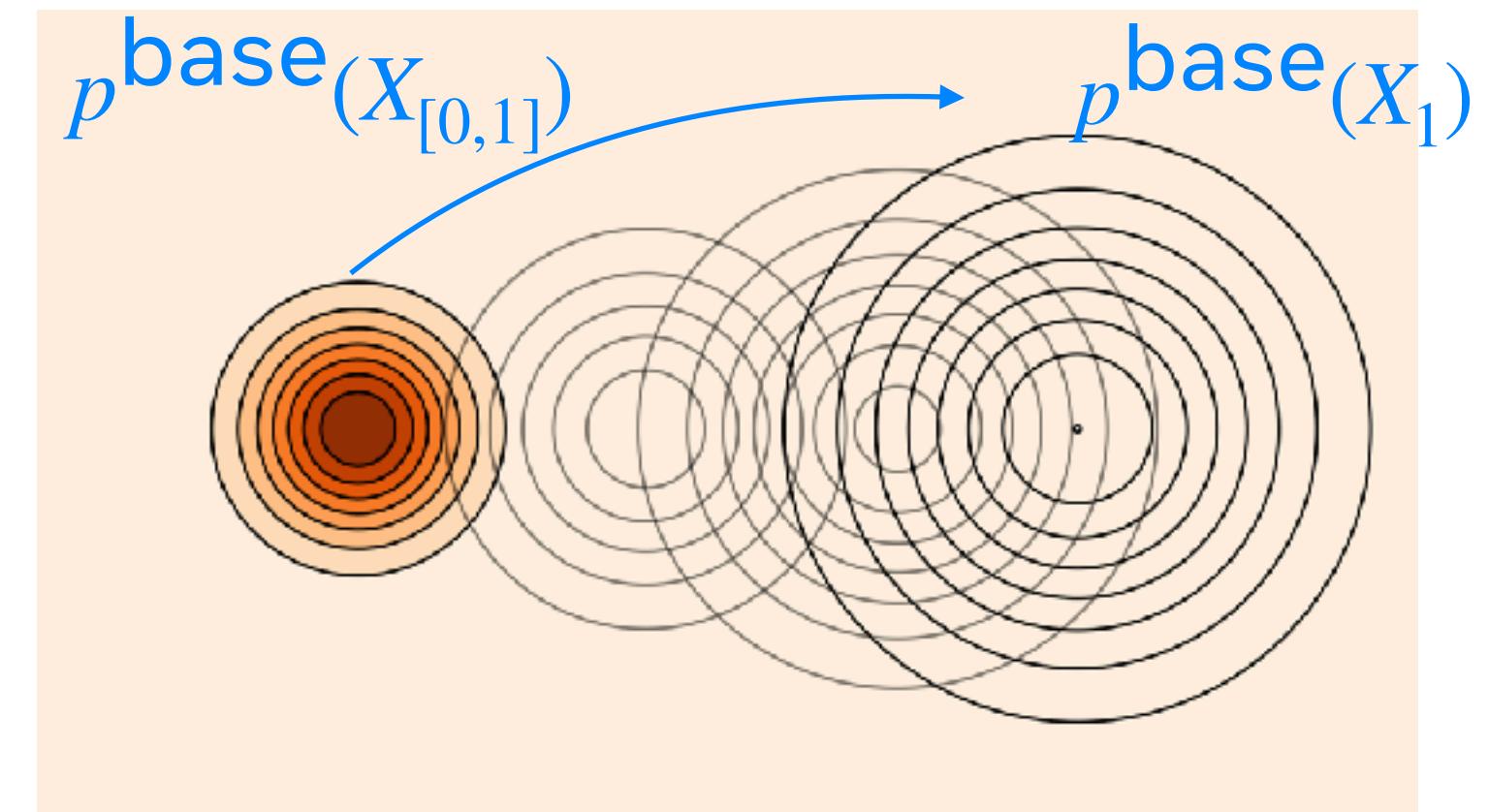
$$p^*(x) \propto p^{\text{base}}(x) \exp\{r(x)\}$$

Base generative process

$$dX_t = b_t(X_t)dt + \sigma_t dB_t \quad X_0 \sim p$$

Controlled generative process

$$dX_t = [b_t(X_t) + \sigma_t u_t^\theta(X_t)]dt + \sigma_t dB_t$$



# KL divergence and path measures

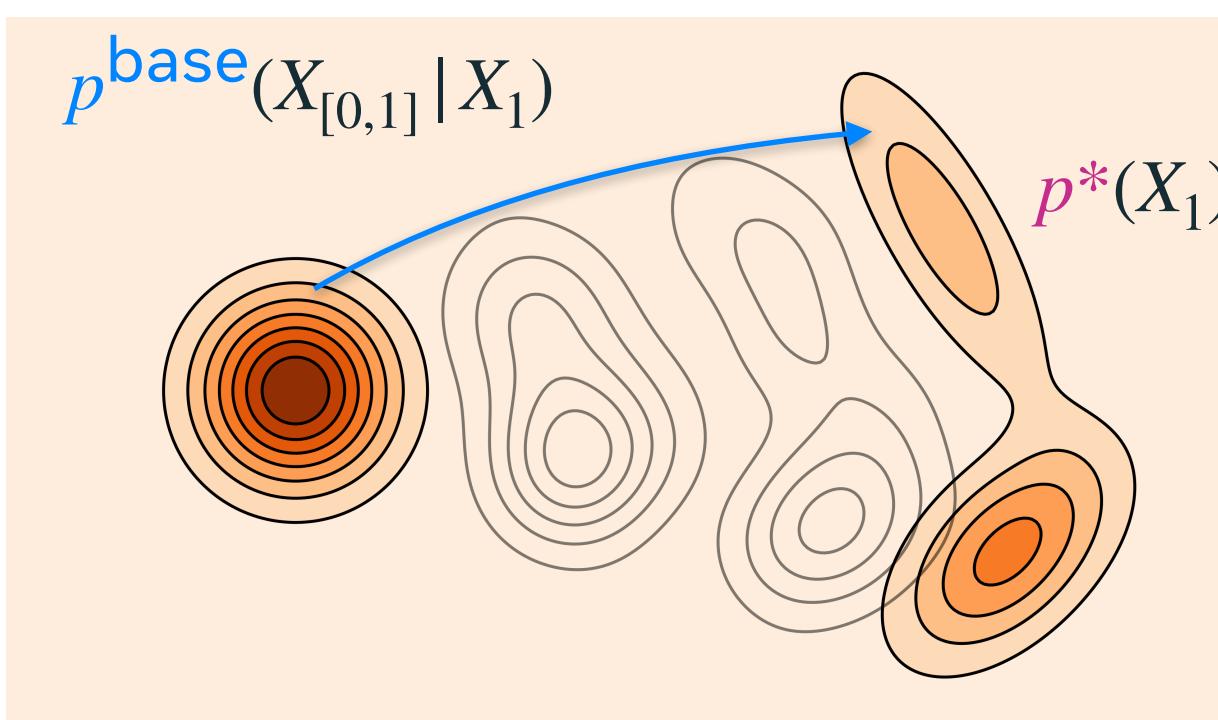
Target distribution

$$p^*(x) \propto p^{\text{base}}(x) \exp\{r(x)\}$$



Extend to stochastic process

$$p^*(X_{[0,1]}) \triangleq p^{\text{base}}(X_{[0,1]} | X_1) p^*(X_1)$$



Base generative process

$$dX_t = b_t(X_t)dt + \sigma_t dB_t \quad X_0 \sim p$$

Controlled generative process

$$dX_t = [b_t(X_t) + \sigma_t u_t^\theta(X_t)]dt + \sigma_t dB_t$$

KL divergence over stochastic process

$$\begin{aligned} & D_{\text{KL}}(p^u(X_1) \| p^*(X_1)) \\ & \leq \boxed{D_{\text{KL}}(p^u(X_{[0,1]}) \| p^*(X_{[0,1]}))} \end{aligned}$$

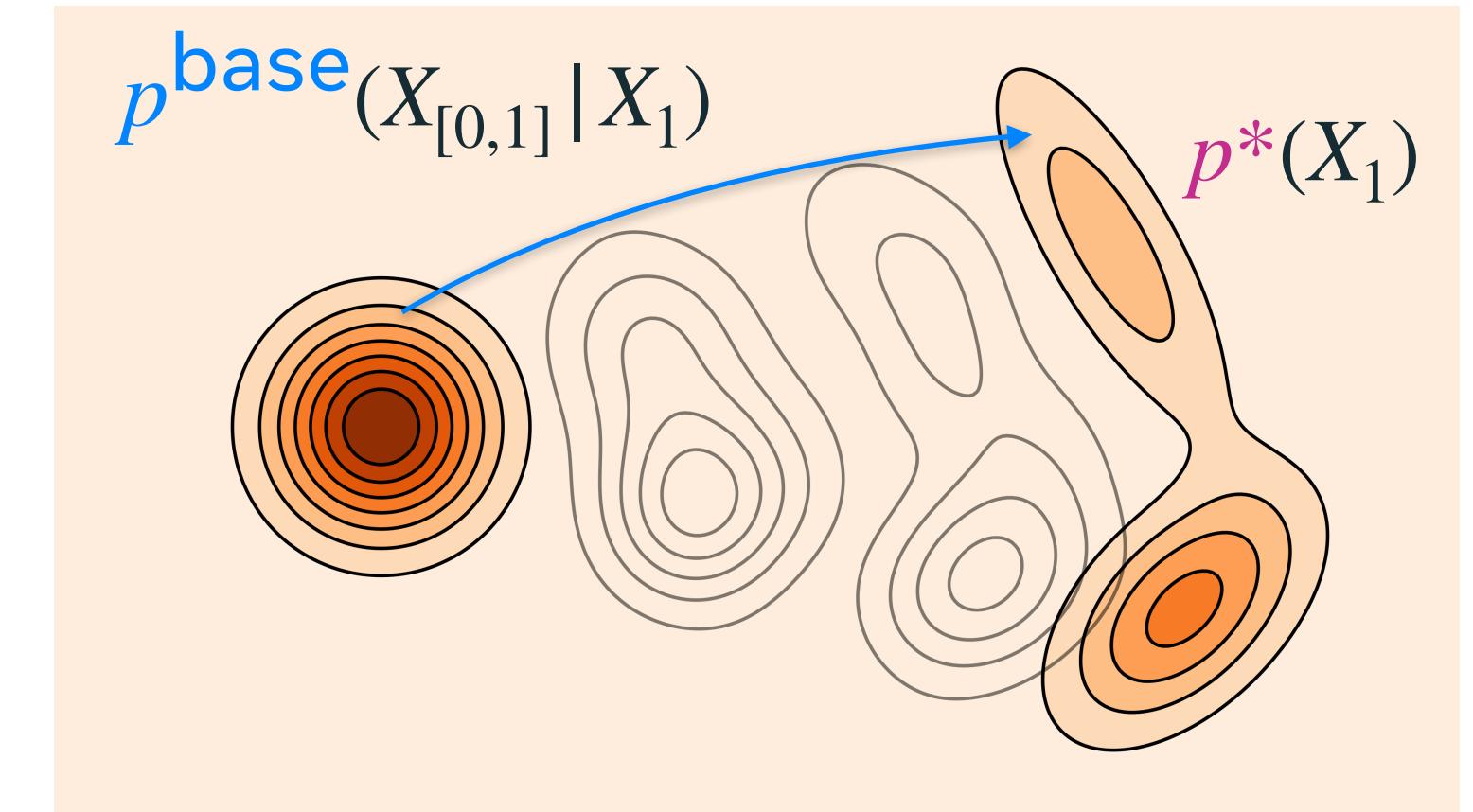
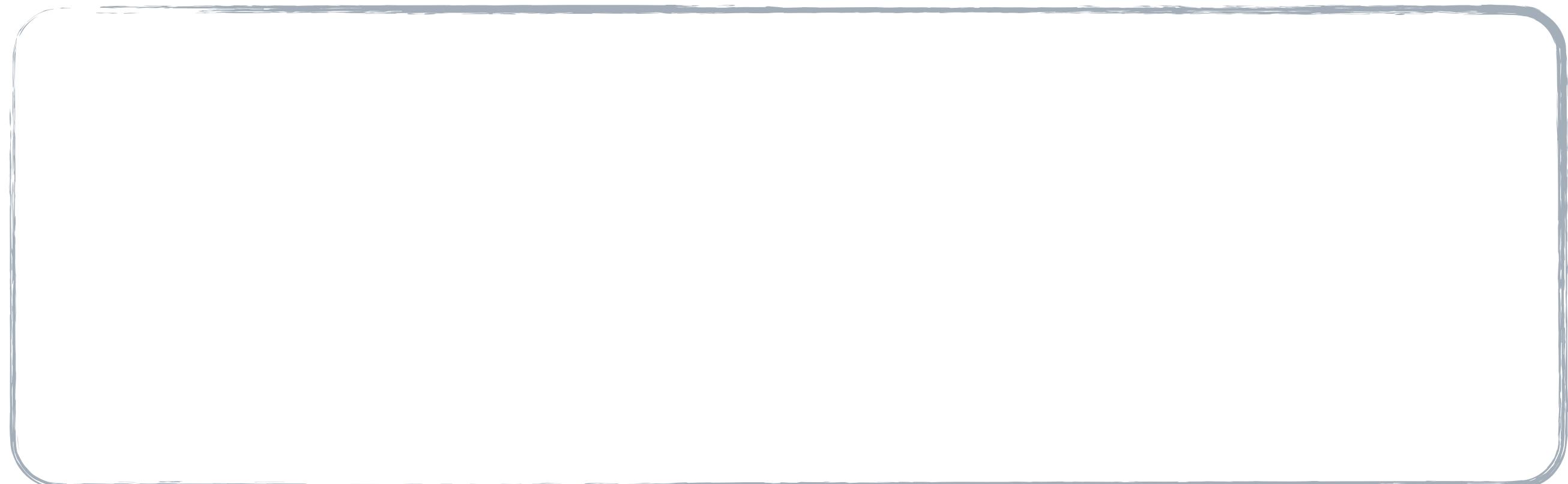
# KL divergence and path measures

Target distribution

$$p^*(x) \propto p^{\text{base}}(x) \exp\{r(x)\}$$

KL objective

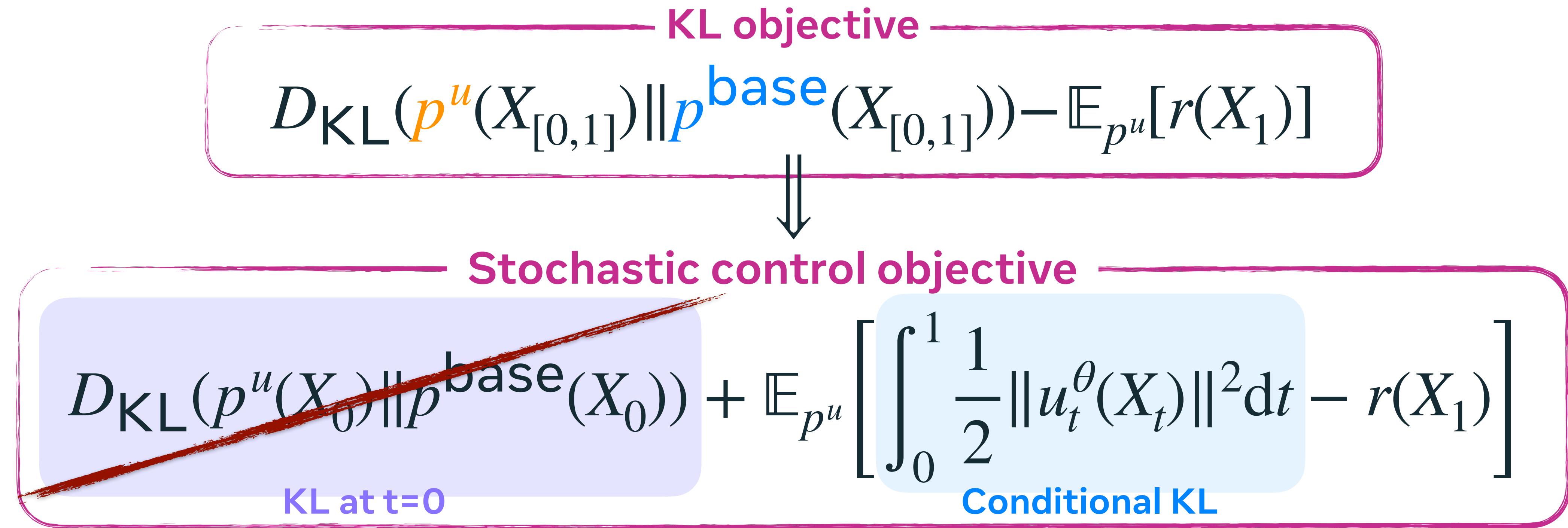
$$D_{\text{KL}}(p^u(X_{[0,1]}) \| p^*(X_{[0,1]}))$$



KL objective

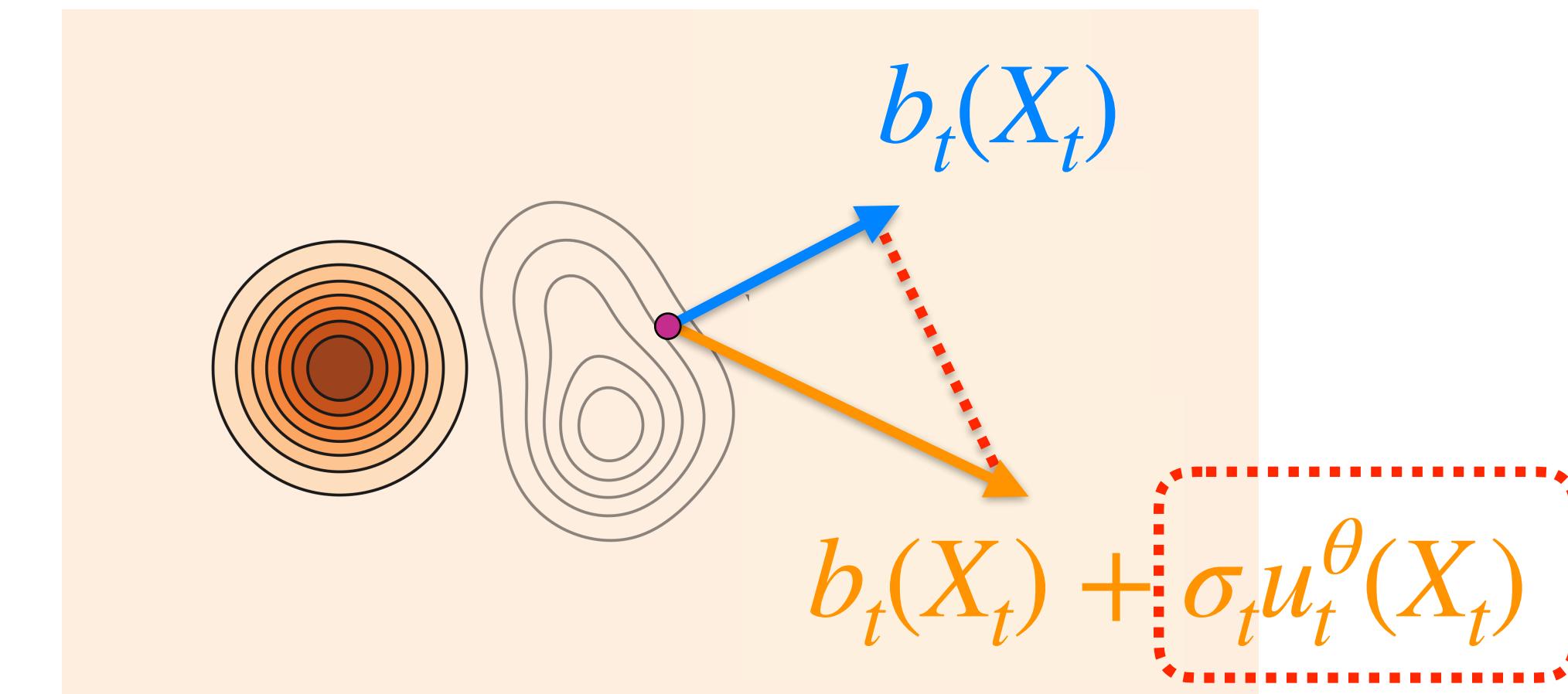
$$D_{\text{KL}}(p^u(X_{[0,1]}) \| p^{\text{base}}(X_{[0,1]})) - \mathbb{E}_{p^u}[r(X_1)]$$

# Memoryless base processes



Don't need to optimize  $p^u(X_0)$  if the base process is “**memoryless**”:

$$p^{\text{base}}(X_0, X_1) = p^{\text{base}}(X_0)p^{\text{base}}(X_1)$$



# Summary of the SOC formulation

Target distribution

$$p^*(x) \propto p^{\text{base}}(x) \exp\{r(x)\}$$

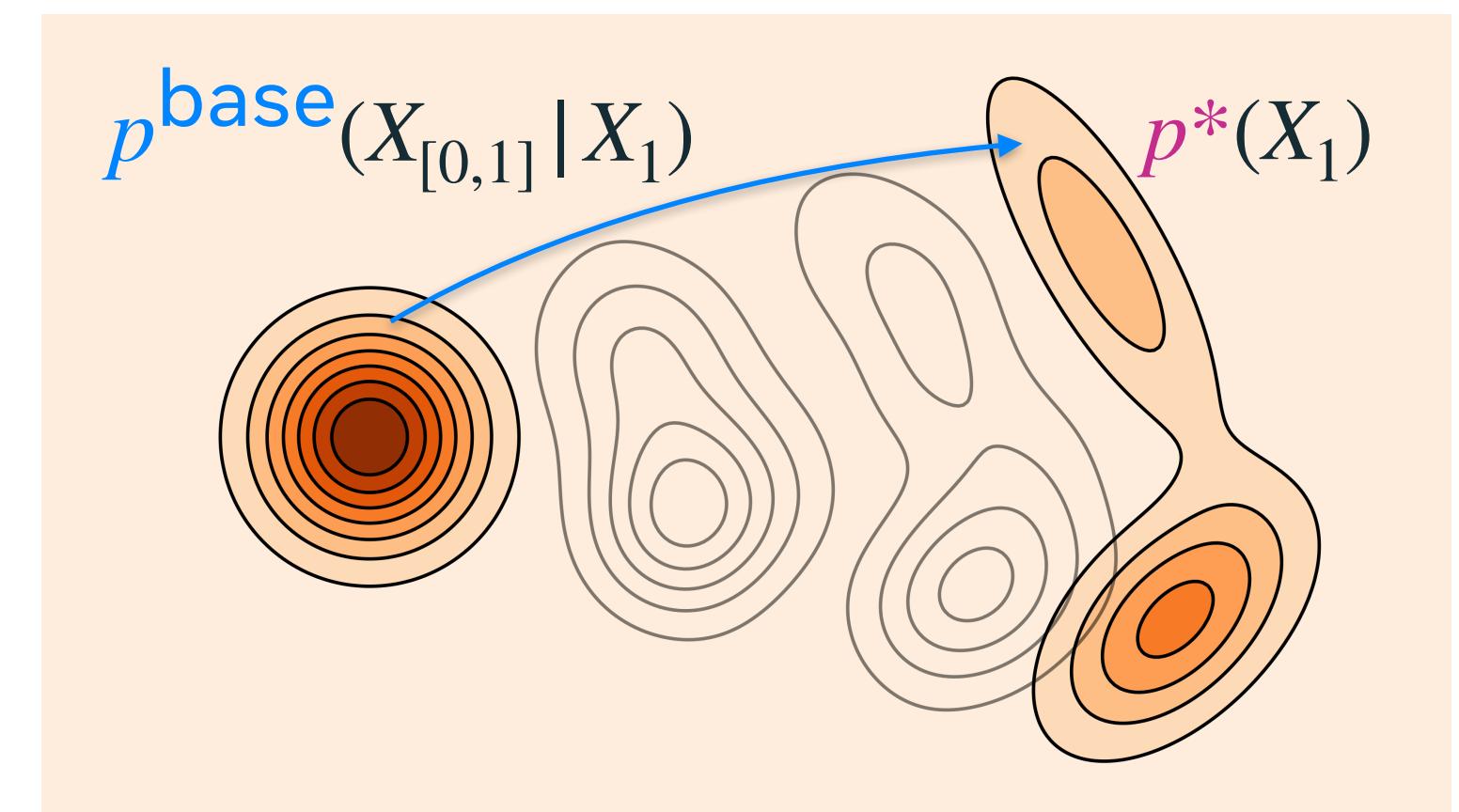
Base generative process

$$dX_t = b_t(X_t)dt + \sigma_t dB_t \quad X_0 \sim p$$

Controlled generative process

$$dX_t = [b_t(X_t) + \sigma_t u_t^\theta(X_t)]dt + \sigma_t dB_t$$

Minimize KL to



# Summary of the SOC formulation

Target distribution

$$p^*(x) \propto p^{\text{base}}(x) \exp\{r(x)\}$$

Base generative process

$$dX_t = b_t(X_t)dt + \sigma_t dB_t \quad X_0 \sim p$$

Controlled generative process

$$dX_t = [b_t(X_t) + \sigma_t u_t^\theta(X_t)]dt + \sigma_t dB_t$$

If the base process is “memoryless”:

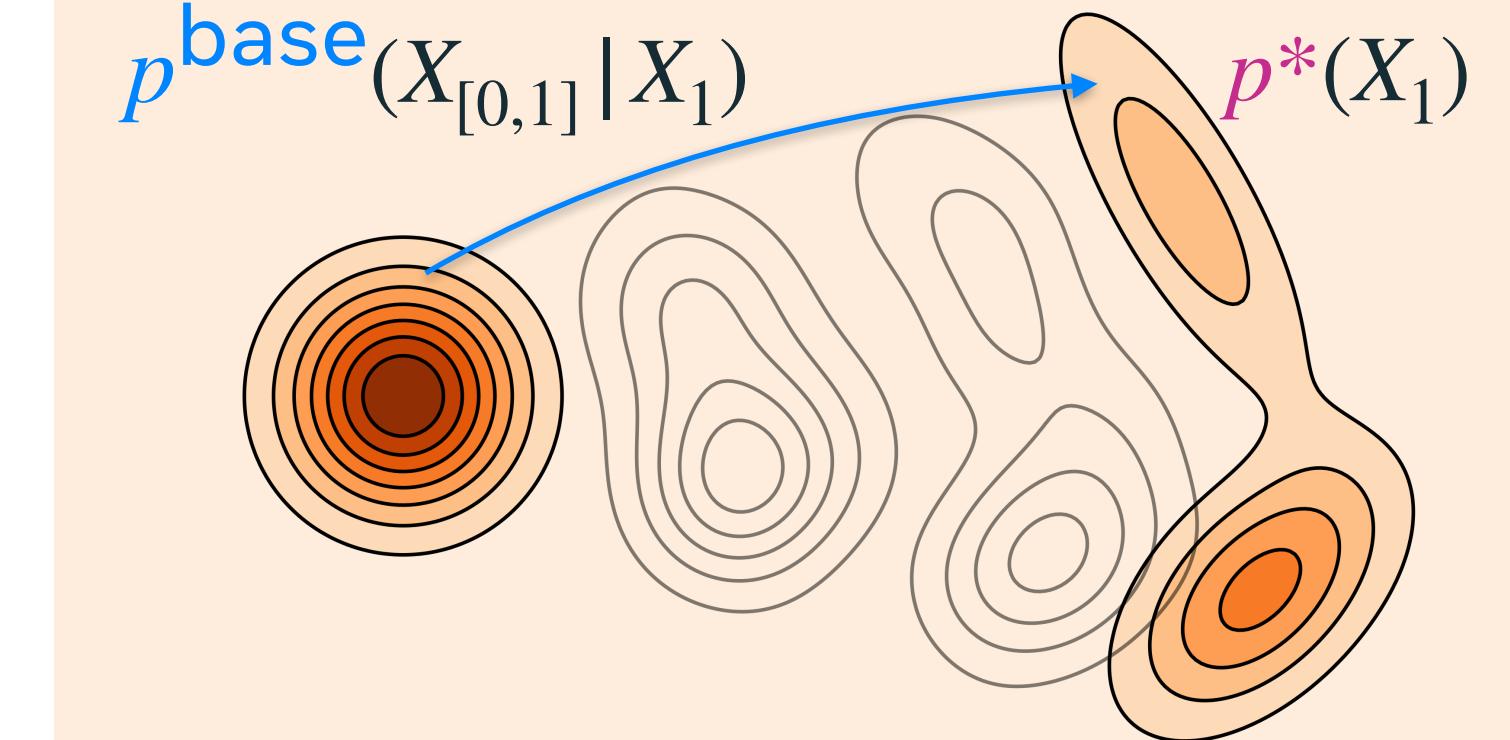
$$p^{\text{base}}(X_0, X_1) = p^{\text{base}}(X_0)p^{\text{base}}(X_1)$$

Then this is equivalent to minimizing KL:

Stochastic control objective

$$\mathbb{E}_{p^u} \left[ \int_0^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \right]$$

Minimize KL to



# A memoryless noise schedule for Flow Matching

If the base process is “memoryless”:

$$p^{\text{base}}(X_0, X_1) = p^{\text{base}}(X_0)p^{\text{base}}(X_1)$$

Flow Matching Base Model ?



Reward Fine-tuned Model



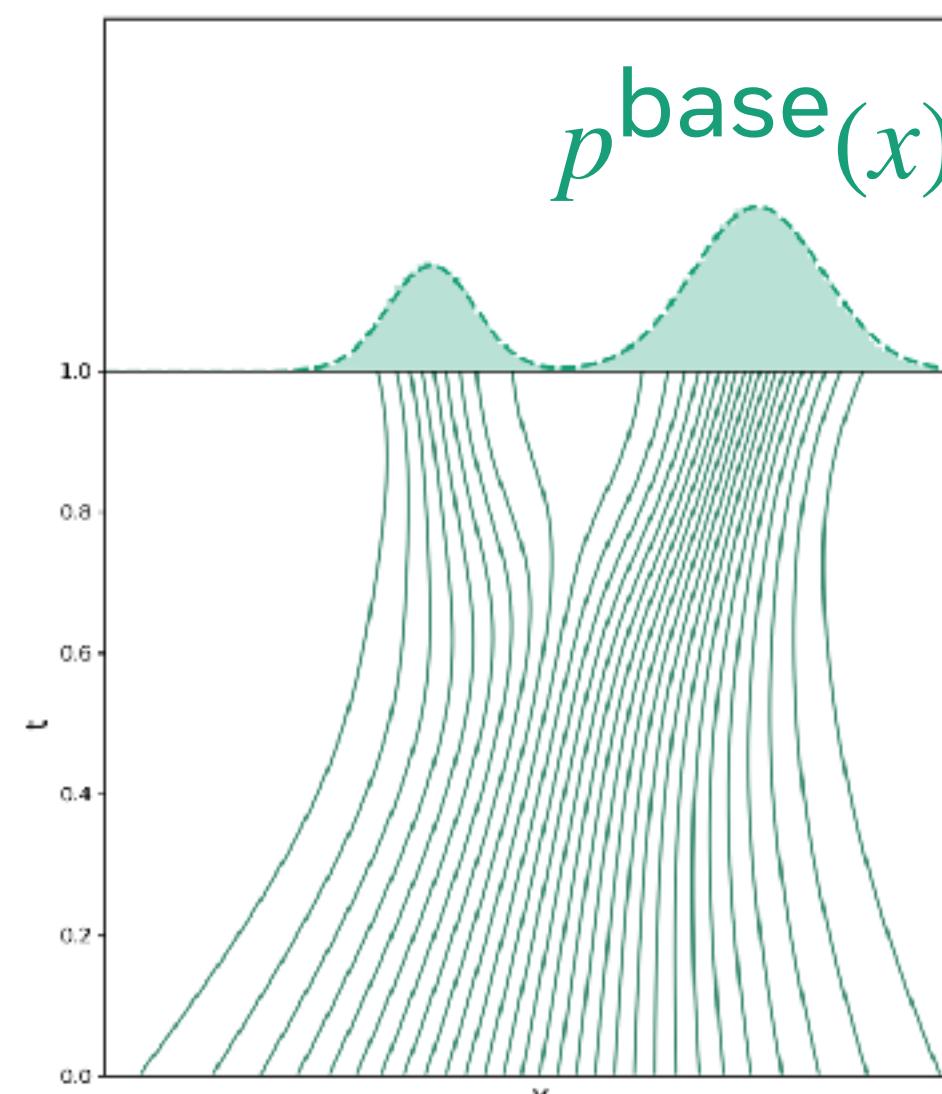
# A memoryless noise schedule for Flow Matching

If the base process is “memoryless”:

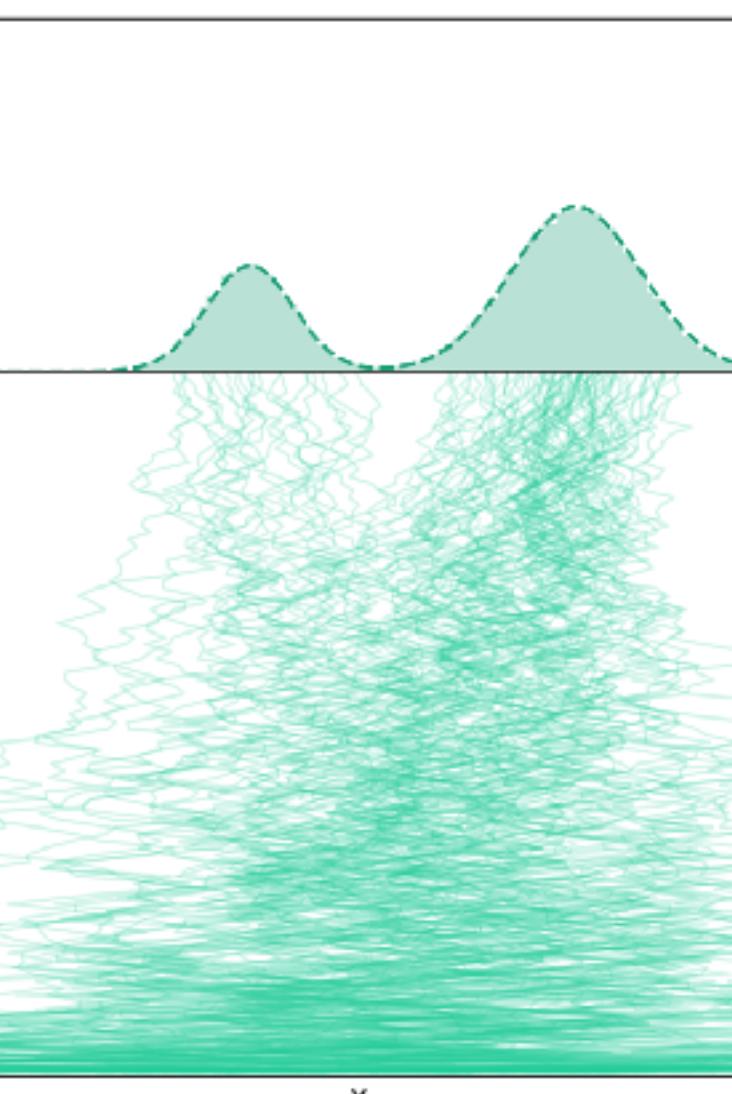
$$p^{\text{base}}(X_0, X_1) = p^{\text{base}}(X_0)p^{\text{base}}(X_1)$$

**Pretrained** Flow Matching model with linear scheduler  $X_t = (1 - t)X_0 + tX_1$ :

$$dX_t = v_t(X_t)dt$$



Convert to  
Memoryless

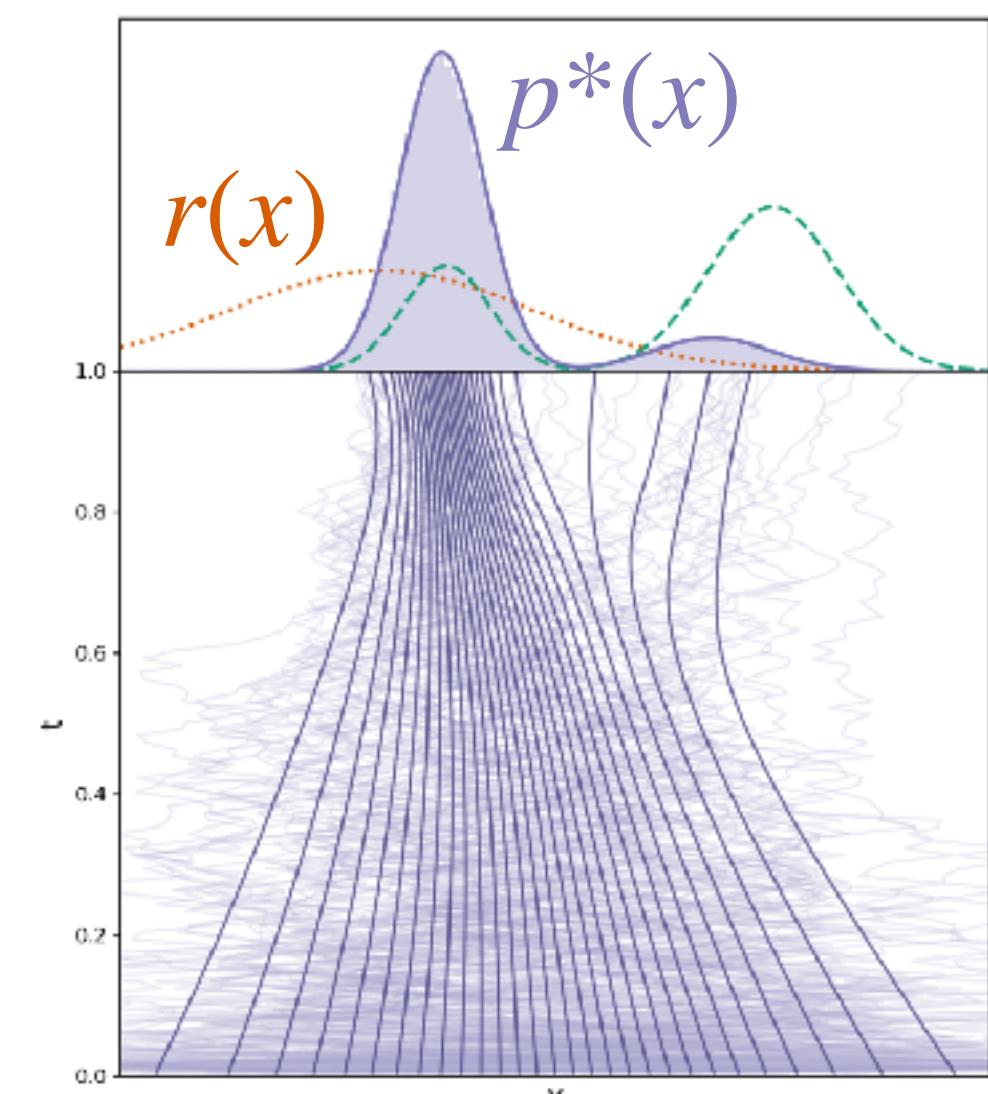


The **Memoryless** Flow Matching conversion is:

$$dX_t = b_t(X_t)dt + \sigma_t dB_t$$

$$\begin{cases} b_t(X_t) = 2v_t(X_t) - \frac{1}{t}X_t \\ \sigma_t = \frac{1-t}{t} \end{cases}$$

Fine-tune  
with SOC



# A memoryless noise schedule for Flow Matching

If the base process is “memoryless”:

$$p^{\text{base}}(X_0, X_1) = p^{\text{base}}(X_0)p^{\text{base}}(X_1)$$

Pretrained Flow Matching model with

linear scheduler  $X_t = (1 - t)X_0 + tX_1$ :

$$dX_t = v_t(X_t)dt$$

The Memoryless Flow Matching conversion is:

$$dX_t = b_t(X_t)dt + \sigma_t dB_t \quad \begin{cases} b_t(X_t) = 2v_t(X_t) - \frac{1}{t}X_t \\ \sigma_t = \frac{1-t}{t} \end{cases}$$

[!] Related to minimizing KL to the (memoryless) process used as training signal.

Refs: DDPM (Ho et al. 2020), Scalable Interpolant Transformers (Ma et al. 2024).

## II. How to solve the problem?

**Adjoint Matching:** a new approach for solving stochastic optimal control

# The most classic approach: adjoint method

## Stochastic control

$$\min_u \mathcal{L}(u) = \mathbb{E}_{p^u} \left[ \int_0^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \right]$$

s.t.  $dX_t = [b_t(X_t) + \sigma_t u_t^\theta(X_t)]dt + \sigma_t dB_t, \quad X_0 \sim p$

# The most classic approach: adjoint method

## Stochastic control

$$\begin{aligned} \min_u \mathcal{L}(u) = & \mathbb{E}_{p^u} \left[ \int_0^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \right] \\ \text{s.t. } dX_t = & [b_t(X_t) + \sigma_t u_t^\theta(X_t)] dt + \sigma_t dB_t, \quad X_0 \sim p \end{aligned}$$

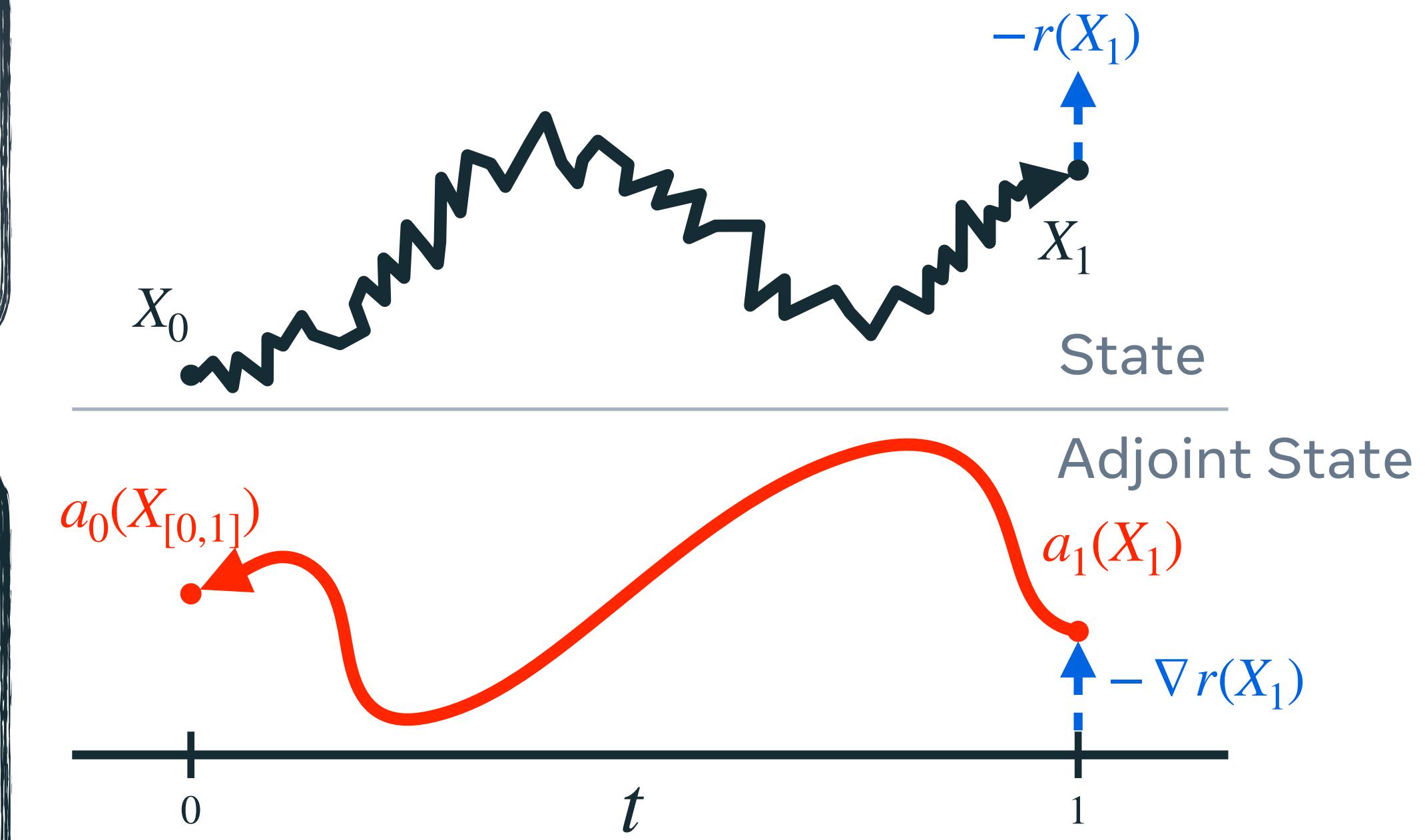
## Adjoint state

(a.k.a. the gradient w.r.t. state)

$$a_t(X_{[0,1]}) := \nabla_{X_t} \left[ \int_t^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \right]$$

## Discretization

$$\begin{aligned} X_{t_{i+1}} &= X_{t_i} + h [b_{t_i}(X_{t_i}) + \sigma_{t_i} u_{t_i}^\theta(X_{t_i})] + \sqrt{h} \sigma_t \varepsilon \\ a_{t_i} &= a_{t_{i+1}}^\top \frac{\partial X_{t_{i+1}}}{\partial X_{t_i}} = a_{t_{i+1}}^\top + h a_{t_{i+1}}^\top \nabla_{X_{t_i}} [b_{t_i}(X_{t_i}) + \sigma_{t_i} u_{t_i}^\theta(X_{t_i})] \end{aligned}$$



# The most classic approach: adjoint method

## Stochastic control

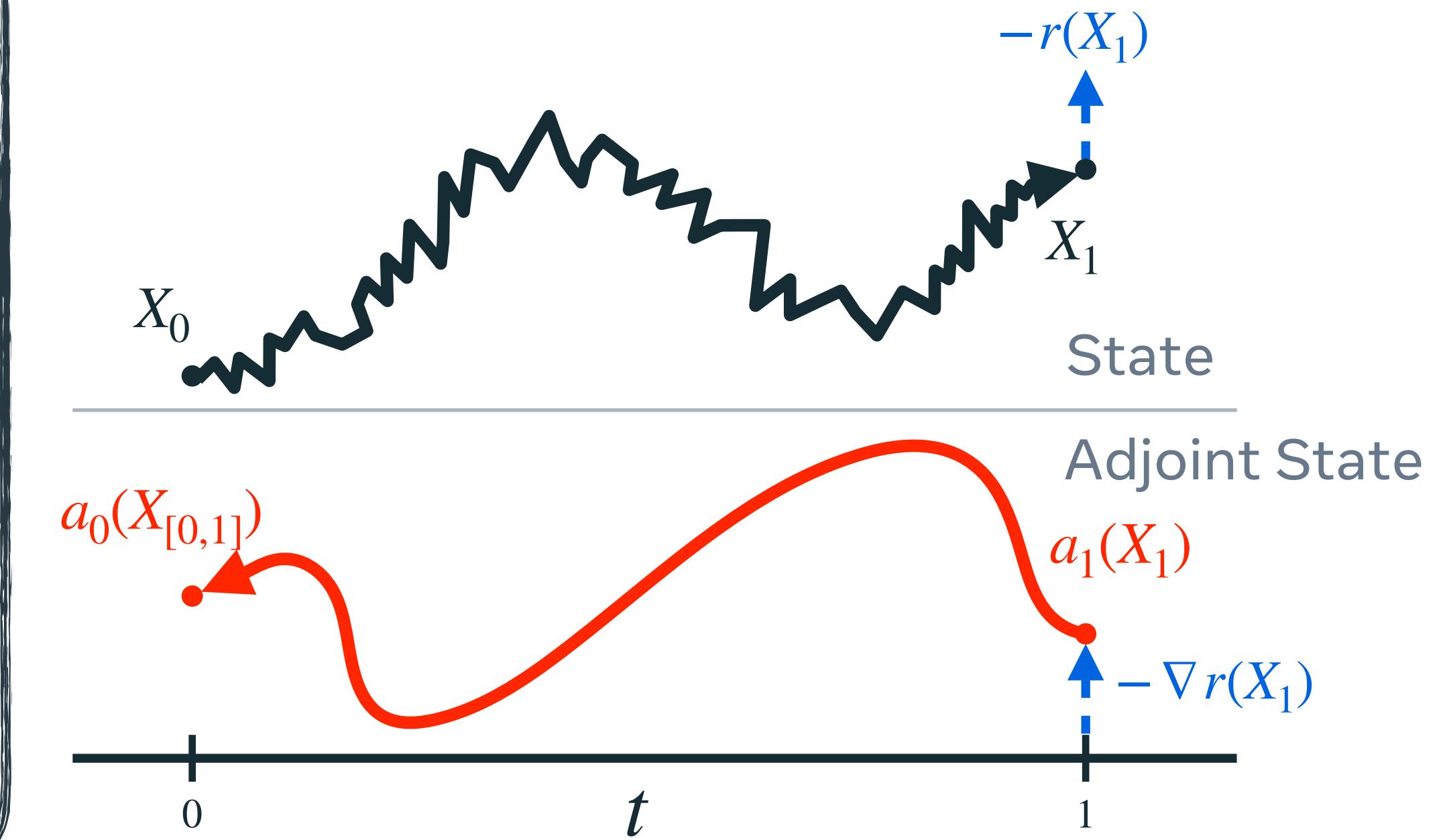
$$\begin{aligned} \min_u \mathcal{L}(u) = & \mathbb{E}_{p^u} \left[ \int_0^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \right] \\ \text{s.t. } dX_t = & [b_t(X_t) + \sigma_t u_t^\theta(X_t)] dt + \sigma_t dB_t, \quad X_0 \sim p \end{aligned}$$

## Adjoint state

(a.k.a. the gradient w.r.t. state)

$$\begin{aligned} a_t(X_{[0,1]}) &:= \nabla_{X_t} \left[ \int_t^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \right] \\ \frac{d}{dt} a_t(X_{[0,1]}) &= -a_t(X_{[0,1]})^\top \nabla_{X_t} (b_t(X_t) + \sigma_t u_t^\theta(X_t)) \end{aligned}$$

$$a_1(X_{[0,1]}) = -\nabla r(X_1)$$



# The most classic approach: adjoint method

## Stochastic control

$$\begin{aligned} \min_u \mathcal{L}(u) = & \mathbb{E}_{p^u} \left[ \int_0^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \right] \\ \text{s.t. } dX_t = & [b_t(X_t) + \sigma_t u_t^\theta(X_t)] dt + \sigma_t dB_t, \quad X_0 \sim p \end{aligned}$$

## Adjoint state

(a.k.a. the gradient w.r.t. state)

$$a_t(X_{[0,1]}) := \nabla_{X_t} \left[ \int_t^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \right]$$

$$\frac{d}{dt} a_t(X_{[0,1]}) = -a_t(X_{[0,1]})^\top \nabla_{X_t} (b_t(X_t) + \sigma_t u_t^\theta(X_t)) - \nabla_{X_t} \left( \frac{1}{2} \|u_t^\theta(X_t)\|^2 \right)$$

$$a_1(X_{[0,1]}) = -\nabla r(X_1)$$

$$\frac{d\mathcal{L}}{d\theta} = \int_0^1 \frac{\partial u_t^\theta(X_t)}{\partial \theta}^\top \sigma_t a_t(X_{[0,1]}) + \frac{\partial}{\partial \theta} \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt$$

Aggregate ↴

# An alternative perspective: fixed point iteration

## Cost functional

(a.k.a. expected future cost)

$$J(u; t, x) := \mathbb{E}_{p^u} \left[ \int_t^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \mid X_t = x \right]$$

## Optimality criterion

(a.k.a. steepest descent)

$$u_t^*(x) = -\sigma_t \nabla_x J(u^*; t, x)$$

## Relation to adjoint state

$$\nabla_x J(u^*; t, x) = \mathbb{E}_{p^u} [a_t(X_{[0,1]}) \mid X_t = x]$$

## “Basic” Adjoint Matching

$$\min_u \mathcal{L}(u) = \int_0^1 \|u_t^\theta(X_t) + \sigma_t \nabla_x J(u^*; t, X_t)\|^2 dt$$

$$\begin{aligned} X_{[0,1]} &\sim p^{\bar{u}} \\ \bar{u} &= \text{stopgrad}(u) \end{aligned}$$

# An alternative perspective: fixed point iteration

## Cost functional

(a.k.a. expected future cost)

$$J(u; t, x) := \mathbb{E}_{p^u} \left[ \int_t^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \mid X_t = x \right]$$

## Optimality criterion

(a.k.a. steepest descent)

$$u_t^*(x) = -\sigma_t \nabla_x J(u^*; t, x)$$

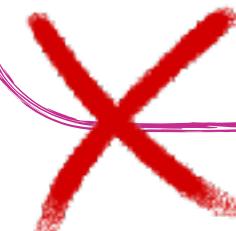
## Relation to adjoint state

$$\nabla_x J(u^*; t, x) = \mathbb{E}_{p^u} [a_t(X_{[0,1]}) \mid X_t = x]$$

## “Basic” Adjoint Matching

$$\min_u \mathcal{L}(u) = \int_0^1 \|u_t^\theta(X_t) + \sigma_t a_t(X_{[0,1]})\|^2 dt$$

$$X_{[0,1]} \sim p^{\bar{u}} \\ \bar{u} = \text{stopgrad}(u)$$



Exactly equivalent to adjoint method:  $\frac{d\mathcal{L}}{d\theta} = \int_0^1 \frac{\partial u_t^\theta(X_t)}{\partial \theta}^\top \sigma_t a_t(X_{[0,1]}) + \frac{\partial}{\partial \theta} \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt$

# Adjoint Matching and the *lean* adjoint state

Adjoint state

$$\frac{d}{dt} \mathbf{a}_t(\mathbf{X}_{[0,1]}) = -\mathbf{a}_t(\mathbf{X}_{[0,1]})^\top \nabla_{\mathbf{X}_t} (\sigma_t u_t^\theta(\mathbf{X}_t) + b_t(\mathbf{X}_t)) - \nabla_{\mathbf{X}_t} \left( \frac{1}{2} \|u_t^\theta(\mathbf{X}_t)\|^2 \right)$$

At optimum

$$u_t^*(x) = -\sigma_t \nabla_x J(u^*; t, x)$$

$$u_t^*(x) = \mathbb{E}_{p^u} [-\sigma_t \mathbf{a}_t(\mathbf{X}_{[0,1]}) | X_t = x]$$

$$\mathbb{E}_{p^u} [u_t^*(X_t) + \sigma_t \mathbf{a}_t(\mathbf{X}_{[0,1]})] = 0$$

$$\mathbb{E}_{p^u} [u_t^*(X_t)^\top \nabla_{X_t} u_t^*(X_t) + \sigma_t \mathbf{a}_t(\mathbf{X}_{[0,1]})^\top \nabla_{X_t} u_t^*(X_t)] = 0$$

# Adjoint Matching and the *lean* adjoint state

## Adjoint state

$$\frac{d}{dt} \tilde{a}_t(X_{[0,1]}) = -\tilde{a}_t(X_{[0,1]})^\top \nabla_{X_t} (\sigma_t u_t^\theta(X_t) + b_t(X_t)) - \nabla_{X_t} \left( \frac{1}{2} \|u_t^\theta(X_t)\|^2 \right)$$

## \*Adjoint Matching\*

### “Lean” adjoint state

$$\frac{d}{dt} \tilde{a}_t(X_{[0,1]}) = -\tilde{a}_t(X_{[0,1]})^\top \nabla_{X_t} b_t(X_t)$$

$$\tilde{a}_1(X_{[0,1]}) = -\nabla r(X_1)$$

$$\min_u \mathcal{L}(u) = \int_0^1 \|u_t^\theta(X_t) + \sigma_t \tilde{a}_t(X_{[0,1]})\|^2 dt \quad X_{[0,1]} \sim p^{\bar{u}}$$

$\bar{u} = \text{stopgrad}(u)$



Fixed point  $\iff$  optimal control

$$\frac{\delta \mathcal{L}(u)}{\delta u} = 0 \iff u = u^*$$

# Adjoint Matching for reward fine-tuning

Target distribution

$$p^*(x) \propto p^{\text{base}}(x) \exp\{r(x)\}$$



Flow  
(Pre-trained)

Flow Matching

$$dX_t = v_t(X_t)dt$$

$$X_0 \sim p$$

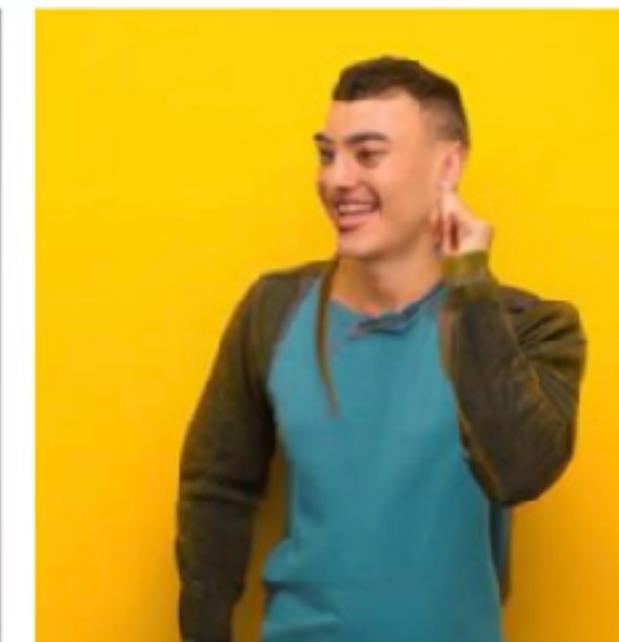
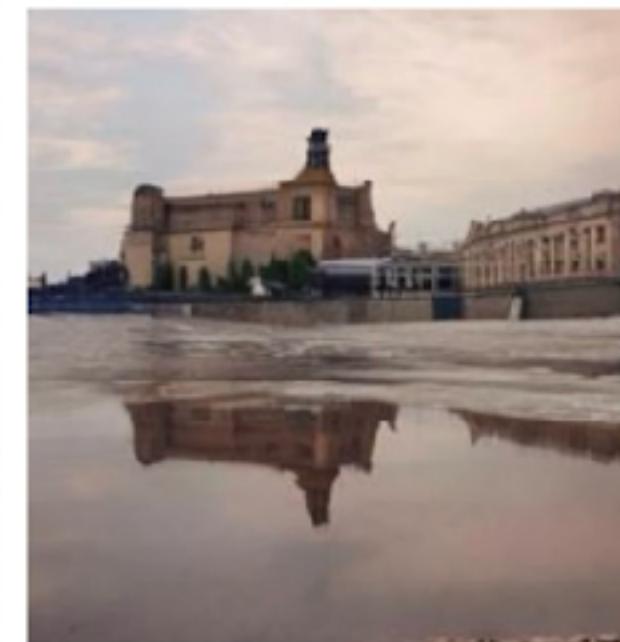
# Adjoint Matching for reward fine-tuning

Target distribution

$$p^*(x) \propto p^{\text{base}}(x) \exp\{r(x)\}$$

Memoryless Flow Matching

$$dX_t = b_t(X_t)dt + \sigma_t dB_t \quad X_0 \sim p$$



Flow  
(Pre-trained)



Memoryless  
(Pre-trained)

# Adjoint Matching for reward fine-tuning

Stochastic control

$$\begin{aligned} \min_u \mathcal{L}(u) = & \mathbb{E}_{p^u} \left[ \int_0^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt - r(X_1) \right] \\ \text{s.t. } dX_t = & [b_t(X_t) + \sigma_t u_t^\theta(X_t)] dt + \sigma_t dB_t, \quad X_0 \sim p \end{aligned}$$



Memoryless  
(Pre-trained)



Memoryless  
(Fine-tuned)



# Adjoint Matching for reward fine-tuning

Memoryless Flow Matching

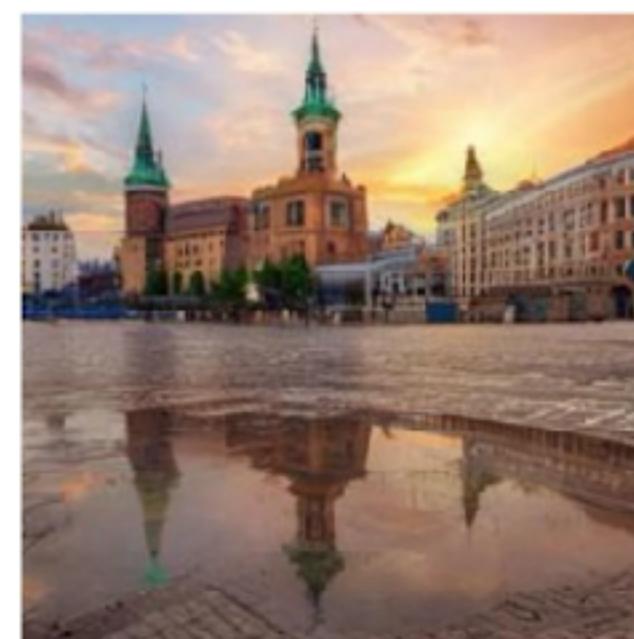
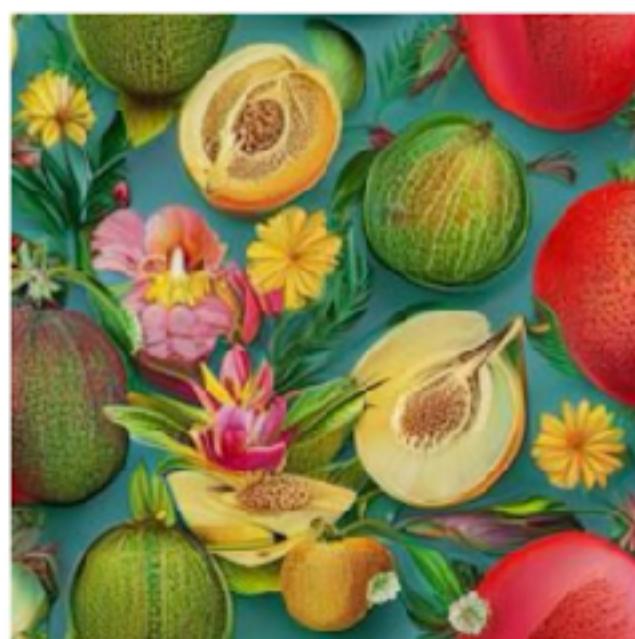
$$dX_t = b_t(X_t) + \sigma_t u_t^\theta(X_t) dt + \sigma_t dB_t$$

Fine-tuned Flow Matching

$$dX_t = v_t^\theta(X_t) dt$$



Memoryless  
(Fine-tuned)



Flow  
(Fine-tuned)

# Adjoint Matching for reward fine-tuning

Memoryless Flow Matching

$$dX_t = b_t(X_t) + \sigma_t u_t^\theta(X_t) dt + \sigma_t dB_t \rightarrow dX_t = v_t^\theta(X_t) dt$$

Fine-tuned Flow Matching



Flow  
(Pre-trained)



Flow  
(Fine-tuned)

# Reward fine-tuning for MovieGen Audio

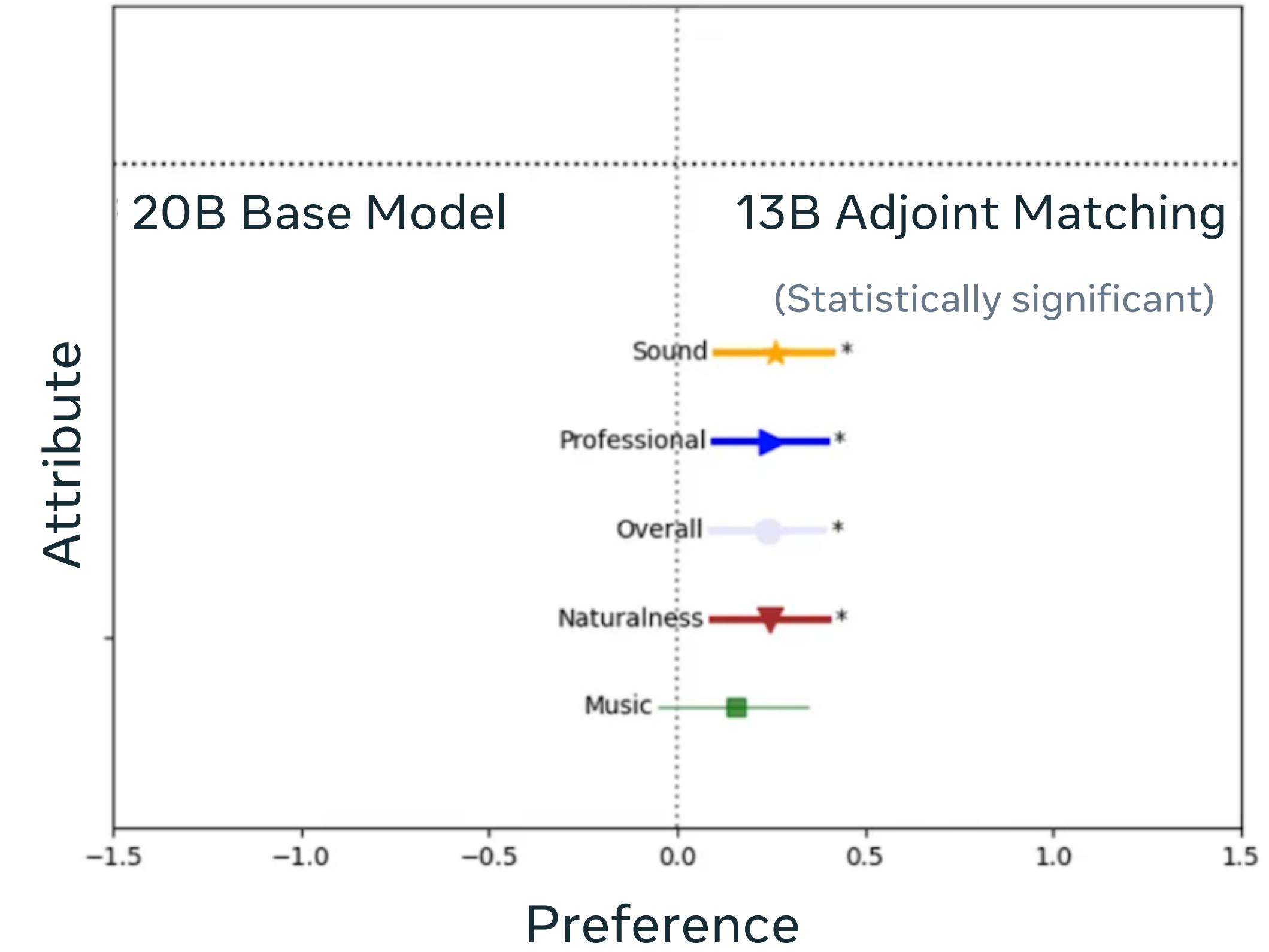
Meta MovieGen



Text & Video → Audio

Text input: “Whistling sounds, followed by a sharp explosion and loud crackling.”

Human Evaluation



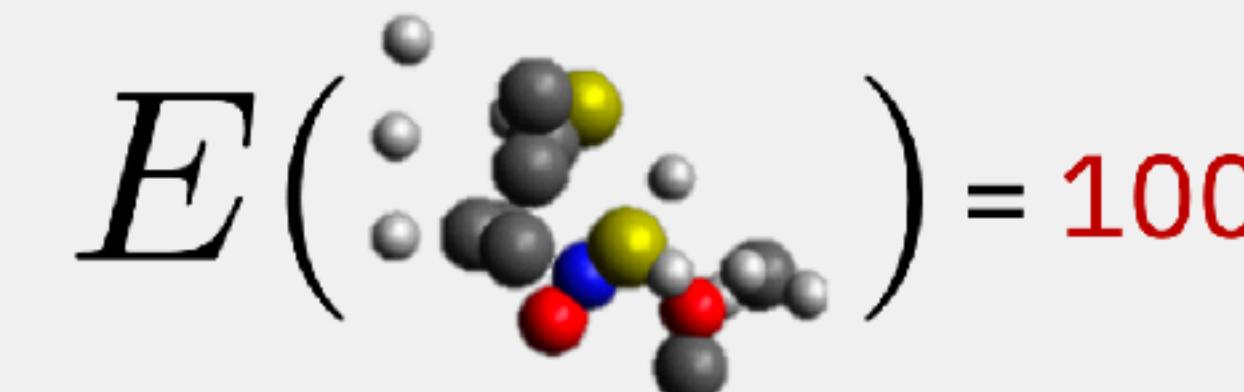
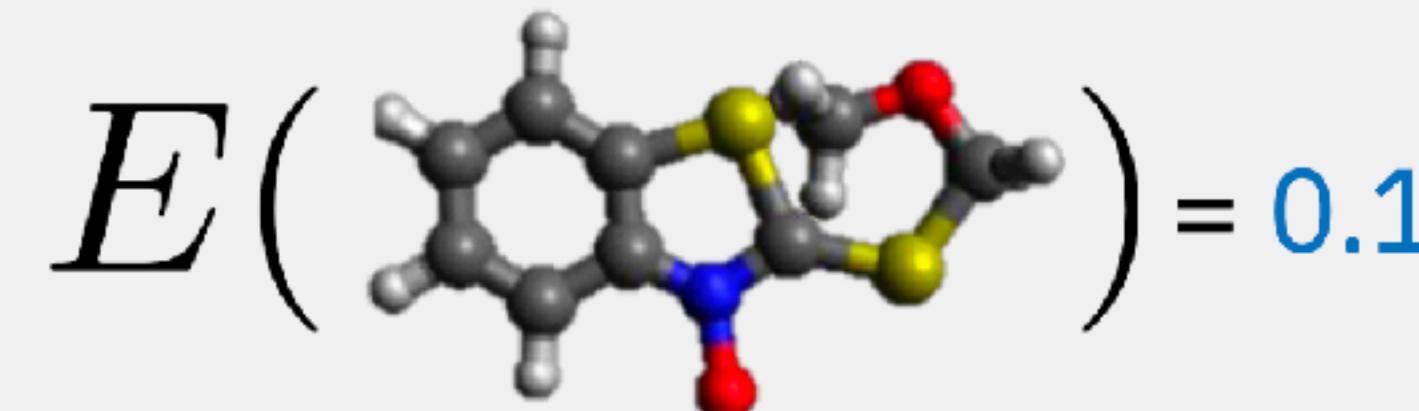
## II. How to scale the method?

**Adjoint Sampling:** highly scalable method for amortized sampling at scale

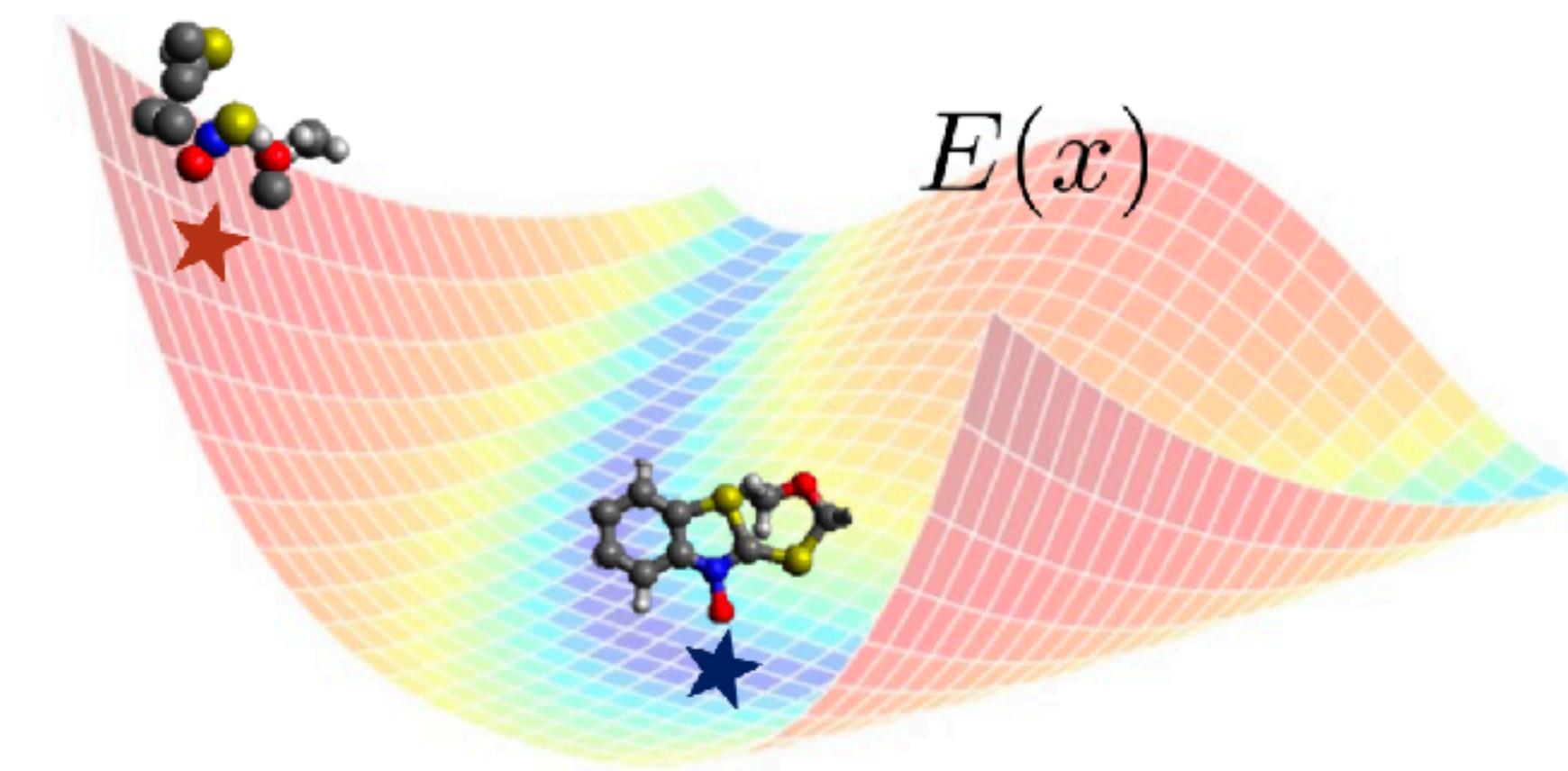
# Stochastic control formulation for sampling

Sampling from unnormalized distribution

$$p^*(x) \propto \exp\{-E(x)\}$$



low energy → stable structure → likely to appear → high probability  
high energy → unstable structure → unlikely to appear → low probability



# Stochastic control formulation for sampling

Sampling from unnormalized distribution

$$p^*(x) \propto \exp\{-E(x)\}$$

Know how to target this:

$$p_1^*(x) \propto p_1^{\text{base}}(x) \exp\{r(X_1)\}$$

Can define reward:

$$r(X_1) = -E(X_1) - \log p_1^{\text{base}}(X_1)$$

Stochastic control objective

$$\min_u \mathcal{L}(u) = \mathbb{E}_{p^u} \left[ \int_0^1 \frac{1}{2} \|u_t^\theta(X_t)\|^2 dt + E(X_1) + \log p_1^{\text{base}}(X_1) \right]$$

$$\text{s.t. } dX_t = [b_t(X_t) + \sigma_t u_t^\theta(X_t)] dt + \sigma_t dB_t, \quad X_0 \sim p$$

# Adjoint Matching simplified

Base generative process

$$dX_t = b_t(X_t)dt + \sigma_t dB_t \quad X_0 = 0$$

\*Adjoint Matching\*

“Lean” adjoint state

$$\frac{d}{dt} \tilde{a}_t(X_{[0,1]}) = -\tilde{a}_t(X_{[0,1]})^\top \nabla_{X_t} b_t(X_t)$$

$$\tilde{a}_1(X_{[0,1]}) = \nabla E(X_1) + \nabla \log p_1^{\text{base}}(X_1)$$

$$\int_0^1 \| u_t^\theta(X_t) + \sigma_t \tilde{a}_t(X_{[0,1]}) \|^2 dt$$

$$X_{[0,1]} \sim p^{\bar{u}}$$

$\bar{u} = \text{stopgrad}(u)$

# Adjoint Matching simplified

Base process

$$dX_t = \sigma_t dB_t \quad X_0 = 0$$

\*Adjoint Matching\*

“Lean” adjoint state

$$\frac{d}{dt} \tilde{a}_t(X_{[0,1]}) = 0$$

$$\tilde{a}_1(X_{[0,1]}) = \nabla E(X_1) + \nabla \log p_1^{\text{base}}(X_1)$$

$$\int_0^1 \| u_t^\theta(X_t) + \sigma_t \tilde{a}_t(X_{[0,1]}) \|^2 dt$$

$$X_{[0,1]} \sim p^{\bar{u}}$$

$\bar{u} = \text{stopgrad}(u)$

# Adjoint Matching simplified

Base process

$$dX_t = \sigma_t dB_t \quad X_0 = 0$$

\*Adjoint Matching\*

$$\int_0^1 \mathbb{E}_{(X_t, X_1) \sim p^u} \| u_t^\theta(X_t) + \sigma_t (\nabla E(X_1) + \nabla \log p_1^{\text{base}}(X_1)) \|^2 dt$$

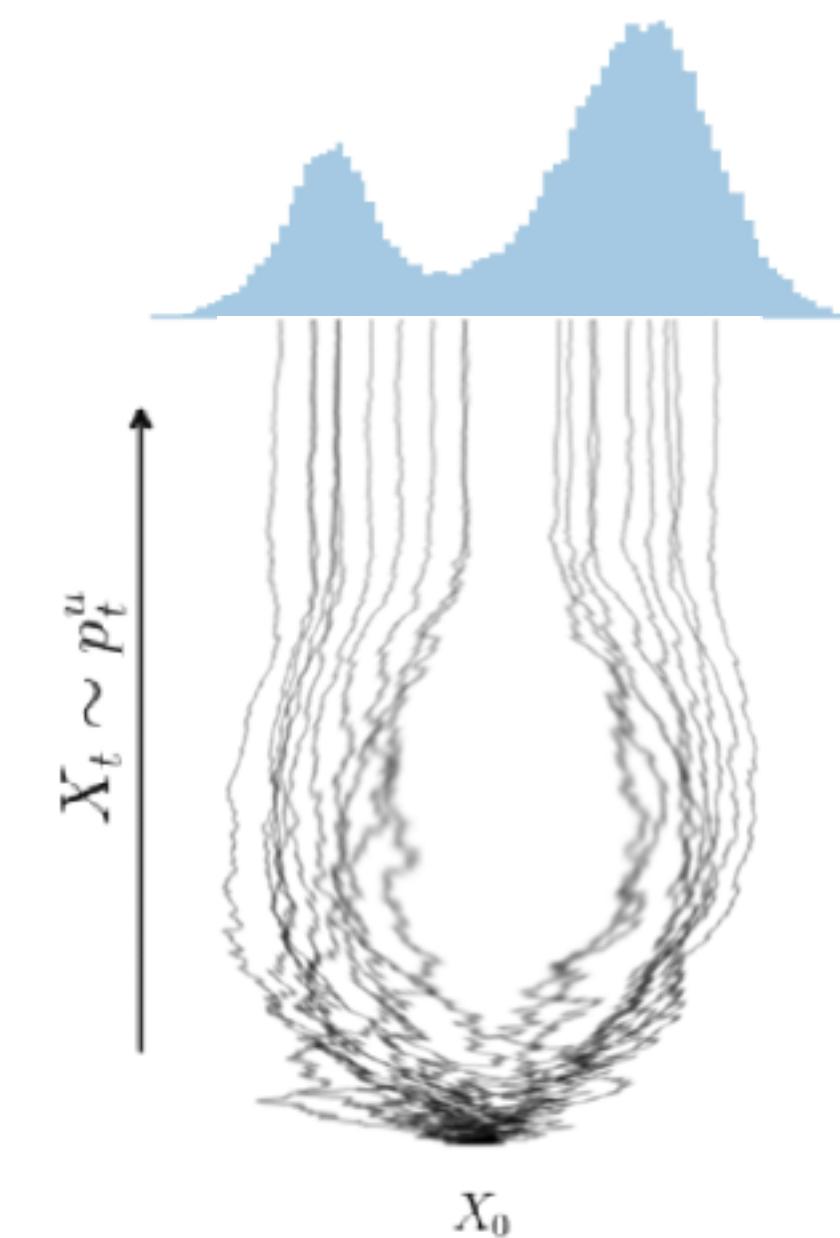
# Improving optimality with a projection

\*Adjoint Matching\*

$$\int_0^1 \mathbb{E}_{(X_t, X_1) \sim p^u} \|u_t^\theta(X_t) + \sigma_t(\nabla E(X_1) + \nabla \log p_1^{\text{base}}(X_1))\|^2 dt$$

Recall optimal stochastic process

$$u^* = \arg \min_{\nu} D_{\text{KL}}(p^\nu(X_{[0,1]}) \| p^{\text{base}}(X_{[0,1]} | X_1) p^*(X_1))$$



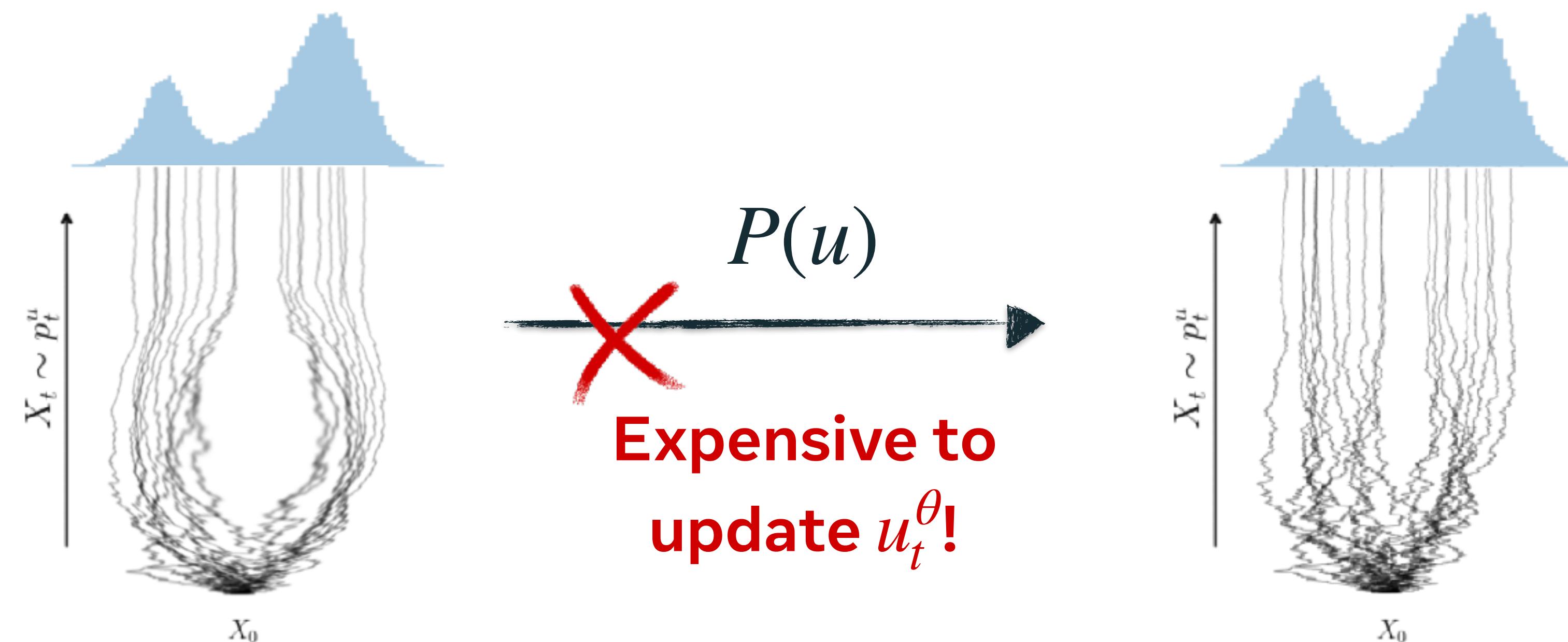
# Improving optimality with a projection

\*Adjoint Matching\*

$$\int_0^1 \mathbb{E}_{(X_t, X_1) \sim p^u} \|u_t^\theta(X_t) + \sigma_t(\nabla E(X_1) + \nabla \log p_1^{\text{base}}(X_1))\|^2 dt$$

Projection of the current control

$$P(u) = \arg \min_{\nu} D_{\text{KL}}(p^\nu(X_{[0,1]}) \| p^{\text{base}}(X_{[0,1]} | X_1) p^u(X_1))$$



# Improving optimality with a projection

\*Reciprocal Adjoint Matching\*

$$\int_0^1 \mathbb{E}_{X_1 \sim p_1^u, X_t \sim p^{\text{base}}(X_t | X_1)} \|u_t^\theta(X_t) + \sigma_t(\nabla E(X_1) + \nabla \log p_1^{\text{base}}(X_1))\|^2 dt$$

Sample  $X_t | X_1$  optimally

# Improving optimality with a projection

\*Reciprocal Adjoint Matching\*

$$\int_0^1 \mathbb{E}_{\substack{X_1 \sim p_1^u, X_t \sim p_{\text{base}}(X_t | X_1)}} \|\nu_t(X_t) + \sigma_t(\nabla E(X_1) + \nabla \log p_1^{\text{base}}(X_1))\|^2 dt$$

Leave Fixed

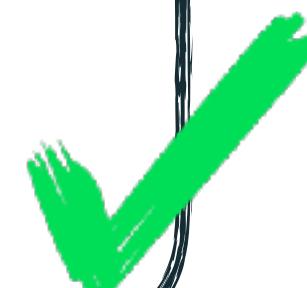
Sample  $X_t | X_1$  optimally

$=: \mathcal{L}_{\text{RAM}}(v; u)$

Theory of RAM: project for free

$$u_{i+1} = \arg \min_v \mathcal{L}_{\text{RAM}}(v; u_i)$$

$$= P(u_i) - \frac{\delta \mathcal{L}_{\text{AM}}}{\delta u}(P(u_i))$$



i.e., Equivalent to projecting  
the control then performing  
Adjoint Matching

Fixed point  $\iff$  optimal control

# Adjoint Sampling

\*Reciprocal Adjoint Matching\*

$$\int_0^1 \mathbb{E}_{\substack{X_1 \sim \mathcal{B}, X_t \sim p}} \text{base}_{(X_t|X_1)} \|u_t^\theta(X_t) + \sigma_t(\nabla E(X_1) + \nabla \log p_1^{\text{base}}(X_1))\|^2 dt$$

## Replay Buffer

### Adjoint Sampling Algorithm

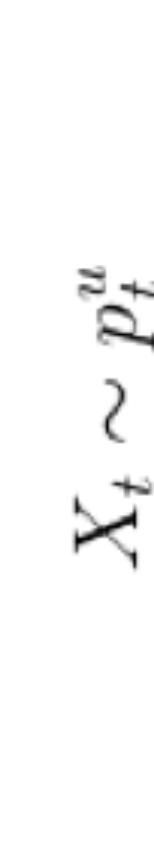
Alternate between:

1.  $\mathcal{B} \leftarrow \{X_1^i, E^i\}$ ,  $X_1^i \sim p_1^u$ ,  $E^i = E(X_1)$

2. Optimize  $\mathcal{L}_{\text{RAM}}$  multiple iterations

Infrequent sample generation & energy evaluation.

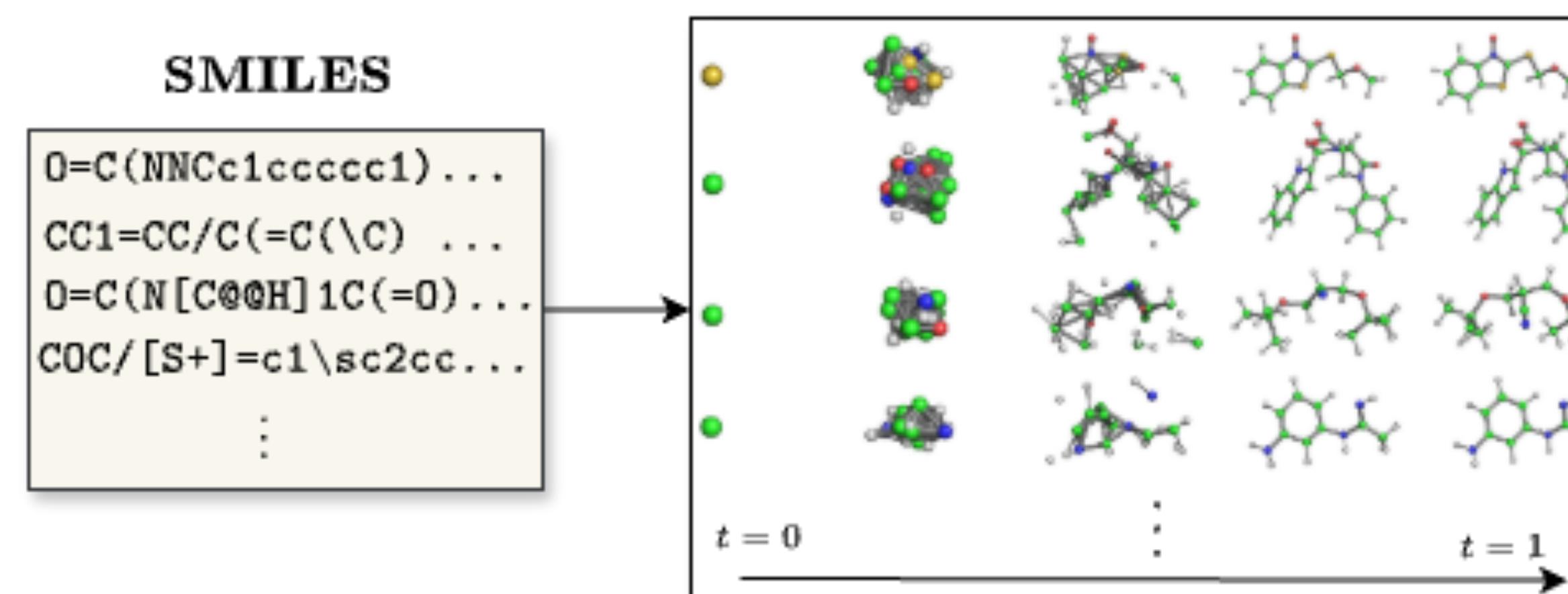
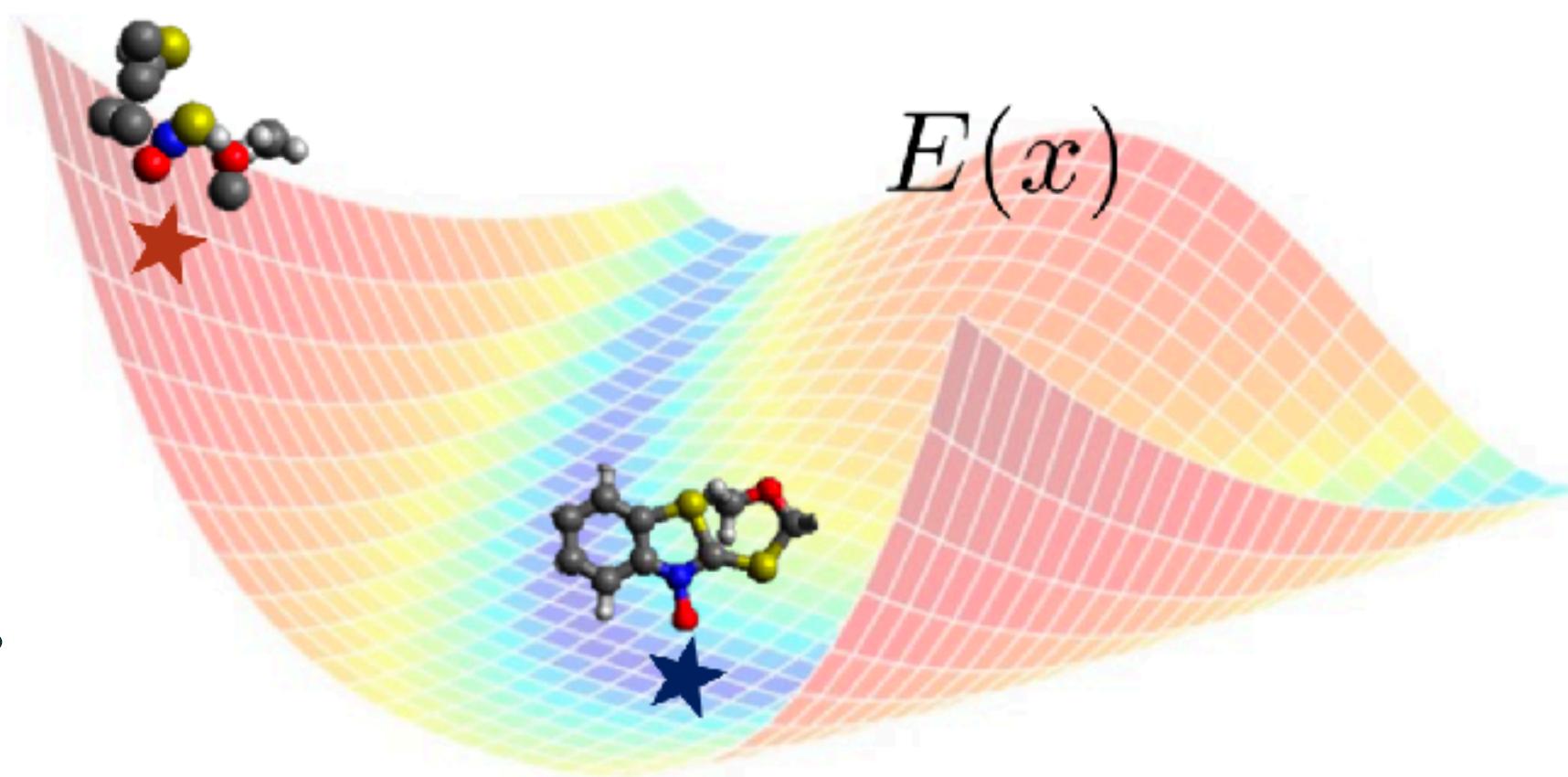
Very fast gradient updates.



# A new benchmark for highly scalable sampling

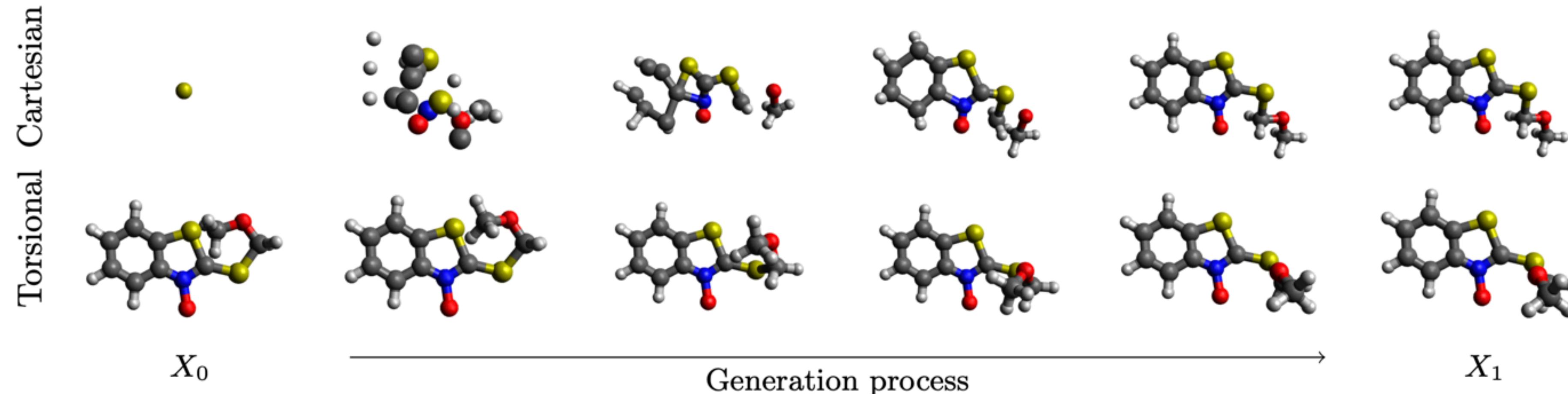
Evaluates sampling methods on both **efficiency** and **generalization**.

Energy evaluations  
modeled by **deep  
graph neural networks**.

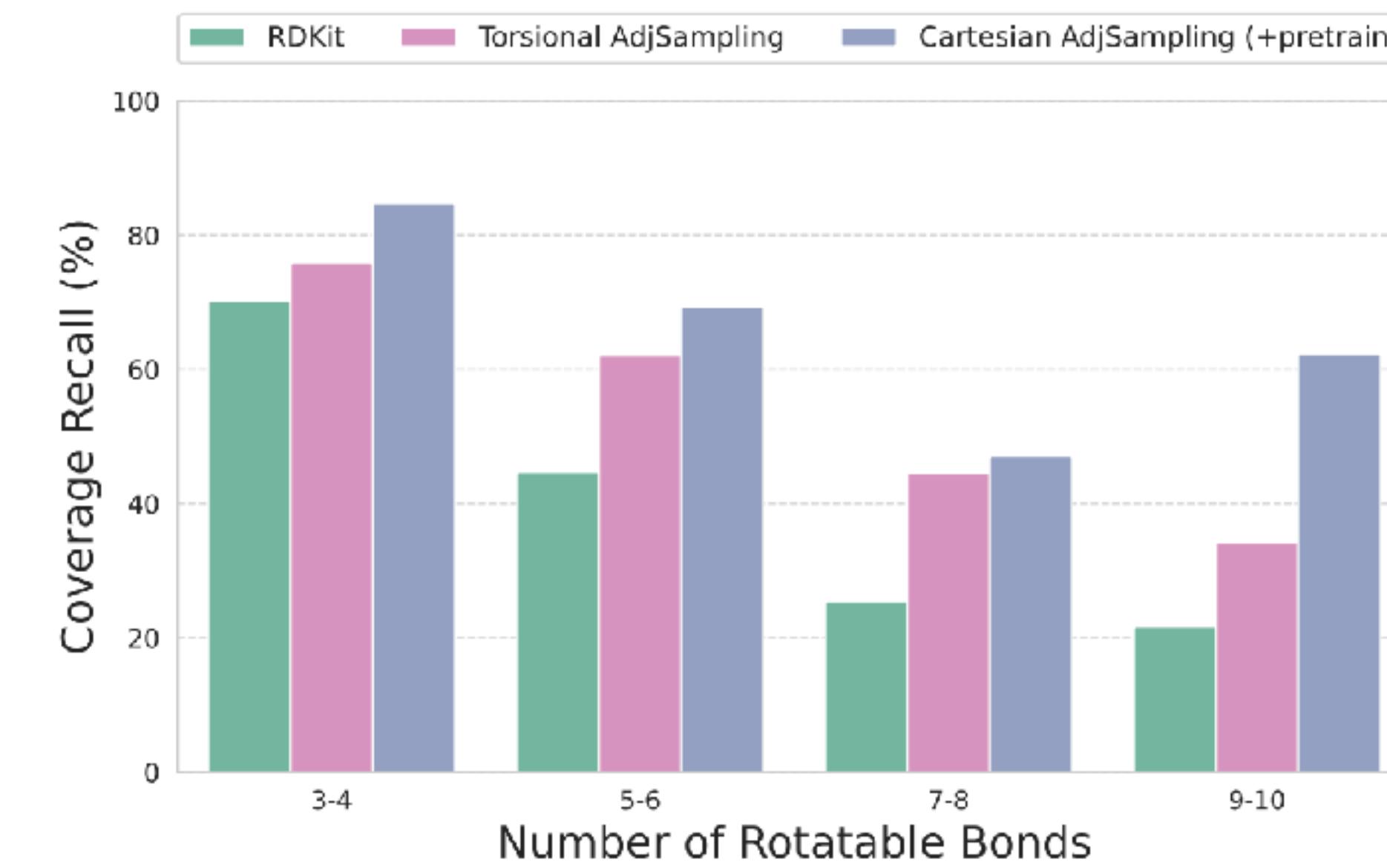


Train on **24,000** molecules.  
Each with  $\mathcal{O}(100)$  of local minima. **Need to find all**.  
Tests generalization to **unseen molecules**.

# A new benchmark for highly scalable sampling



More improvement  
over RD-KIT at  
higher difficulties



More difficult / more local minima

# Papers & Collaborators

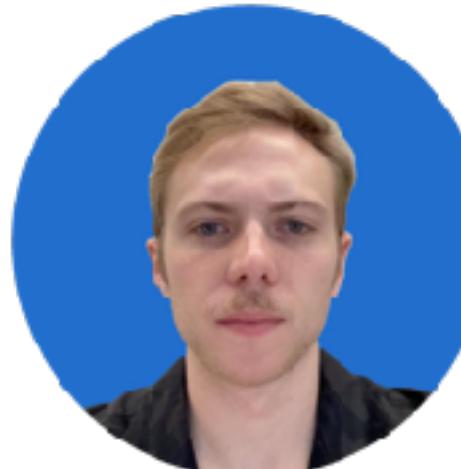
## Adjoint Matching: Fine-tuning Flow and Diffusion Generative Models with Memoryless Stochastic Optimal Control

Carles Domingo-Enrich<sup>1</sup>, Michal Drozdzal<sup>1</sup>, Brian Karrer<sup>1</sup>, Ricky T. Q. Chen<sup>1</sup>

<sup>1</sup>FAIR, Meta



Carles Domingo I  
Enrich



Aaron Joseph  
Havens



Ben Miller



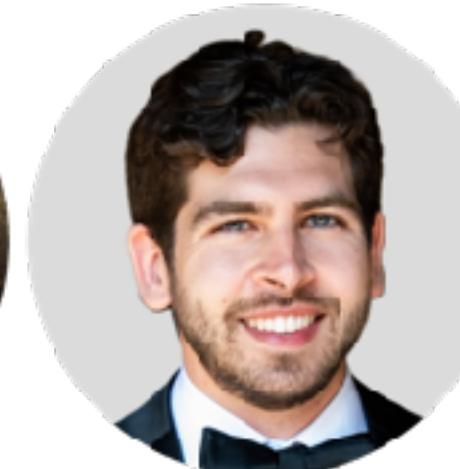
Bing Yan



Anuroop Sriram



Brandon Wood



Daniel Levine



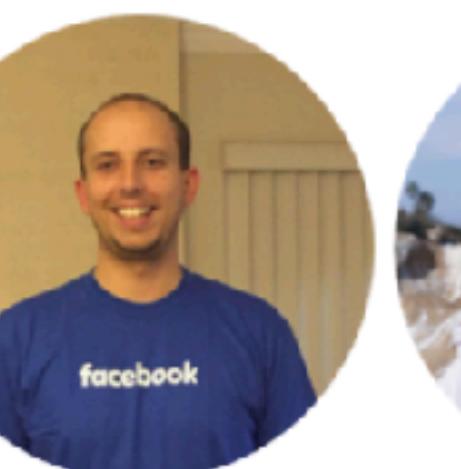
Bin Hu



Brandon Amos



Brian Karrer



Michal Drozdzal



Xiang Fu



Guan-Horng Liu

## Adjoint Sampling: Highly Scalable Diffusion Samplers via Adjoint Matching

Aaron Havens<sup>2,†,\*</sup>, Benjamin Kurt Miller<sup>1,\*</sup>, Bing Yan<sup>1,3,\*</sup>, Carles Domingo-Enrich<sup>4</sup>, Anuroop Sriram<sup>1</sup>, Brandon Wood<sup>1</sup>, Daniel Levine<sup>1</sup>, Bin Hu<sup>2</sup>, Brandon Amos<sup>1</sup>, Brian Karrer<sup>1</sup>, Xiang Fu<sup>1,\*</sup>, Guan-Horng Liu<sup>1,\*</sup>, Ricky T. Q. Chen<sup>1,\*</sup>

<sup>1</sup>FAIR at Meta, <sup>2</sup>University of Illinois, <sup>3</sup>New York University, <sup>4</sup>Microsoft Research New England

\*Core contributors, †Work done during internship at FAIR