Optimizing DDPM Sampling with Shortcut Fine-Tuning

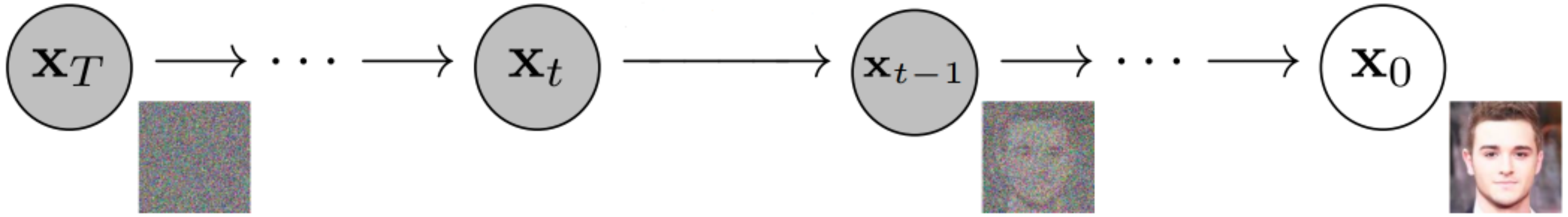
Ying Fan

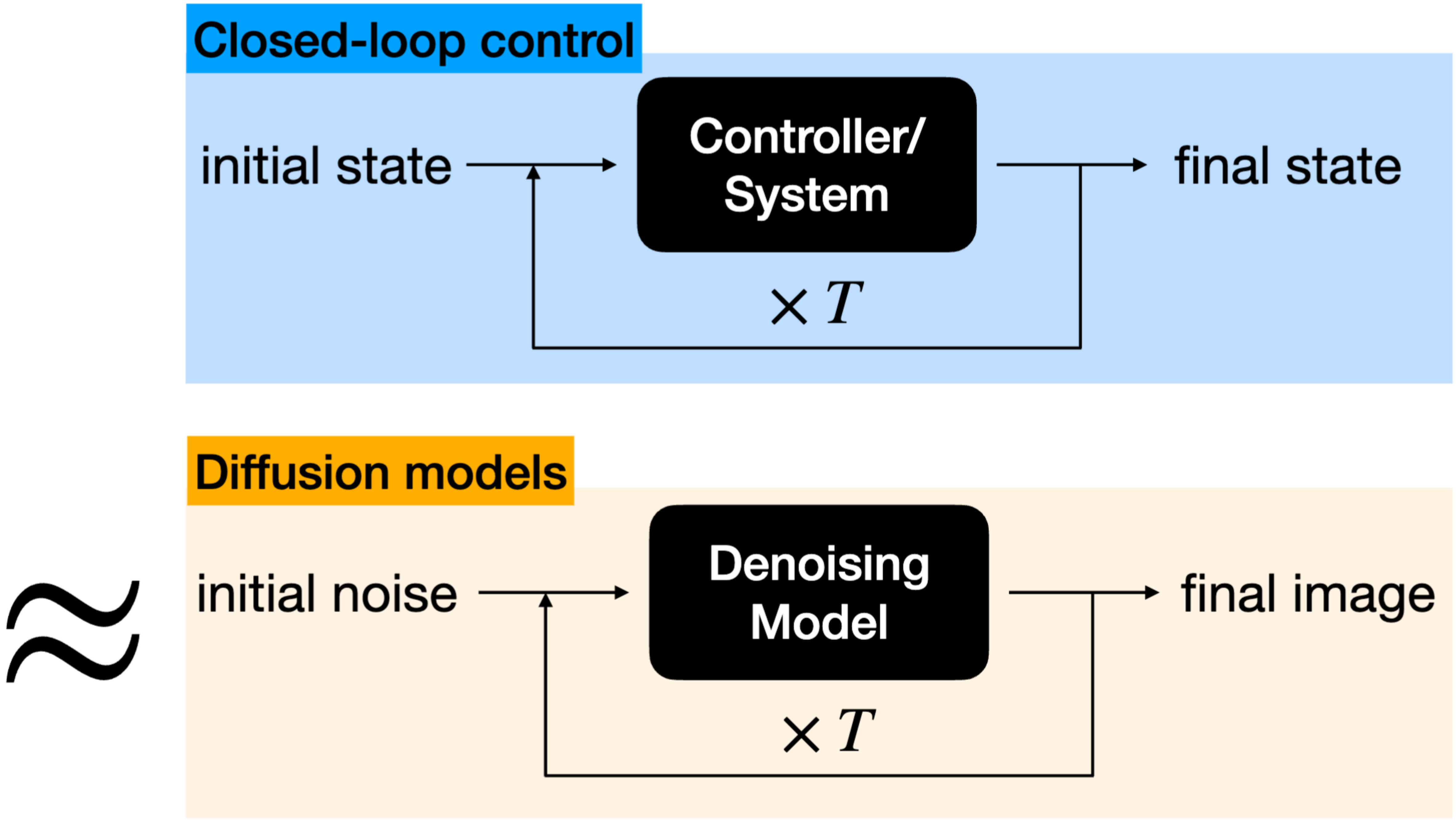
Kangwook Lee

University of Wisconsin-Madison

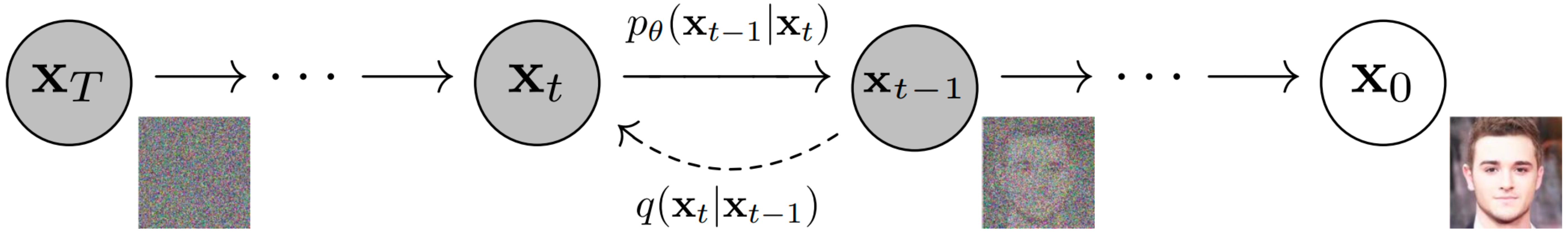
Motivation

- Diffusion models find paths from pure noise to natural images





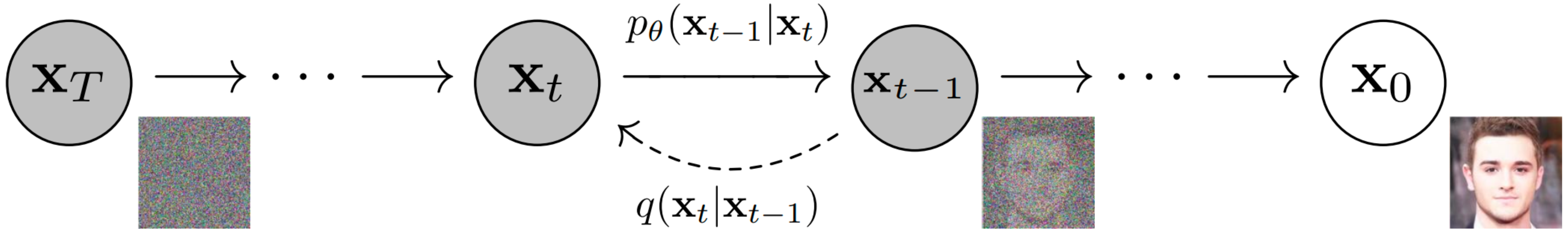
Motivation



- The training objective of DDPM [1] is ***behavior cloning***:
  - Given a fixed Markov forward process q

[1] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840-6851.

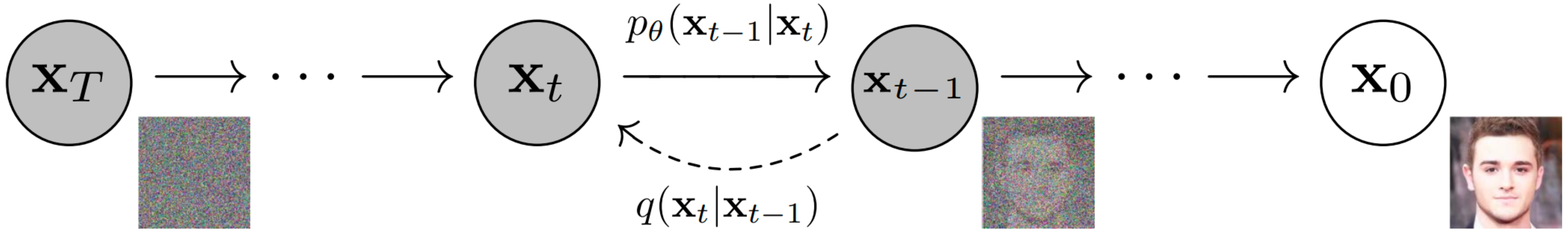
Motivation



- The training objective of DDPM [1] is ***behavior cloning***:
  - Given a fixed Markov forward process *q*
  - Compute the reverse Markov process of *q*

[1] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840-6851.

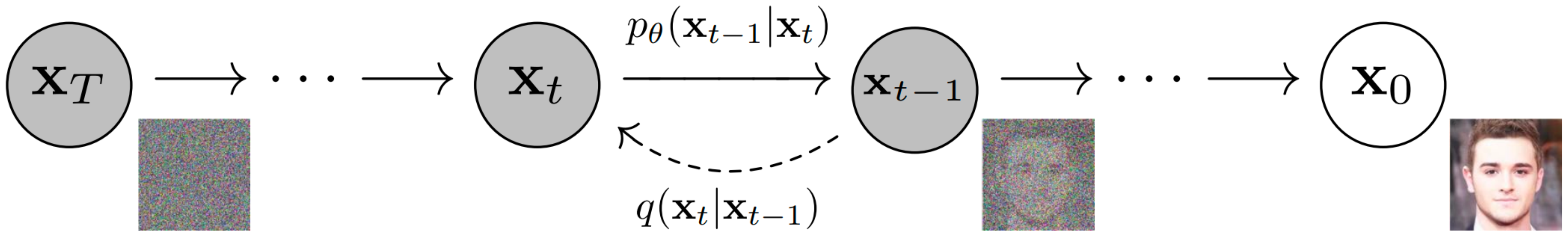
Motivation



- The training objective of DDPM [1] is ***behavior cloning***:
  - Given a fixed Markov forward process q
  - Compute the reverse Markov process of q
  - Train a model θ to follow the reverse process

[1] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840-6851.

Motivation



- The training objective of DDPM [1] is ***behavior cloning***:
  - Given a fixed Markov forward process q
  - Compute the reverse Markov process of q
  - Train a model θ to follow the reverse process

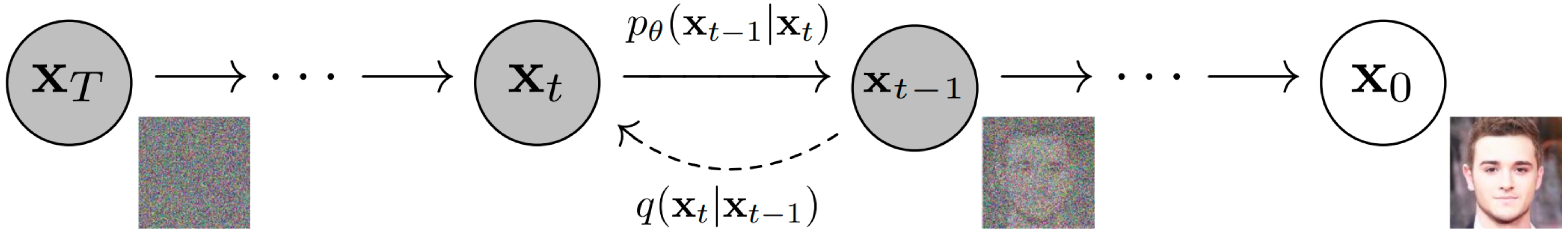
$$J = \mathbb{E}_q \left[\sum_{t=0}^{T-1} D_{KL}(q(x_t \mid x_{t+1}, x_0), p_t^\theta(x_t \mid x_{t+1})) \right]$$

$$\downarrow$$

Match all marginal distributions: $p_\theta(x_t) \approx q(x_t)$

[1] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840-6851.

Motivation



- The training objective of DDPM [1] is ***behavior cloning***:
  - Given a fixed Markov forward process q
  - Compute the reverse Markov process of q
  - Train a model θ to follow the reverse process

Behavior cloning is not always the best way

↓

What if we do ***online learning*** to find alternative paths/shortcuts?

[1] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840-6851.

From Behavior Cloning to Online Learning

- For BC, we assume the behavior policy is from the “expert”
  - No “reward”/“cost” is needed
  - Just imitating the behavior trajectories is enough

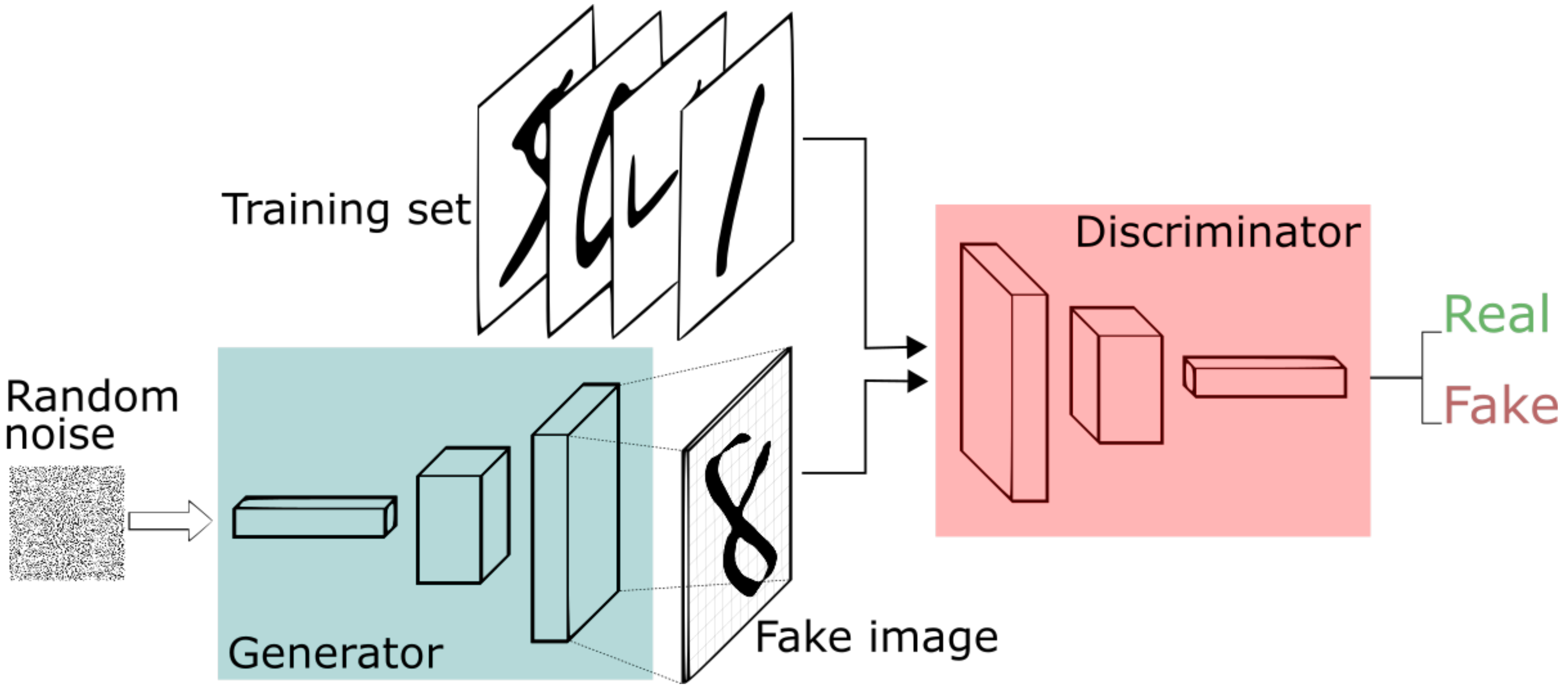
From Behavior Cloning to Online Learning

- For BC, we assume the behavior policy is from the "expert"
  - No "reward"/"cost" is needed
  - Just imitating the behavior trajectories is enough

- For online learning, we need feedback from the environment (as reward/cost)
  - Where does the feedback come from?

From Behavior Cloning to Online Learning

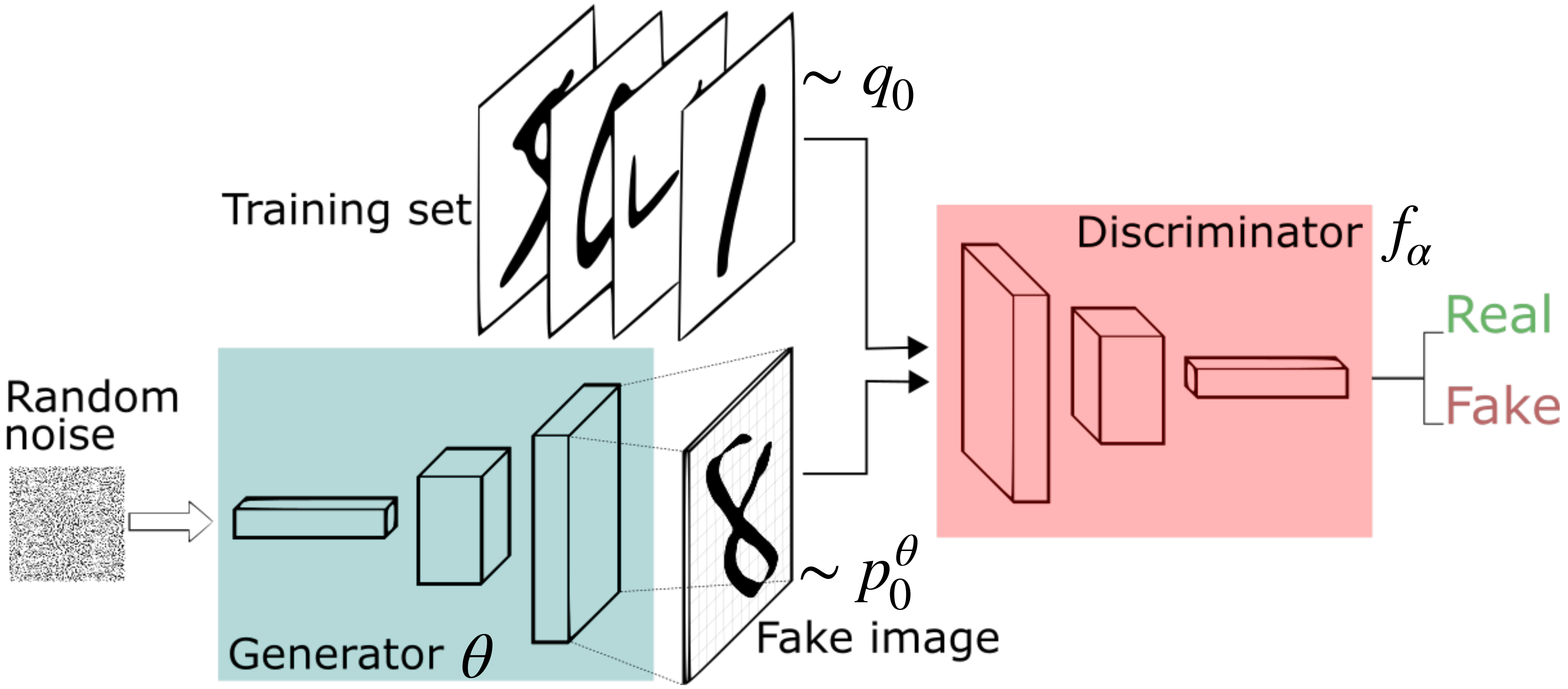
- For online learning, we need feedback from the environment (as reward/cost)
  - Where does the feedback come from?
  - Recall the task: generative modeling
  - Recall GAN [2] training:
    - We can train a discriminator to compute some metric that describes the discrepancy between the synthetic & real distributions
    - The generator is updated to minimize this metric



[2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139-144.

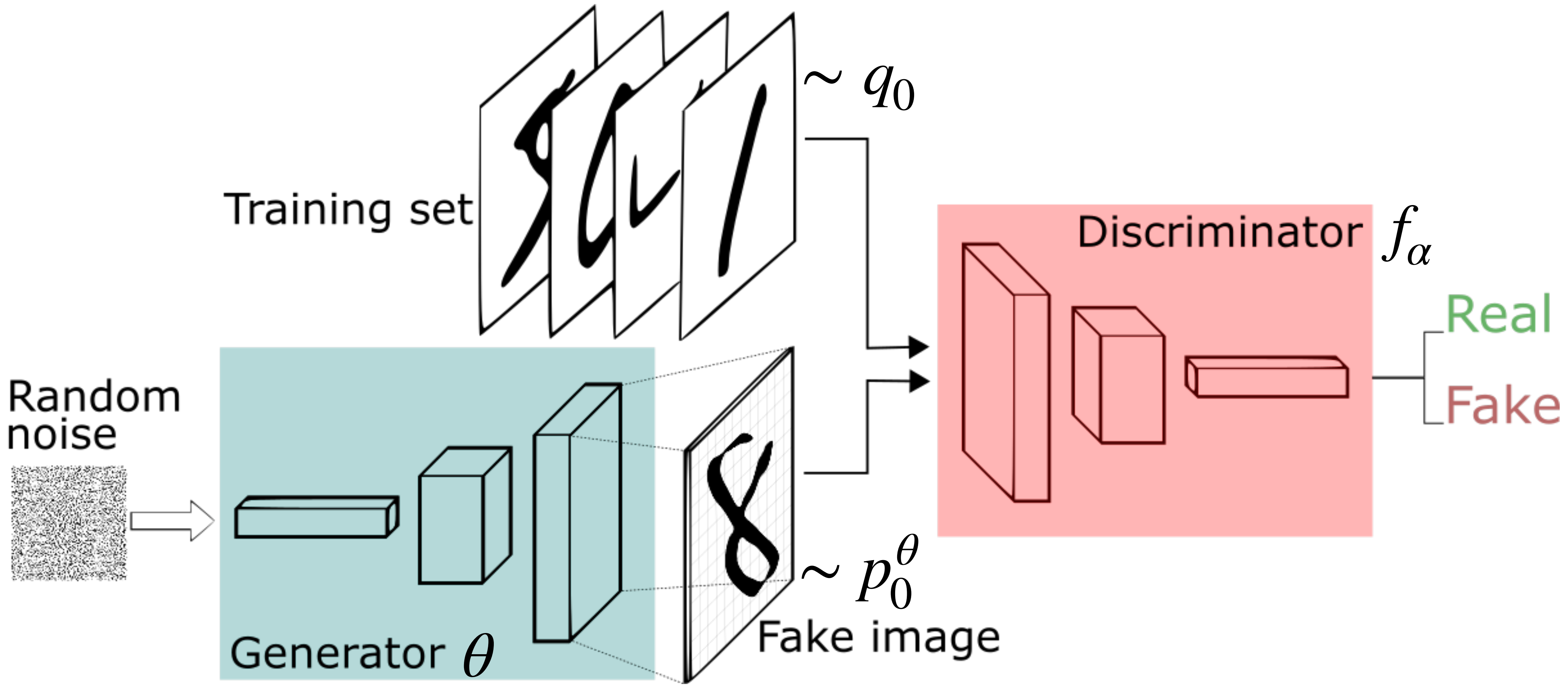
Online Learning: Generator & Discriminator

- We formulate the diffusion model as a multi-step *generator*
- We can train a *discriminator* model based on the generated images & the true images



Online Learning: Generator & Discriminator

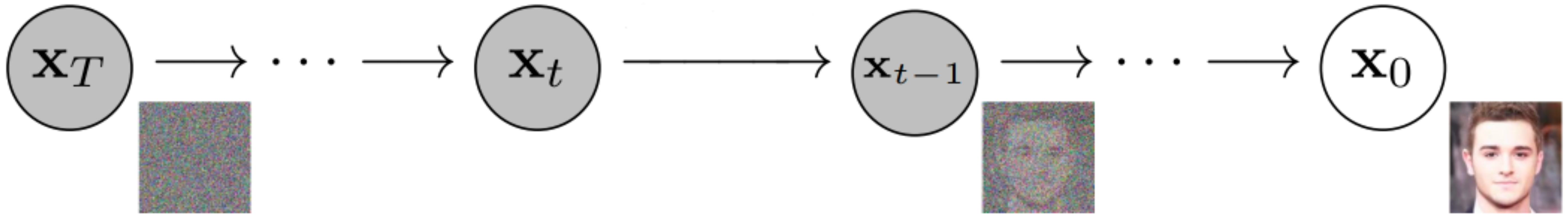
- We formulate the diffusion model as a multi-step *generator*
- We can train a *discriminator* model based on the generated images & the true images



- Our objective function: Integral Probability Metric (IPM)

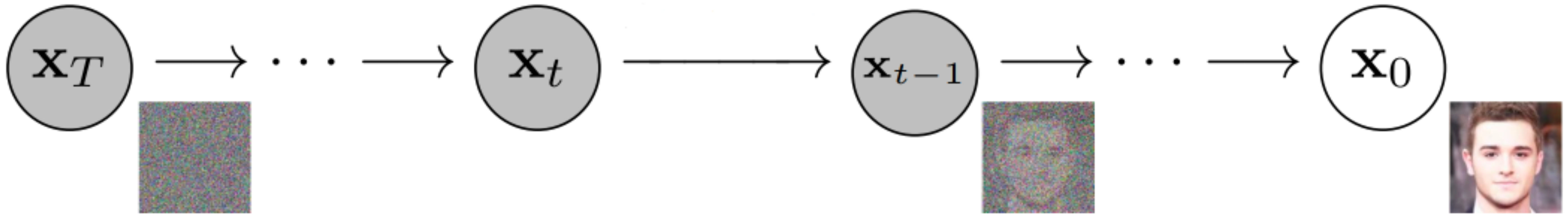
$$\Phi(p_0^\theta, q_0) = \sup_{\alpha \in \mathcal{A}} \quad \mathbb{E}_{x_0 \sim p_0^\theta}[f_\alpha(x_0)] - \mathbb{E}_{x_0 \sim q_0}[f_\alpha(x_0)]$$

Online Learning: Generator & Discriminator



- How to update the multi-step generator with discriminator signals?

Online Learning: Generator & Discriminator



- How to update the multi-step generator with discriminator signals?
  - Multi-step generator as an RNN?
    - It requires $T\times$ original memory usage
    - $T\times$ multiplication of gradients could results in instability in training
  - Is there an alternative way?

Online Learning: Policy Gradient Equivalence

- Recall our objective function: Integral Probability Metric (IPM)

$$\Phi(p_0^\theta, q_0) = \sup_{\alpha \in \mathscr{A}} \quad \mathbb{E}_{x_0 \sim p_0^\theta}[f_\alpha(x_0)] - \mathbb{E}_{x_0 \sim q_0}[f_\alpha(x_0)]$$

- Optimizing the sampling process with some discriminator = policy gradient under regularity assumptions:

$$\nabla_\theta \Phi(p_0^\theta, q_0) = \mathbb{E}_{p^\theta_{x_{0:T}}} \Big[\underbrace{f_{\alpha^*(p_0^\theta, q_0)}(x_0)}_{\text{“Cumulative cost”}} \nabla_\theta \log \sum_{t=0}^{T-1} p_t^\theta(x_t \mid x_{t+1}) \Big]$$

- RL formulation:
  - Discriminator as a cost function at the end of trajectory and 0 elsewhere
  - Diffusion model as a policy, the next state to be identical to the action
  - Then it is equivalent to policy gradient!
  - We can also train a value function for variance reduction

Online Learning: Towards Monotonic Improvements

$$\nabla_\theta \Phi(p_0^\theta, q_0) = \mathop{\mathbb{E}}_{p^\theta_{x_{0:T}}} \left[\underline{f_{\alpha^*(p_0^\theta, q_0)}}(x_0)\, \nabla_\theta \log \sum_{t=0}^{T-1} p_t^\theta(x_t \mid x_{t+1}) \right]$$

The cost is dependent on current θ

- Is there a way to reuse the previous discriminator even if we update the generator?
  - Can we still can get a descent in the objective?
  - Yes, if the change is small enough!

Online Learning: Towards Monotonic Improvements

$$\nabla_\theta \Phi(p_0^\theta, q_0) = \mathop{\mathbb{E}}_{p^\theta_{x_{0:T}}} \left[\underline{f_{\alpha^*(p_0^\theta, q_0)}(x_0)} \, \nabla_\theta \log \sum_{t=0}^{T-1} p_t^\theta(x_t \mid x_{t+1}) \right]$$

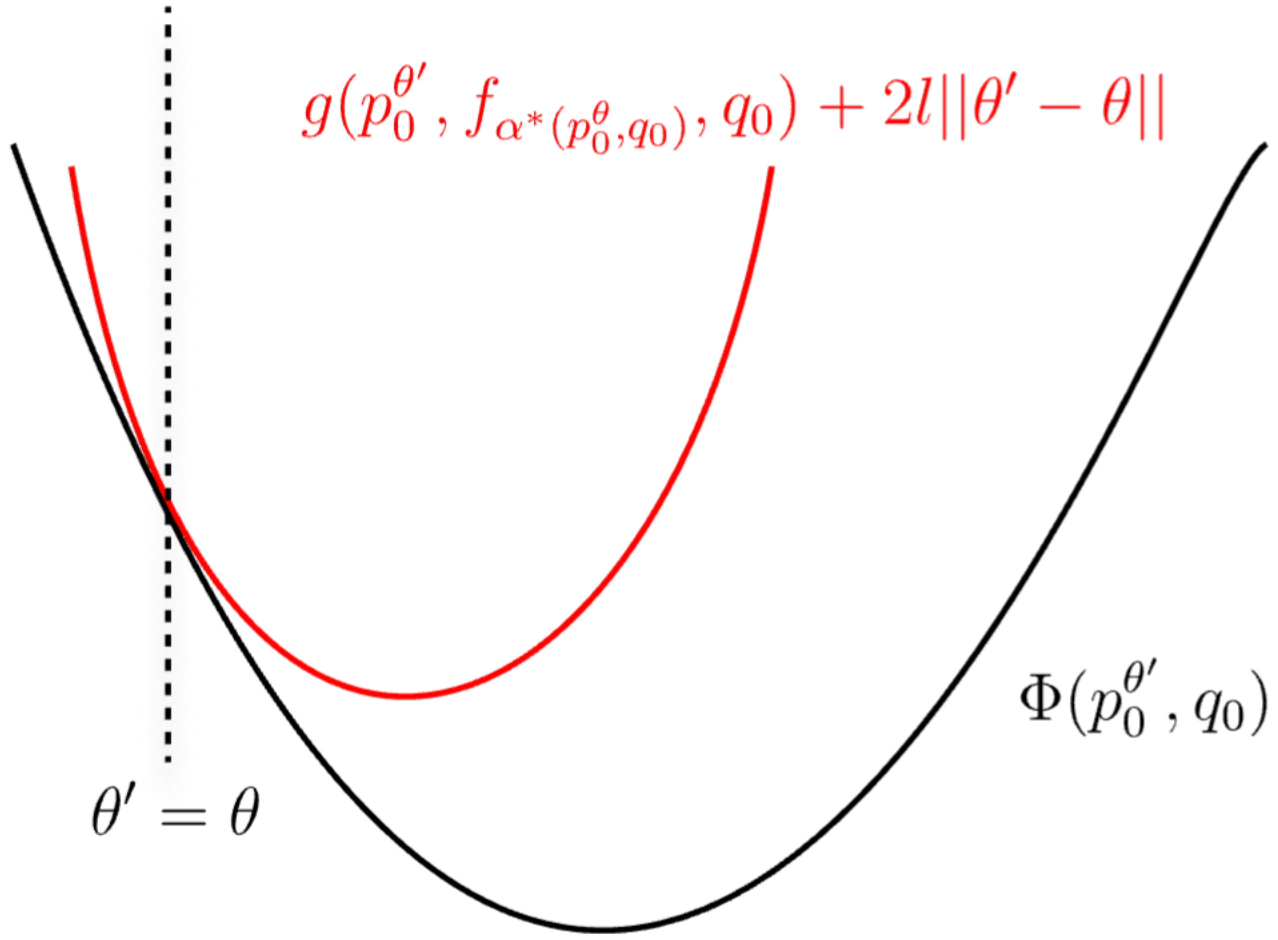
The cost is dependent on current θ

- Is there a way to reuse the previous discriminator even if we update the generator?
  - Can we still can get a descent in the objective?
  - Yes, if the change is small enough!

- Under Lipschitz assumptions, we have

$$\Phi(p_0^{\theta'}, q_0) \le \underline{g(p_0^{\theta'}, f_{\alpha^*(p_0^\theta, q_0)}, q_0) + 2l \,||\theta' - \theta|}$$

As a surrogate function



- Similar to TRPO [3] — but not the same
- To optimize the surrogate function, one can simply clip the gradient during update

[3] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In *International conference on machine learning* (pp. 1889-1897). PMLR.

Online Learning: Regularizing the Discriminator

- The discriminator needs to provide meaningful gradients during the generator updates
  - The generator only uses the scalar value of the discriminator, not its gradient like WGAN [4]
  - We only need the scalar value to include both high&low costs
  - We can afford a wider class of discriminators to provide more signals! (*No need of gradient penalty*)

[4] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214-223). PMLR.

Online Learning: Combining the Pipeline

- The training pipeline: **SFT-PG** (*shortcut fine-tuning with policy gradient*)

  - **For discriminator steps**:
    - Generate samples from the generator
    - Train the discriminator to maximize the discrepancy between real&fake images
    - Train the value function and also regularize the discriminator

  - **For generator steps:**
    - Perform policy gradient update with gradient clipping

  - Continue the loop till convergence

Experimental Results

- Initialize the diffusion model ($T = 1000$) with subsampling $T' = 10$
- Optimize the model with our algorithm to improve the model
- Results:

(a) CIFAR10, Initialization (b) CIFAR10, SFT-PG (B) (c) CelebA, Initialization (d) CelebA, SFT-PG (B)

- FID:

Method	**CIFAR-10** (32×32)	**CelebA** (64×64)
DDPM	34.76	36.69
FastDPM	29.43	28.98
Analytic-DPM	22.94	28.99
SN-DDPM	16.33	20.60
SFT-PG (B)	**2.28**	**2.01**

Experimental Results

- Initialize the diffusion model ($T = 1000$) with subsampling $T' = 10$
- Optimize the model with our algorithm to improve the model
- Results:

(a) CIFAR10, Initialization (b) CIFAR10, SFT-PG (B) (c) CelebA, Initialization (d) CelebA, SFT-PG (B)

- FID:

Method (DDPM, stochastic)	NFE	FID	Method (DDIM, deterministic)	NFE	FID
DDPM	10	34.76	DDIM	10	17.33
SN-DDPM	10	16.33	DPM-solver	10	4.70
SFT-PG*	10	2.28			
SFT-PG*	8	2.64	Progressive distillation*	8	2.57

Summary

- We propose a novel algorithm, using policy gradient to fine-tuning diffusion models w.r.t. IPM, which first utilizes reinforcement learning (RL) methods to train diffusion models

- We propose a surrogate function of IPM for monotonic improvements

- We propose a gradient-free regularization for the critic

- Our fine-tuning can improve DDPM sampling to reduce the number of sampling steps

GitHub https://github.com/UW-Madison-Lee-Lab/SFT-PG

Follow-up Works

- Follow-up works: Fine-tuning text-to-image models using some rewards defined on the text-image space

  - DPOK: Reinforcement Learning for Fine-tuning Text-to-Image Diffusion Models [5]

  - Training Diffusion Models with Reinforcement Learning [6]

[5] Fan, Y., Watkins, O., Du, Y., Liu, H., Ryu, M., Boutilier, C., ... & Lee, K. (2023). DPOK: Reinforcement Learning for Fine-tuning Text-to-Image Diffusion Models. *arXiv preprint arXiv:2305.16381*.
[6] Black, K., Janner, M., Du, Y., Kostrikov, I., & Levine, S. (2023). Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*.

Thank you!