Diffusion Models: Acceleration and Distillation

Aibek Alanov

Head of the Center of Deep Learning and Bayesian Methods HSE University

Lead of Controllable Generative AI at AIRI

Contents

- Development of generative models
- Diffusion models
- Acceleration approaches

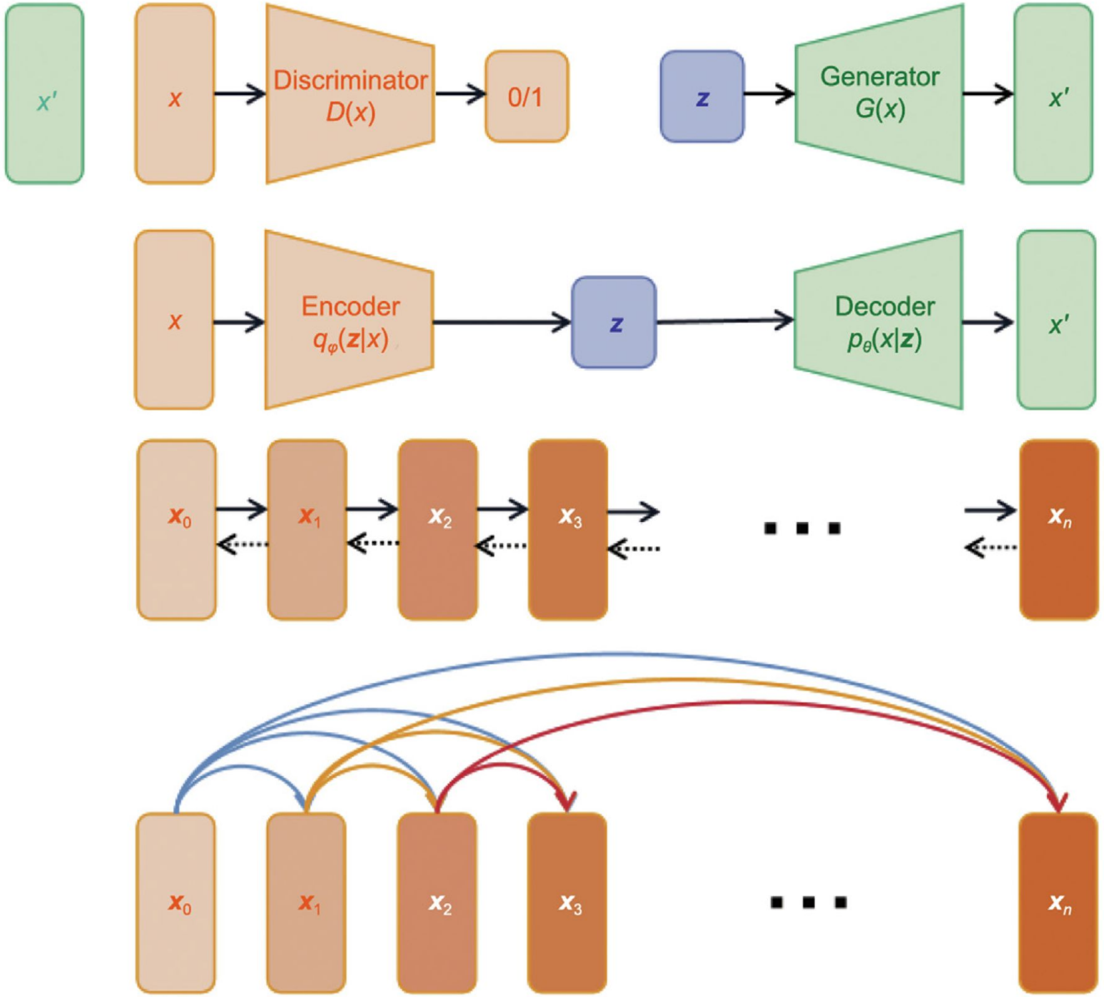
Development of generative models

GANs

VAE

Diffusion models

Autoregressive models



Development of generative models for images/video

2014 **2016** **2018**

VAE

GANs

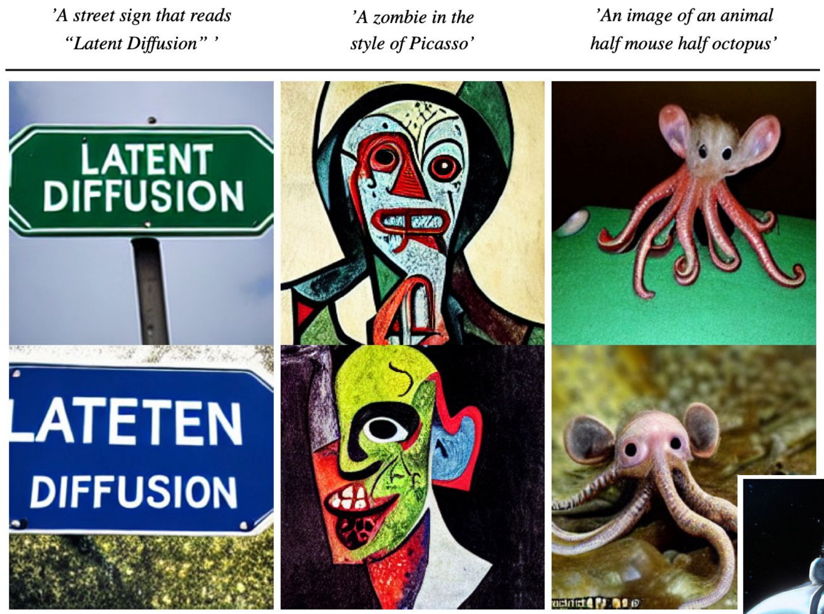
Autoregressive PixelCNN

StyleGAN

BigGAN

Development of generative models for images/video

2021 **2022** **2023**



Stable Diffusion

Dall-E 2

a dolphin in an astronaut suit on saturn, artstation

a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese

a teddy bear on a skateboard in times square

SDXL

AIRI

Development of generative models for images/video

2024

Flux

Veo 2

AIRI

Development of generative models for images/video

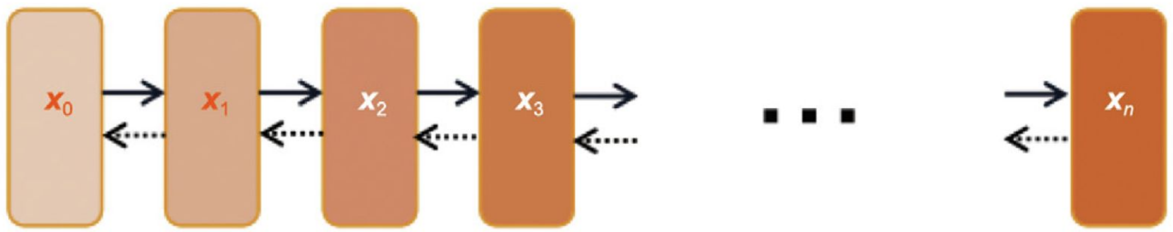
2025

GPT-4o

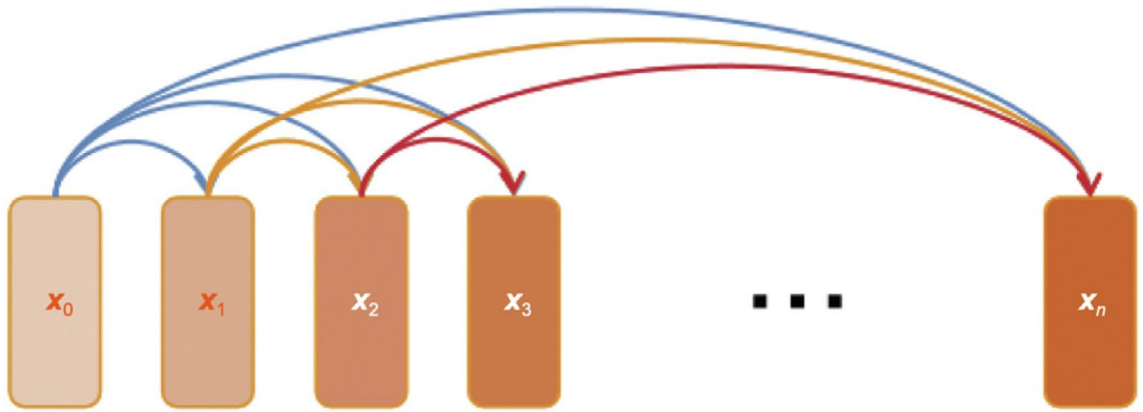
Veo 3

The most successful generative models

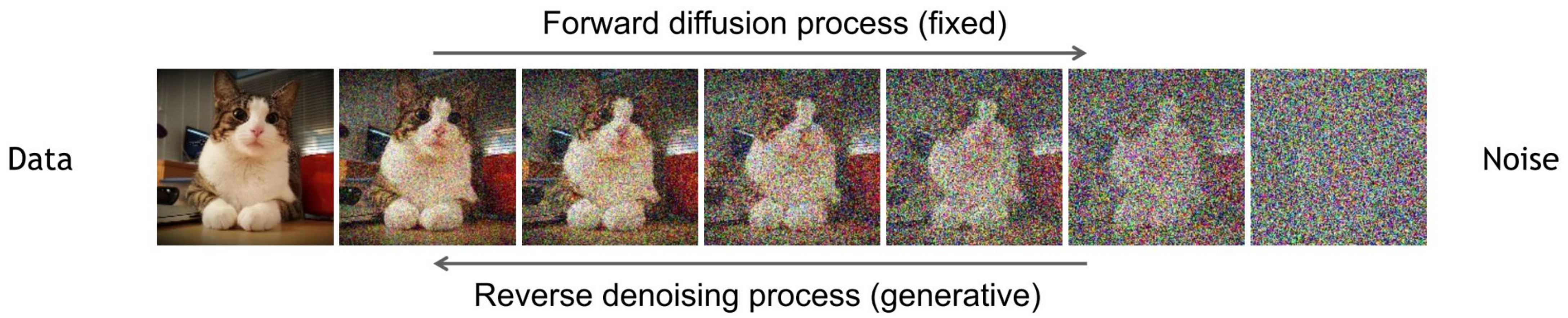
Diffusion models



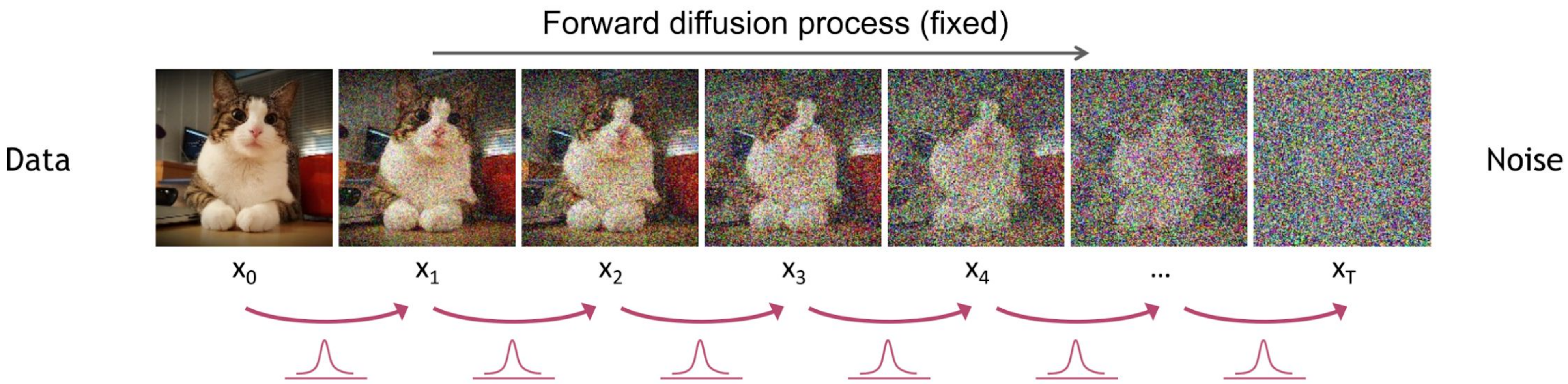
Autoregressive models



Diffusion models



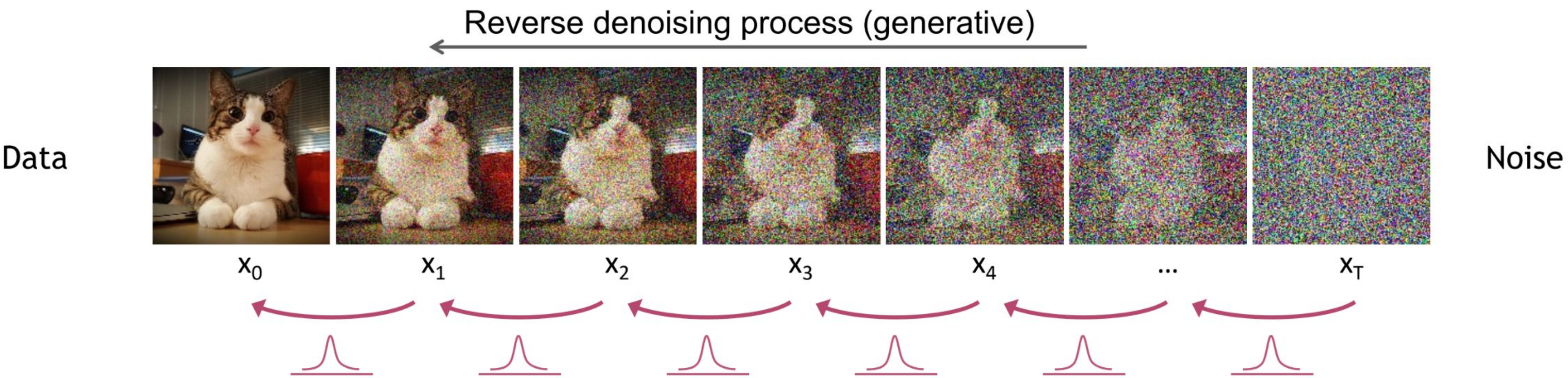
Diffusion models



$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

(joint)

Diffusion models



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}_{\text{Trainable network}}, \sigma_t^2\mathbf{I})$$

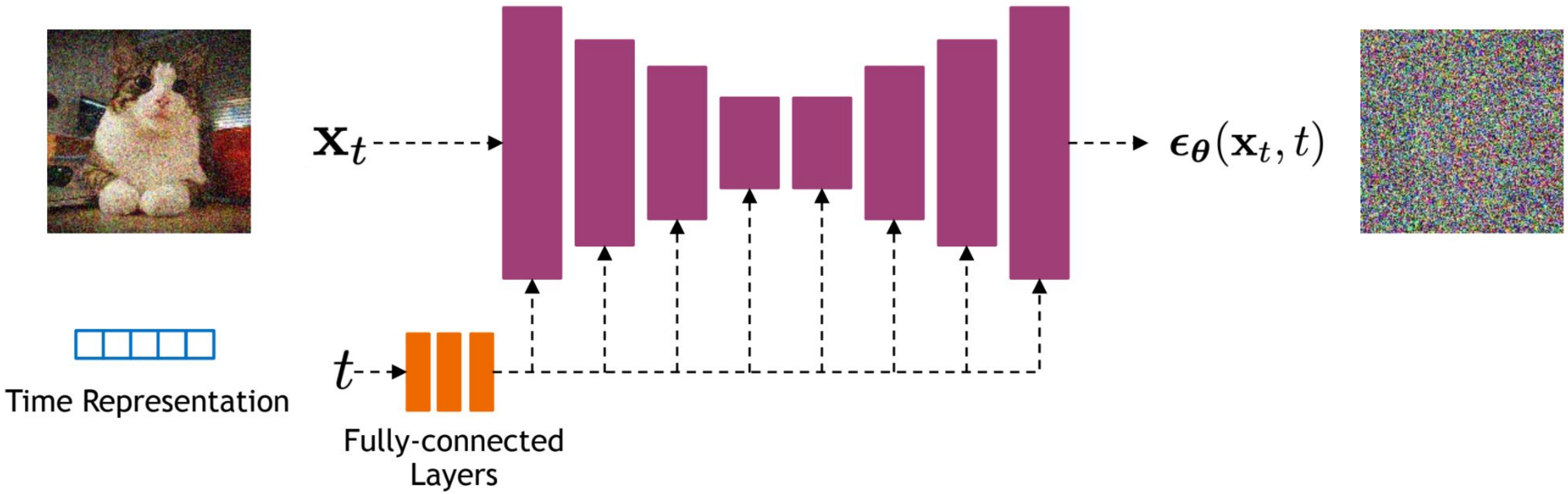
$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Diffusion models

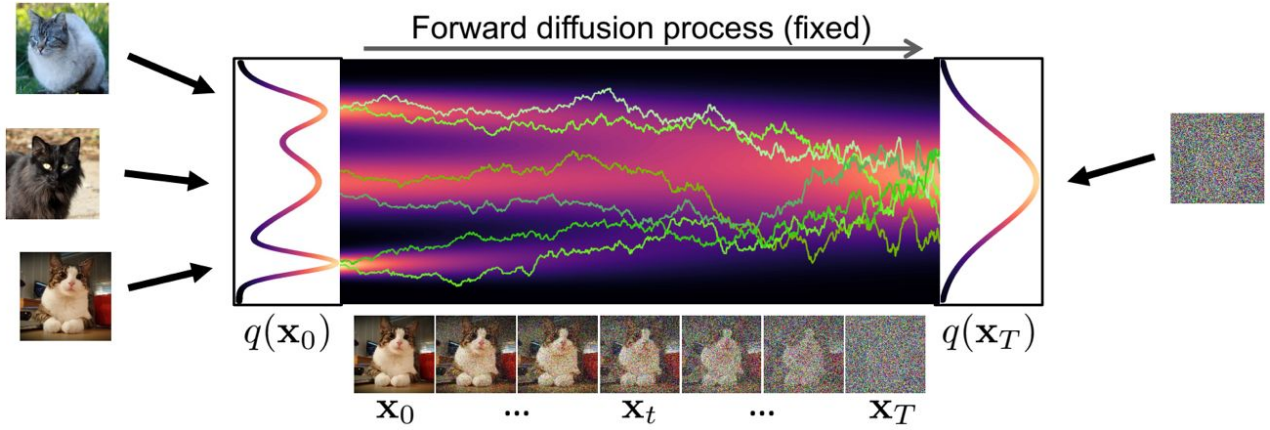
$$\mathbb{E}_{q(\mathbf{x}_0)}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] =: L$$

$$L = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), t \sim \mathcal{U}\{1,T\}, \epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})}\left[\lambda_t ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon, t)||^2\right]$$

Diffusion models



The Generative Reverse Stochastic Differential Equation



Forward Diffusion SDE:

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Reverse Generative Diffusion SDE:

$$\mathrm{d}\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]}^{\text{drift term}}\mathrm{d}t + \overbrace{\sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$

SOTA diffusion models

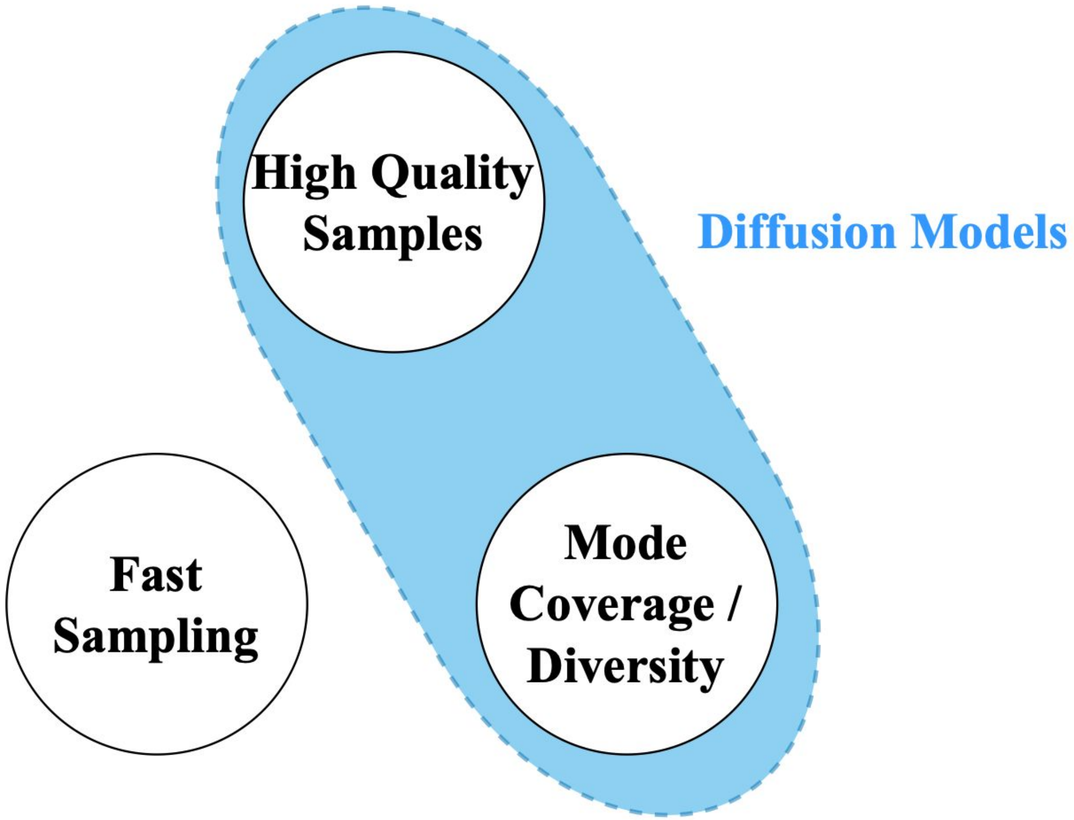
FLUX.1
A new era of creation

stability.ai
STABLE DIFFUSION

Veo 3

Generation examples

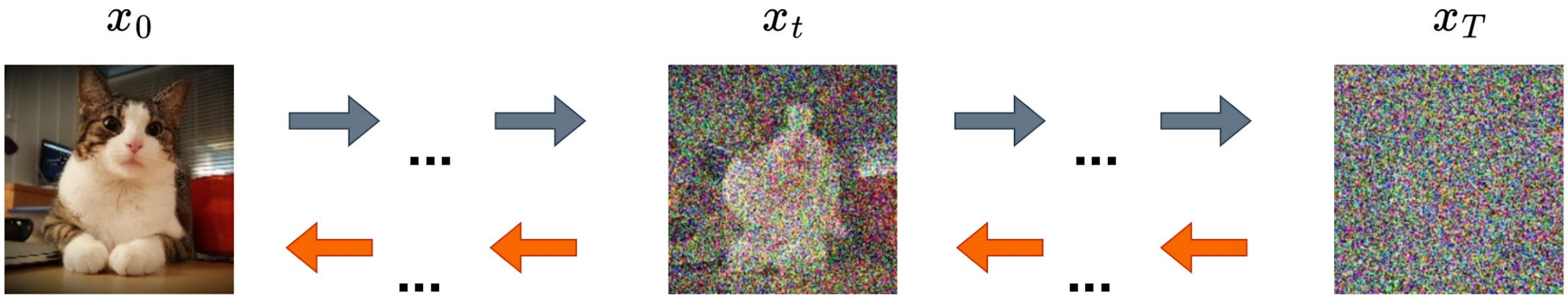
Generative Trilemma



Diffusion SDE

Forward SDE

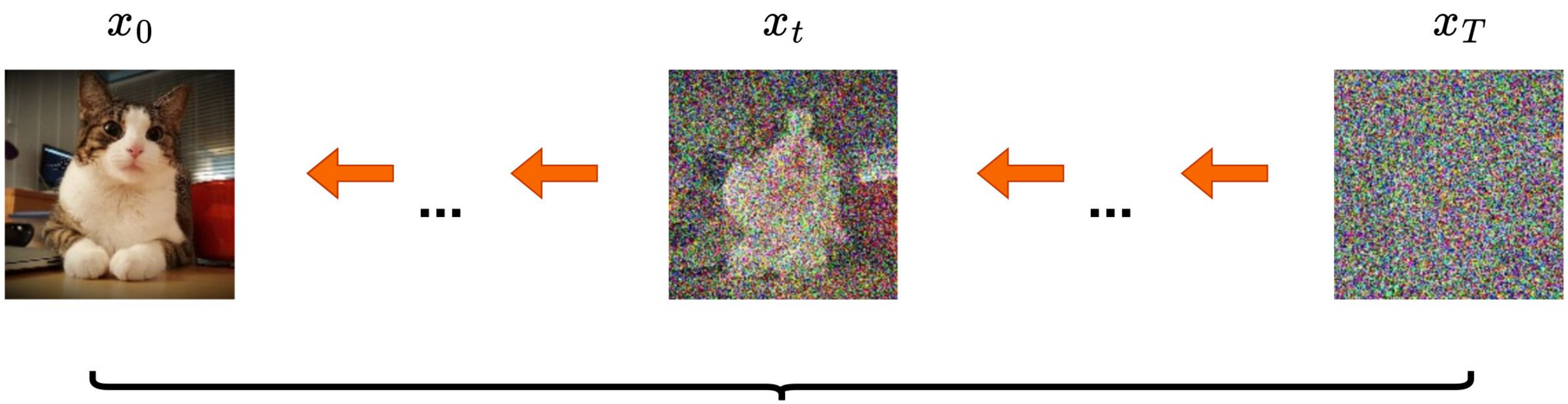
$$\mathrm{d}x = f(x,t)\mathrm{d}t + g(t)\mathrm{d}w$$



$$\mathrm{d}x = [f(x,t) - g(t)^2 \nabla_x \log p_t(x)]\mathrm{d}t + g(t)\mathrm{d}\bar{w}$$

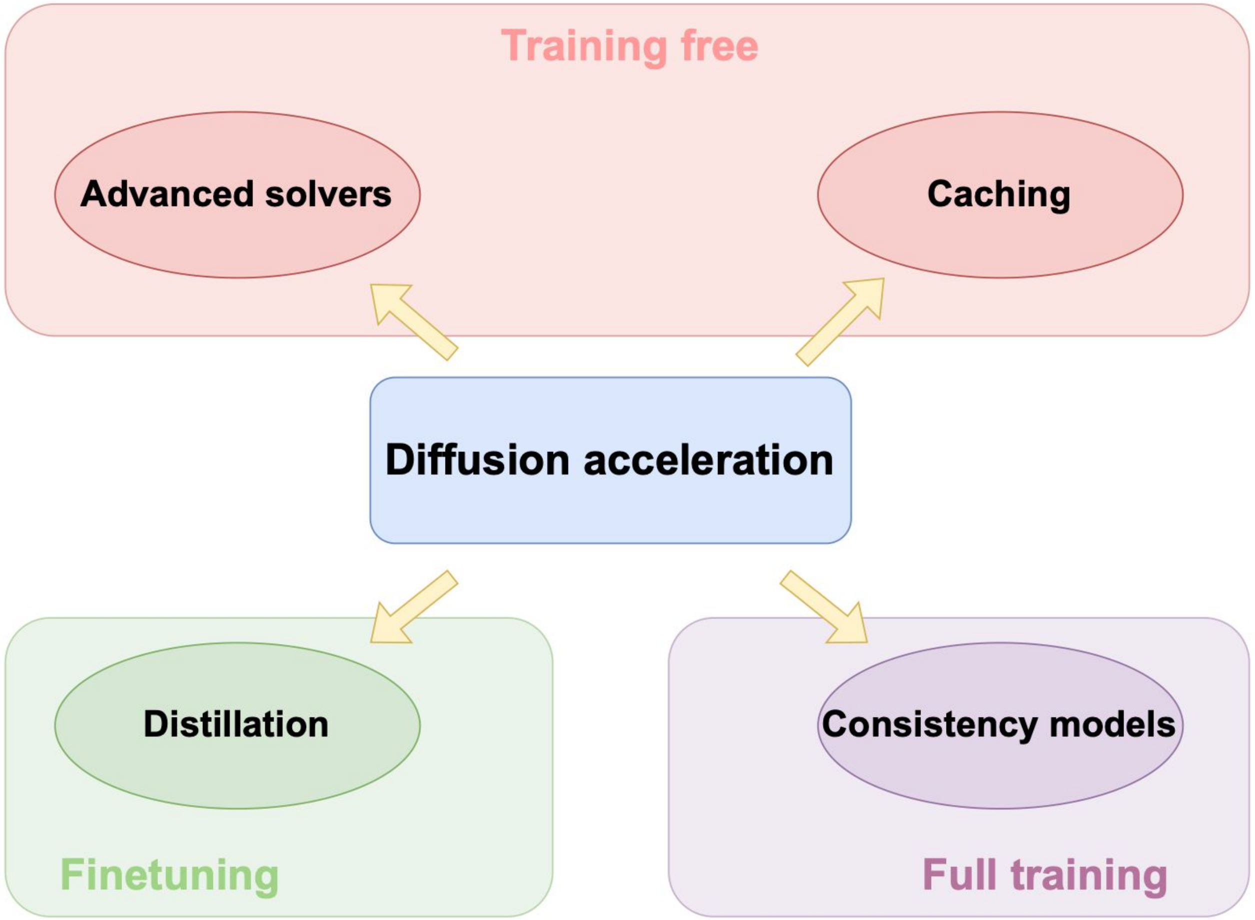
Reverse SDE

Diffusion sampling



~ 100+ steps

Diffusion acceleration approaches



Advanced solvers

Reverse SDE

$$\mathrm{d}x = [f(x,t) - g(t)^2 \nabla_x \log p_t(x)]\mathrm{d}t + g(t)\mathrm{d}\bar{w}$$

$$\Downarrow$$

$$\mathrm{d}x = [f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x)]\mathrm{d}t$$

Reverse ODE

Reverse ODE solution

Reverse ODE

$$\mathrm{d}x = \left[f(x,t) - \frac{1}{2}g(t)^2 \nabla_x \log p_t(x)\right]\mathrm{d}t$$

Diffusion ODE

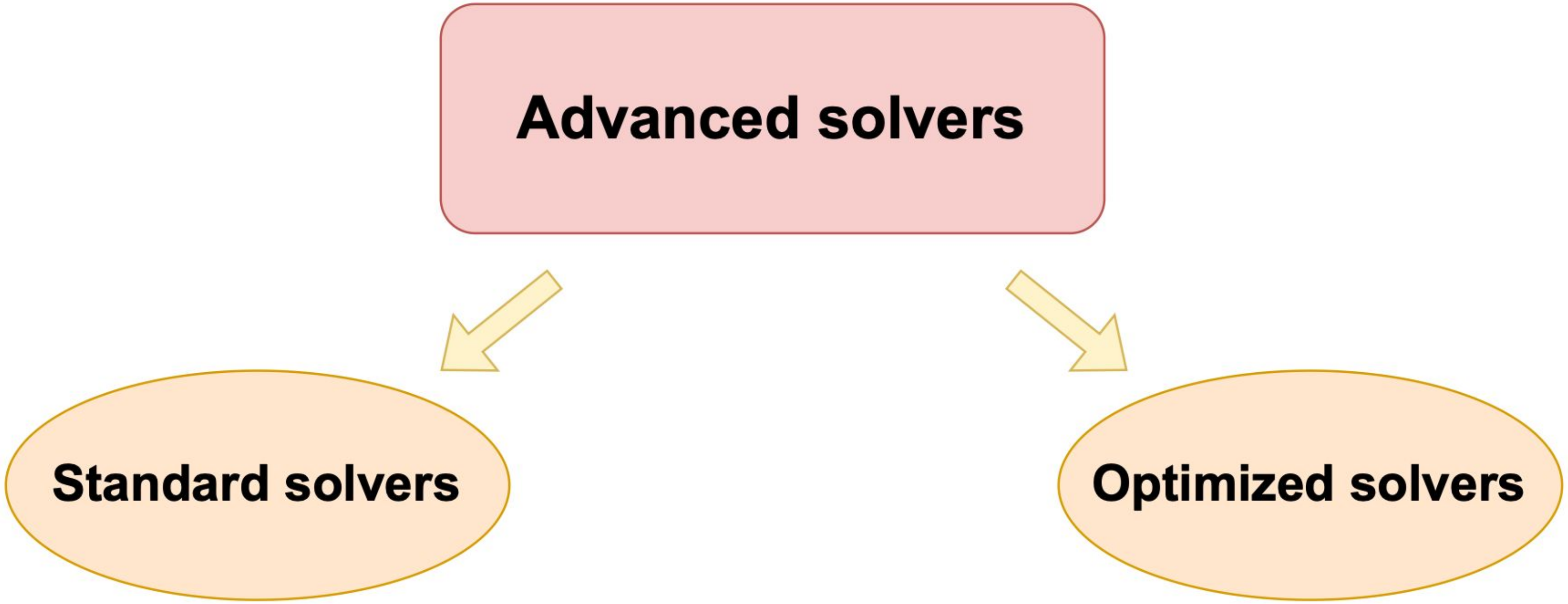
$$\mathrm{d}x = \left[f(x,t) - \frac{1}{2}g(t)^2 \boldsymbol{\varepsilon_\theta(x,t)}\right]\mathrm{d}t, \quad \varepsilon_\theta(x,t) \approx \nabla_x \log p_t(x)$$

Diffusion Solution

$$x_t = \frac{\alpha_t}{\alpha_s} \cdot x_s - \boxed{\int_s^t \frac{\alpha_t}{\alpha_\tau} \cdot \frac{g^2(\tau)}{2} \boldsymbol{\varepsilon_\theta(x_\tau,\tau)}\mathrm{d}\tau}$$

intractable integral!

Advanced solvers



Standard solvers

$$x_t = \frac{\alpha_t}{\alpha_s} \cdot x_s - \boxed{\int_s^t \frac{\alpha_t}{\alpha_\tau} \cdot \frac{g^2(\tau)}{2} \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_\tau, \boldsymbol{\tau}) \mathrm{d}\tau}$$

intractable integral!

After discretization:

$$x_{n+1} = \sum_{j=0}^{n} a_{j,n} x_j + \sum_{j=0}^{n} c_{j,n} \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_j, \boldsymbol{t}_j)$$

Coefficients $a_{j,n}, c_{j,n}$ are defined differently depending on solvers:

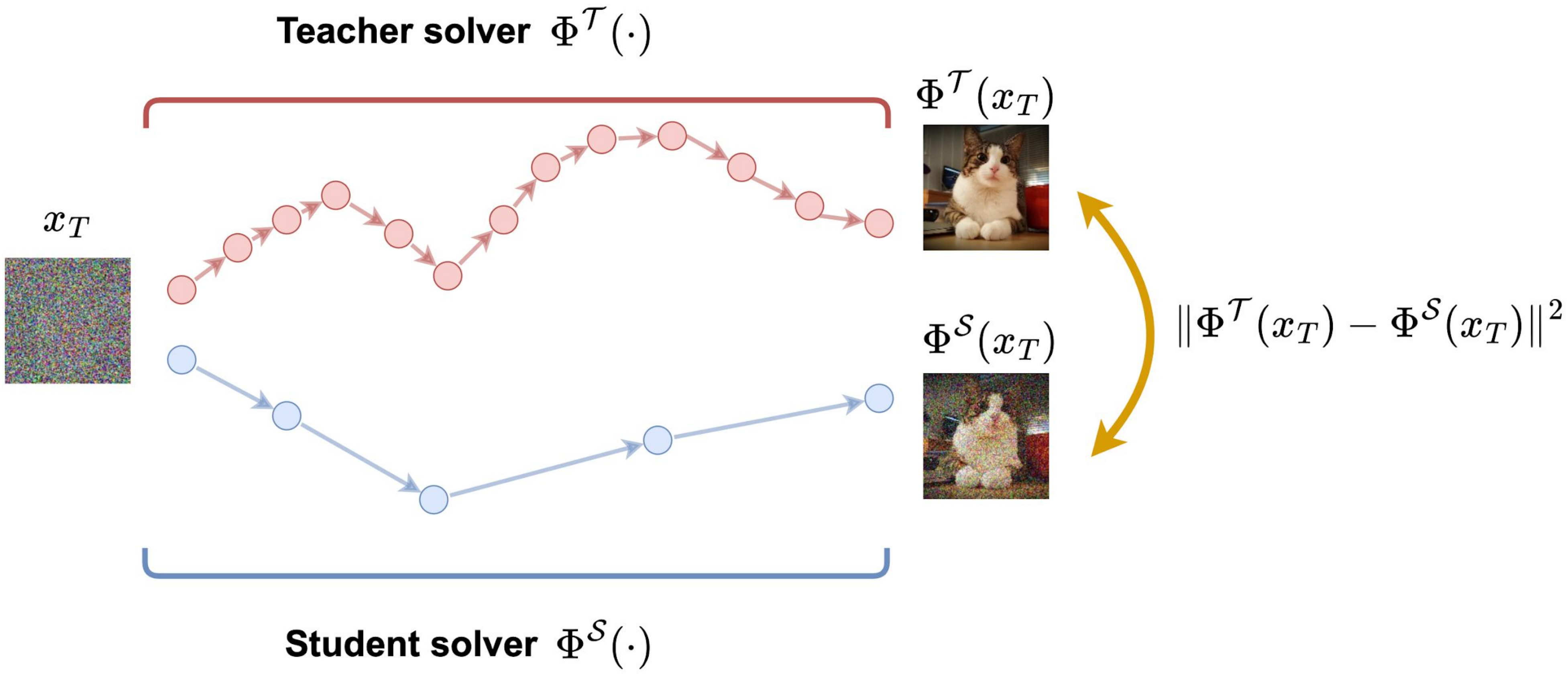
- **Euler/DDIM;**
- **DPM-Solver;**
- **DEIS-Solver;**
- **UniPC-Solver.**

Optimized solvers

Introduce trainable parameters for solver:

$$x_{n+1} = \sum_{j=0}^{n} a_{j,n}(\varphi)x_j + \sum_{j=0}^{n} c_{j,n}(\varphi)\boldsymbol{\varepsilon_\theta}(\boldsymbol{x_j}, \boldsymbol{t_j} + \boldsymbol{\psi_{n-j}})$$

Optimized solvers



Optimized solvers

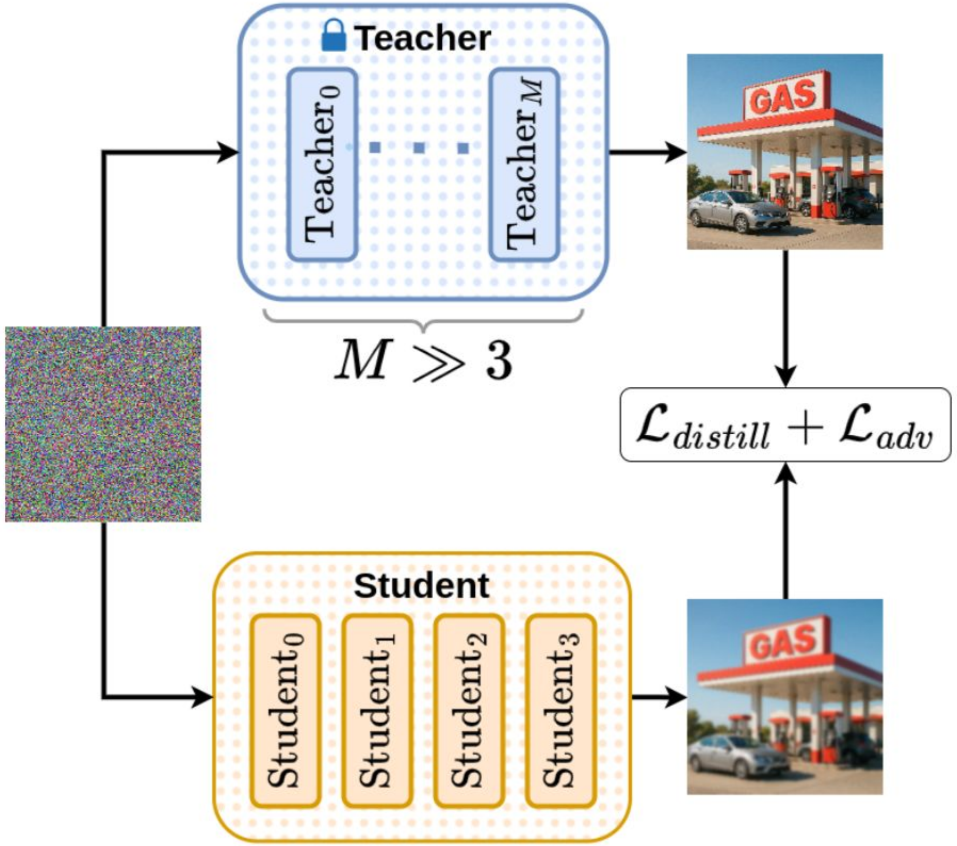
Student solver:

$$x_{n+1} = \sum_{j=0}^{n} a_{j,n}(\varphi) x_j + \sum_{j=0}^{n} c_{j,n}(\varphi) \boldsymbol{\varepsilon_\theta(x_j, t_j + \psi_{n-j})}$$

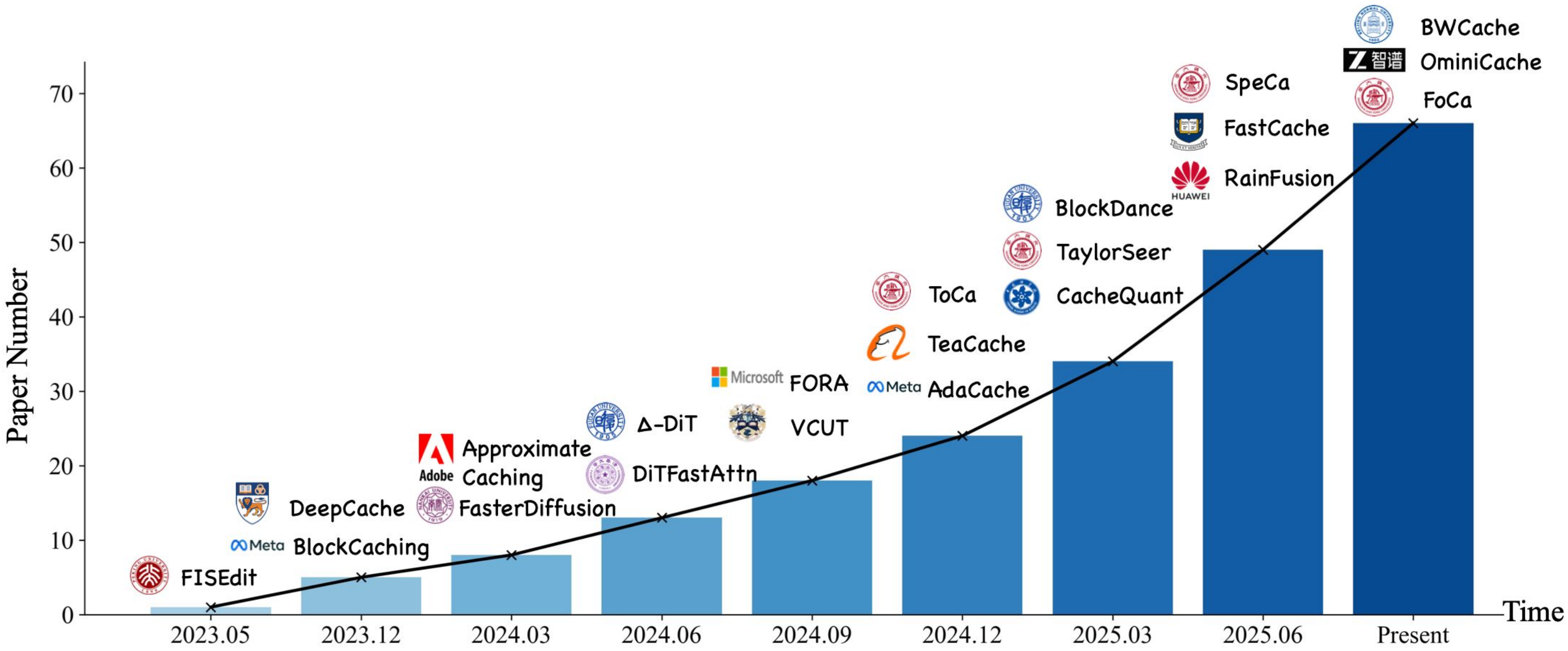
Solver optimization:

$$\mathbb{E}_{x_T} \|\Phi^{\mathcal{T}}(x_T) - \Phi^{\mathcal{S}}(x_T | \varphi, \psi)\|^2 \to \min \varphi, \psi, \quad x_T \sim \mathcal{N}(0, I)$$

GAS: Improving Discretization of Diffusion ODEs via Generalized Adversarial Solver

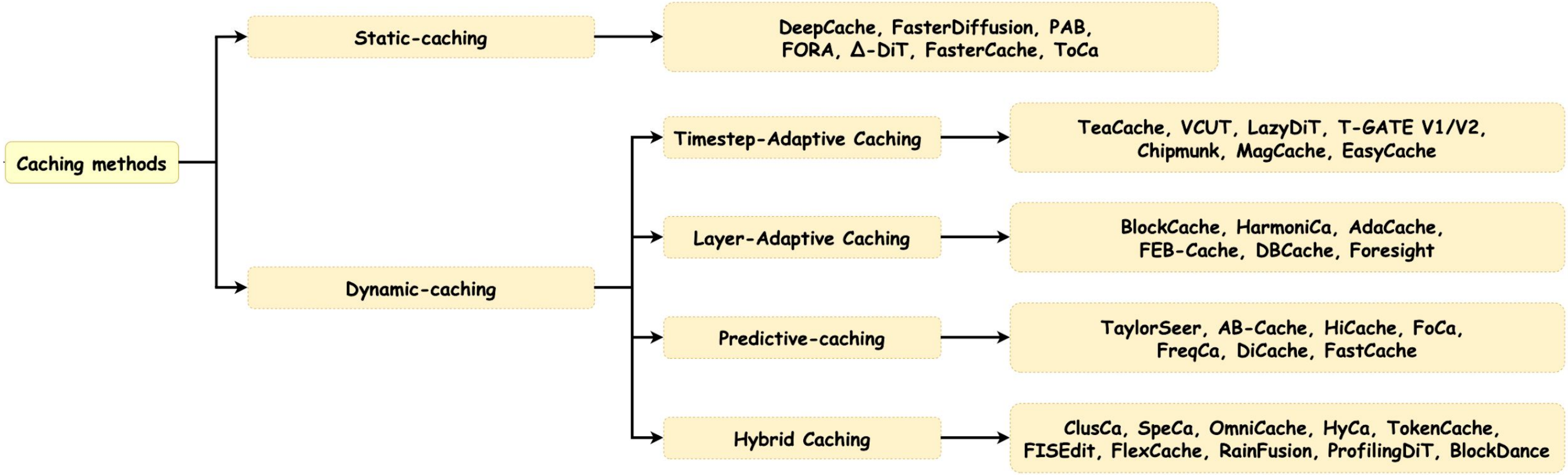


Caching

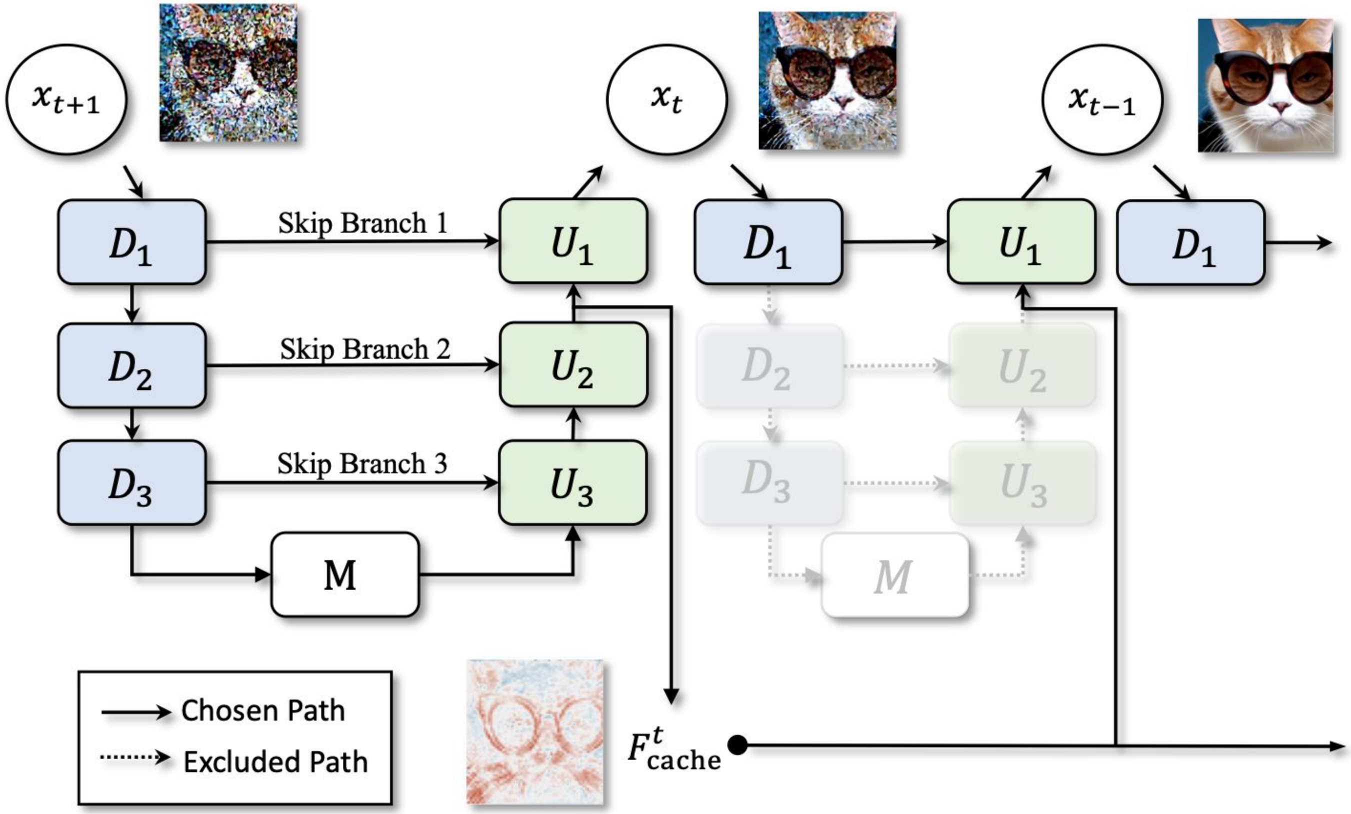


AIRI

Caching

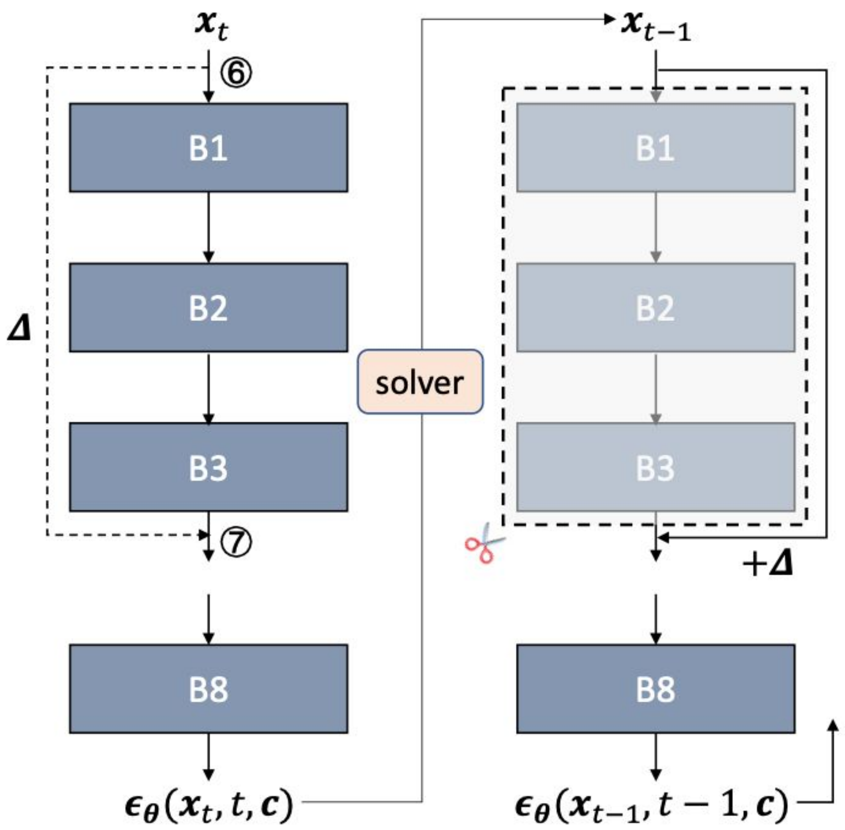

Caching methods
Static-caching
DeepCache, FasterDiffusion, PAB,
FORA, Δ-DiT, FasterCache, ToCa
Dynamic-caching
Timestep-Adaptive Caching
TeaCache, VCUT, LazyDiT, T-GATE V1/V2,
Chipmunk, MagCache, EasyCache
Layer-Adaptive Caching
BlockCache, HarmoniCa, AdaCache,
FEB-Cache, DBCache, Foresight
Predictive-caching
TaylorSeer, AB-Cache, HiCache, FoCa,
FreqCa, DiCache, FastCache
Hybrid Caching
ClusCa, SpeCa, OmniCache, HyCa, TokenCache,
FISEdit, FlexCache, RainFusion, ProfilingDiT, BlockDance

DeepCache



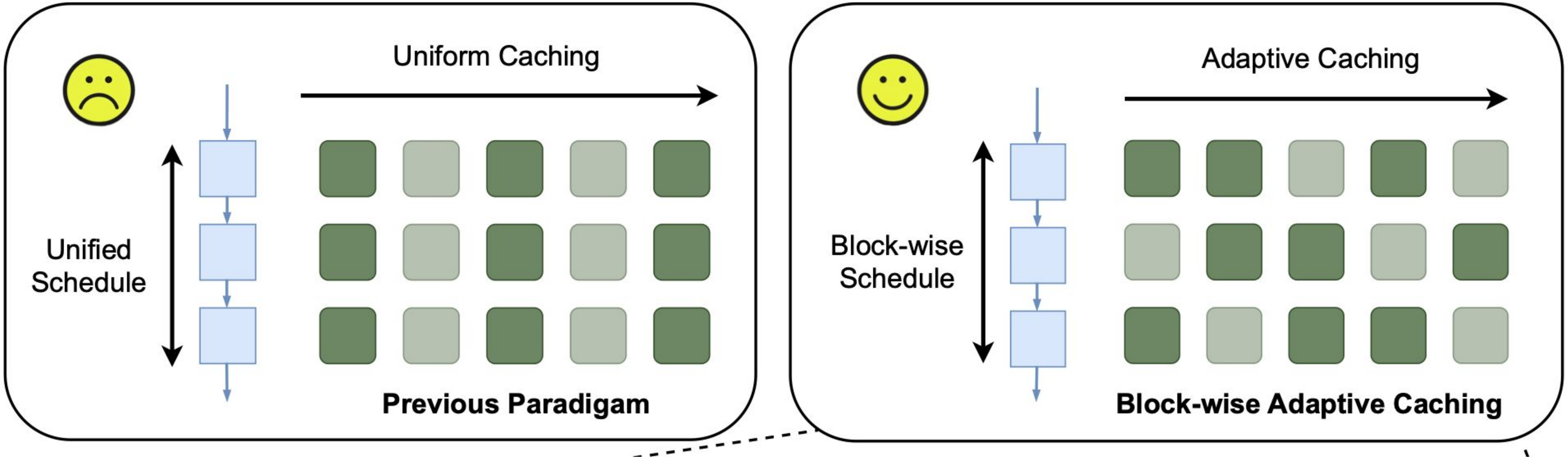
x_{t+1}
x_t
x_{t-1}
D_1
D_2
D_3
U_1
U_2
U_3
M
Skip Branch 1
Skip Branch 2
Skip Branch 3
D_1
D_2
D_3
U_1
U_2
U_3
M
D_1
Chosen Path
Excluded Path
F^t_{cache}

Delta-DiT

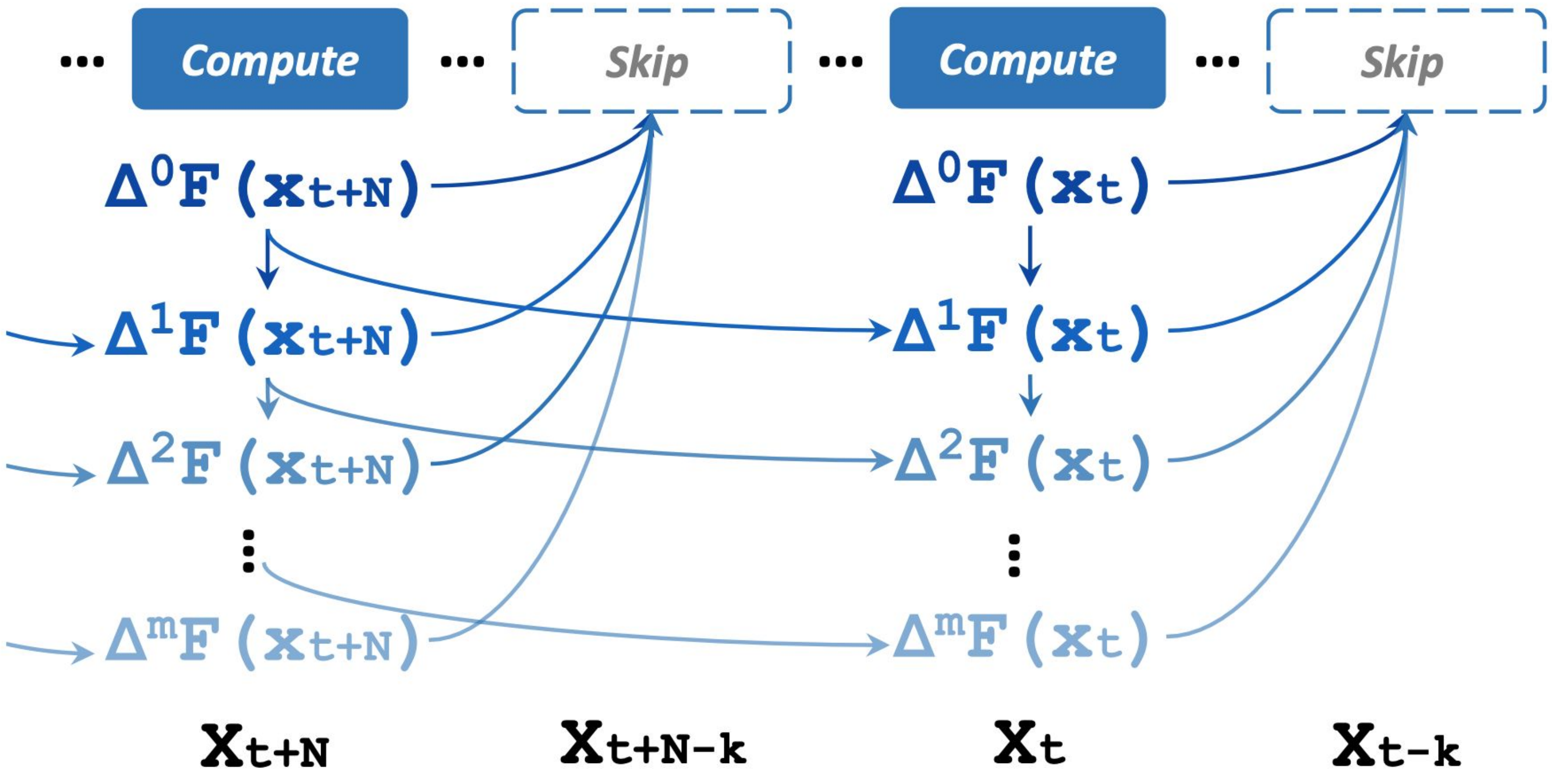


$\boldsymbol{x}_t$
⑥
B1
B2
B3
Δ
⑦
B8
$\epsilon_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$
solver
$\boldsymbol{x}_{t-1}$
B1
B2
B3
$+\Delta$
B8
$\epsilon_\theta(\boldsymbol{x}_{t-1}, t-1, \boldsymbol{c})$

BlockCache

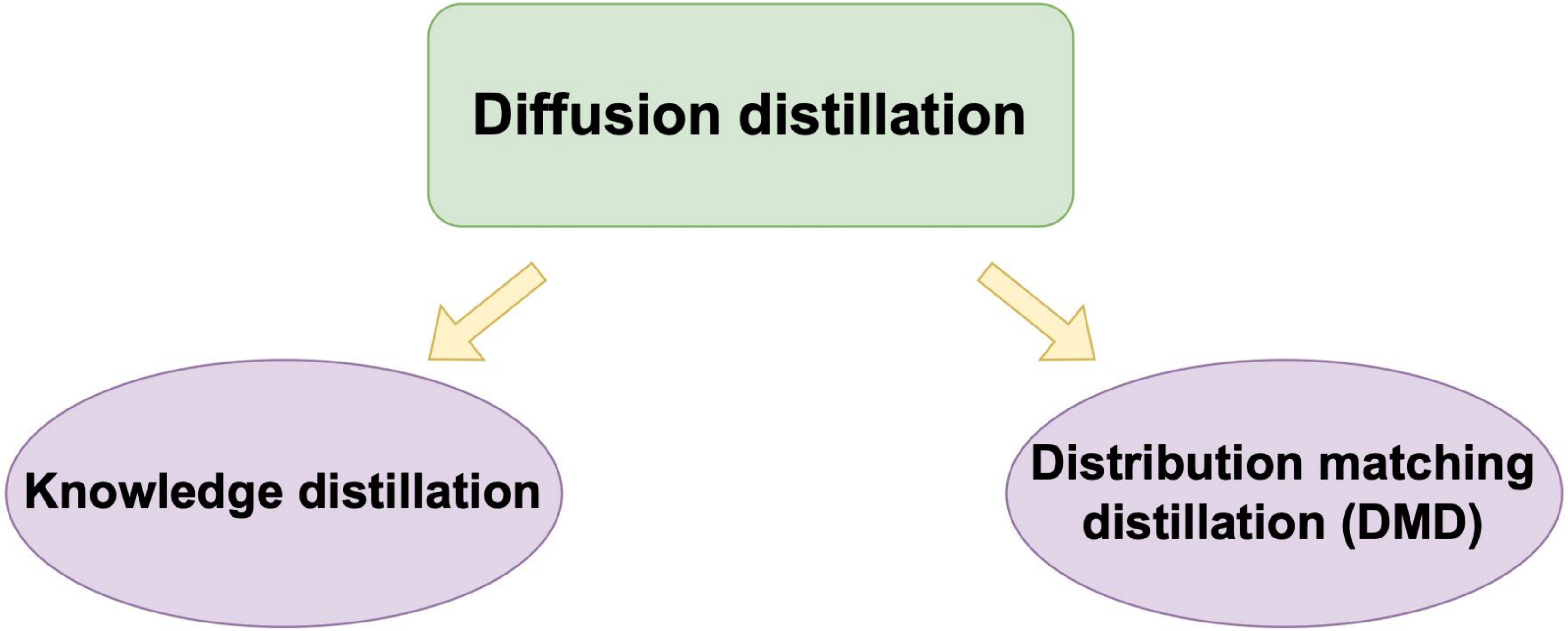

Uniform Caching
Unified
Schedule
Previous Paradigam
Adaptive Caching
Block-wise
Schedule
Block-wise Adaptive Caching

TaylorSeer

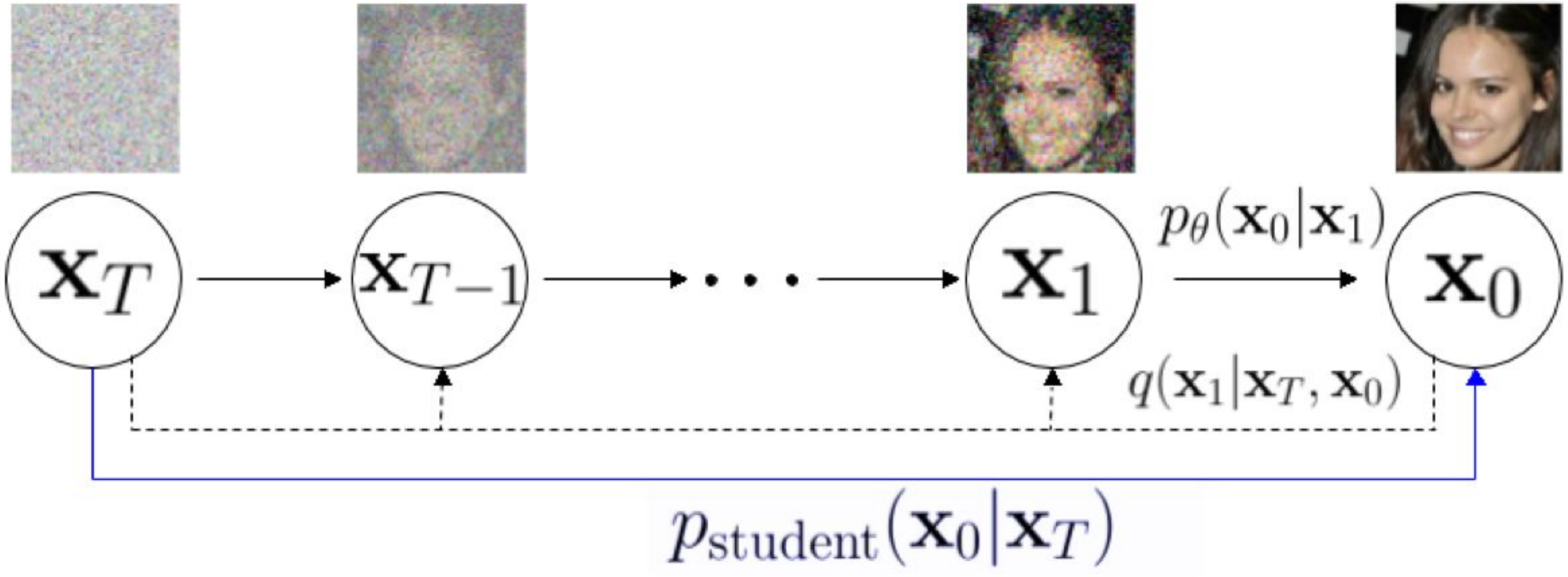


...
Compute
...
Skip
...
Compute
...
Skip
$\Delta^0 F(x_{t+N})$
$\Delta^1 F(x_{t+N})$
$\Delta^2 F(x_{t+N})$
$\Delta^m F(x_{t+N})$
$\Delta^0 F(x_t)$
$\Delta^1 F(x_t)$
$\Delta^2 F(x_t)$
$\Delta^m F(x_t)$
x_{t+N}
x_{t+N-k}
x_t
x_{t-k}

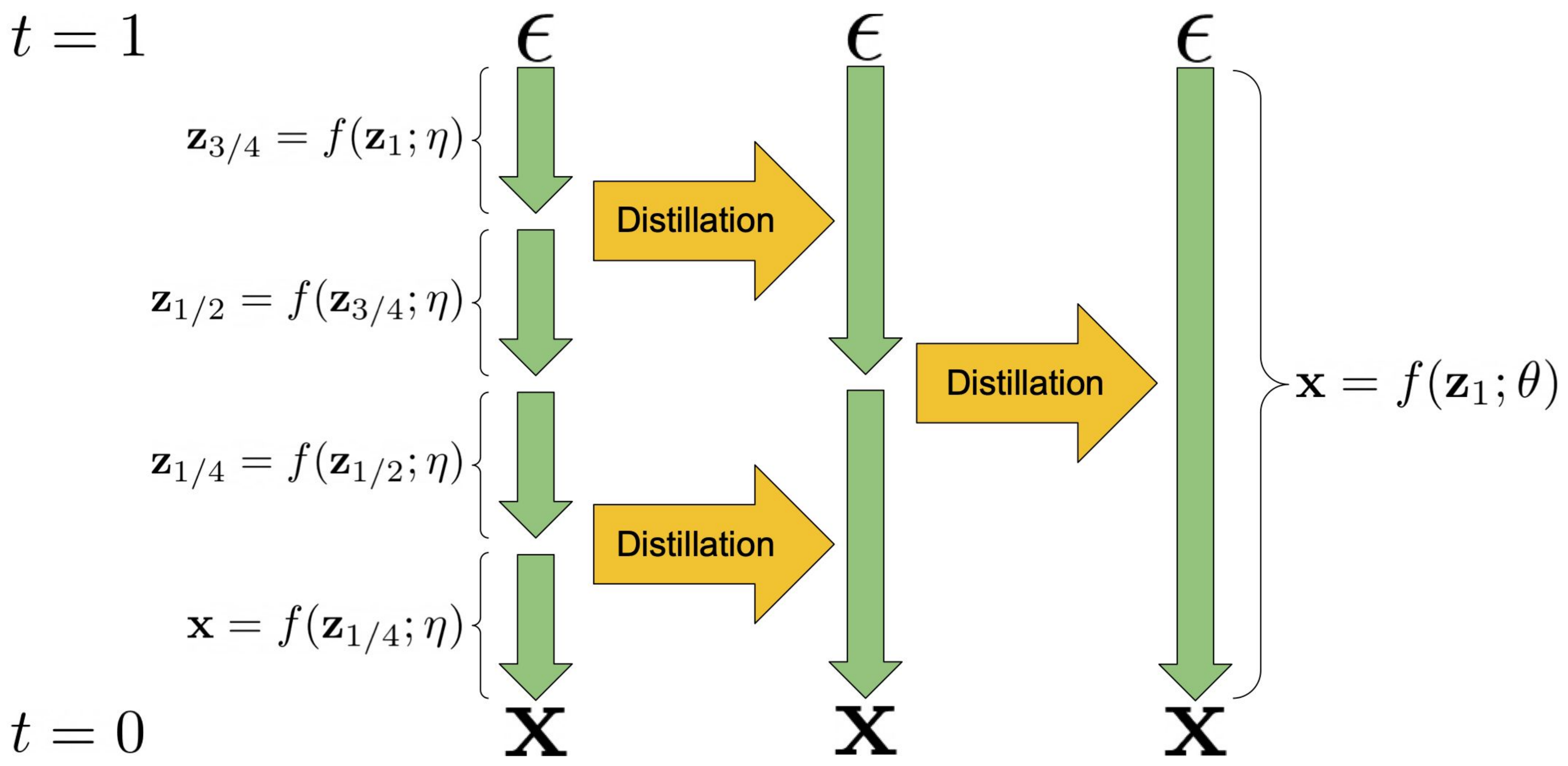
Diffusion distillation



Knowledge distillation



Progressive distillation



Distribution matching distillation (DMD)

We want to train a one-step generator $G_\theta(z)$**:**

$$D_{KL}\left(p_{\text{fake}} \parallel p_{\text{real}}\right) = \mathop{\mathbb{E}}_{x \sim p_{\text{fake}}} \left(\log \left(\frac{p_{\text{fake}}(x)}{p_{\text{real}}(x)} \right) \right)$$

$$= \mathop{\mathbb{E}}_{\substack{z \sim \mathcal{N}(0;\mathbf{I}) \\ x = G_\theta(z)}} -\left(\log \; p_{\text{real}}(x) - \log \; p_{\text{fake}}(x) \right)$$

Distribution matching distillation (DMD)

$$D_{KL}\left(p_{\text{fake}} \parallel p_{\text{real}}\right) = \mathop{\mathbb{E}}_{x\sim p_{\text{fake}}}\left(\log\left(\frac{p_{\text{fake}}(x)}{p_{\text{real}}(x)}\right)\right)$$
$$= \mathop{\mathbb{E}}_{\substack{z\sim\mathcal{N}(0;\mathbf{I})\\ x=G_\theta(z)}} -\left(\log\ p_{\text{real}}(x) - \log\ p_{\text{fake}}(x)\right)$$

$$\nabla_\theta D_{KL} = \mathop{\mathbb{E}}_{\substack{z\sim\mathcal{N}(0;\mathbf{I})\\ x=G_\theta(z)}}\left[-\left(s_{\text{real}}(x) - s_{\text{fake}}(x)\right)\frac{dG}{d\theta}\right],$$

where $s_{\text{real}}(x) = \nabla_x \log\ p_{\text{real}}(x)$, $s_{\text{fake}}(x) = \nabla_x \log\ p_{\text{fake}}(x)$

Distribution matching distillation (DMD)

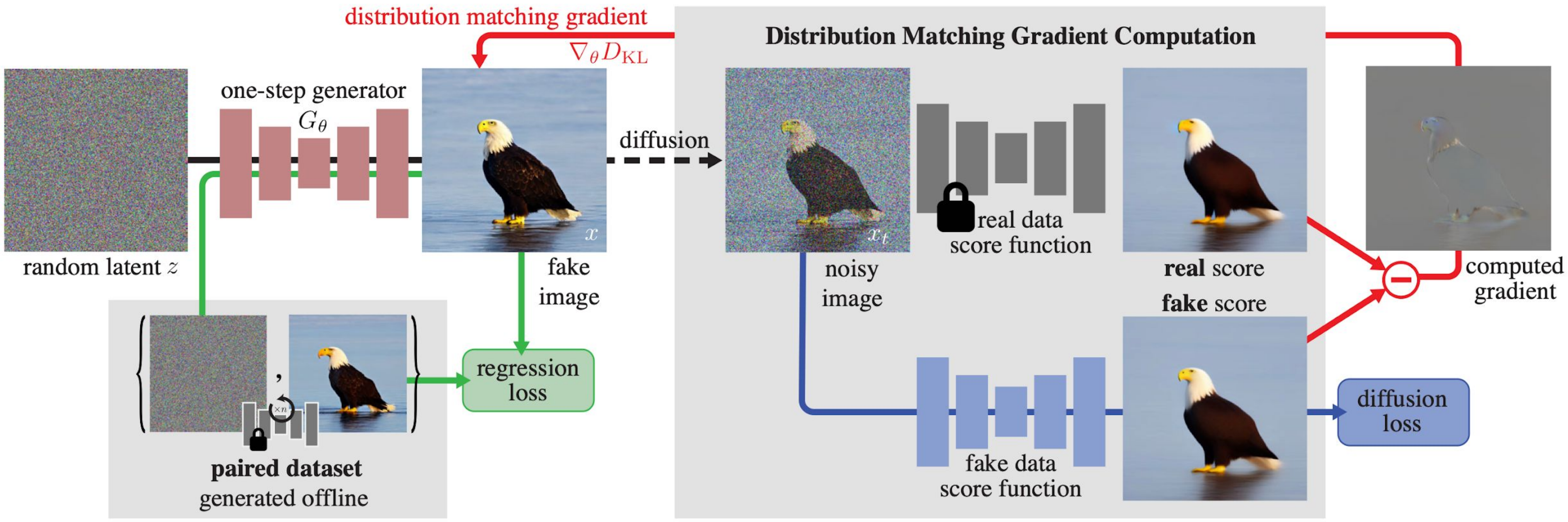
DMD algorithm:

1. Generator update: $$\nabla_\theta D_{KL} \simeq \mathop{\mathbb{E}}_{z,t,x,x_t}\left[w_t\alpha_t\left(s_{\text{fake}}(x_t,t) - s_{\text{real}}(x_t,t)\right)\frac{dG}{d\theta}\right]$$

2. Fake score update: $$\nabla_\varphi \mathbb{E}_{t,x,x_t}\|s^\varphi_{fake}(x_t,t) - \nabla_{x_t}\log q(x_t|x)\|^2$$

3. Go to 1

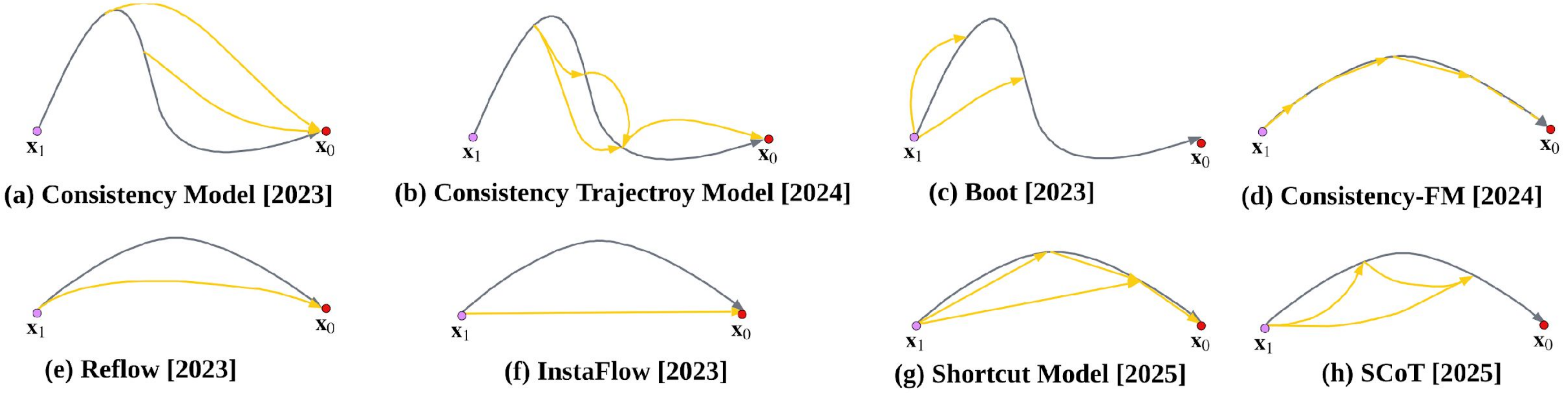
Distribution matching distillation (DMD)



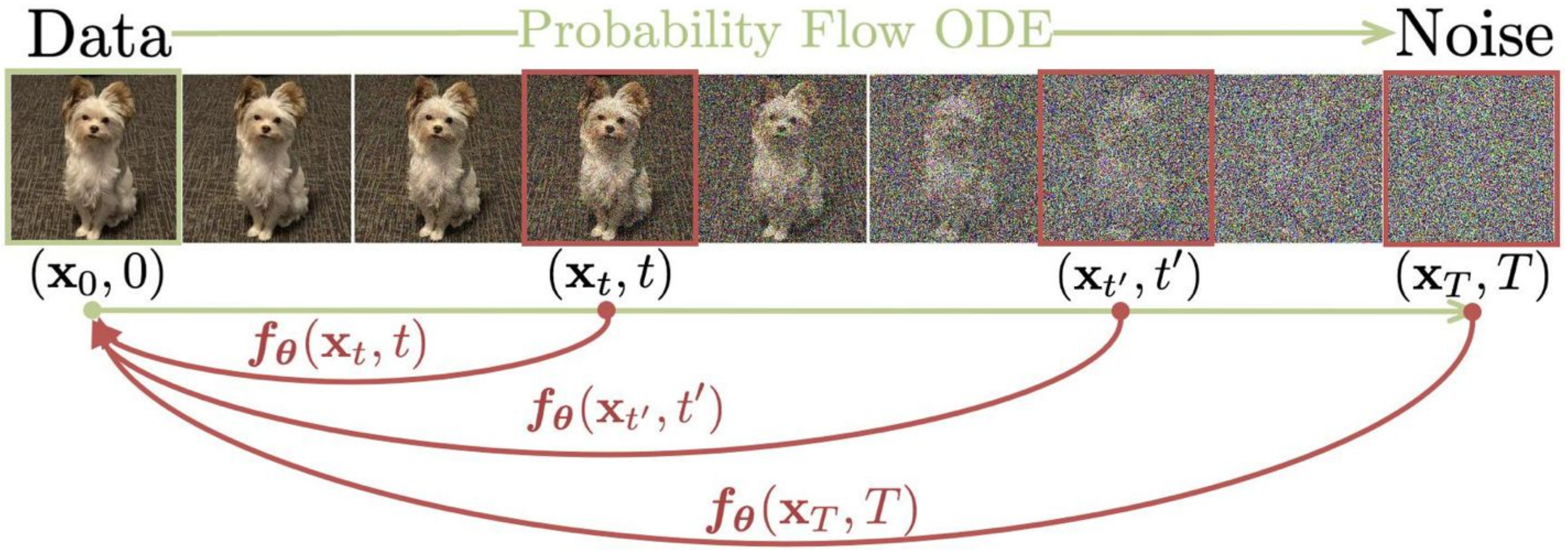
Further developments of DMD

- **DMD2**
- **Adversarial Diffusion Distillation (ADD)**
- **Latent Adversarial Diffusion Distillation (LADD)**

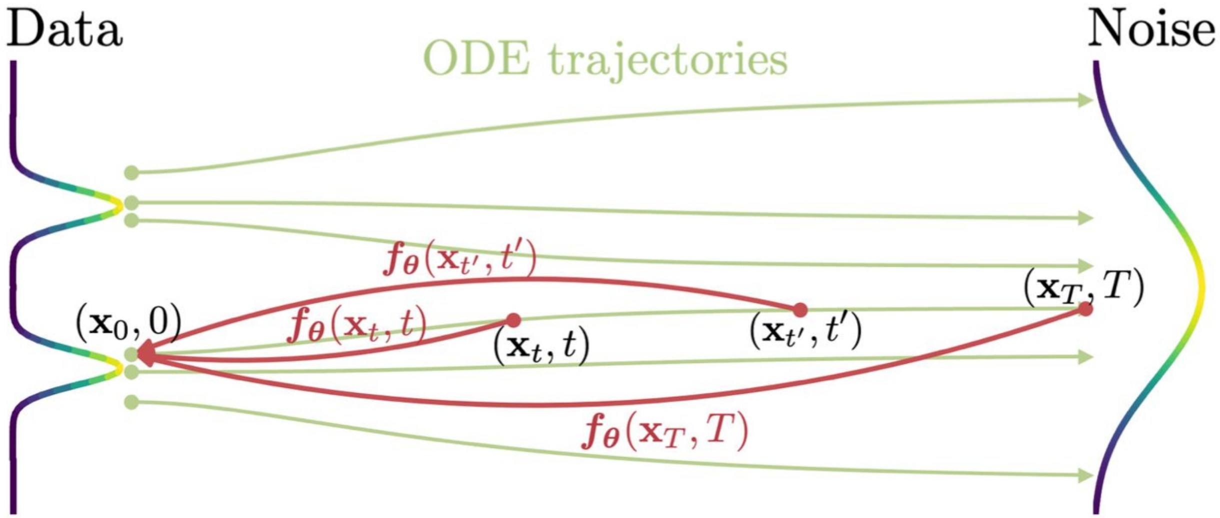
Consistency models



Consistency model

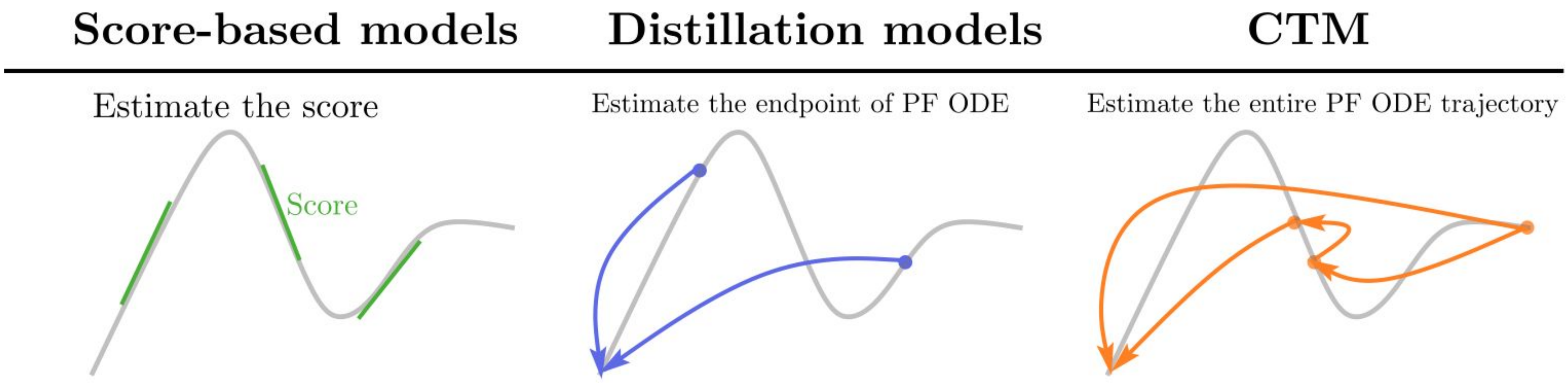


Consistency model

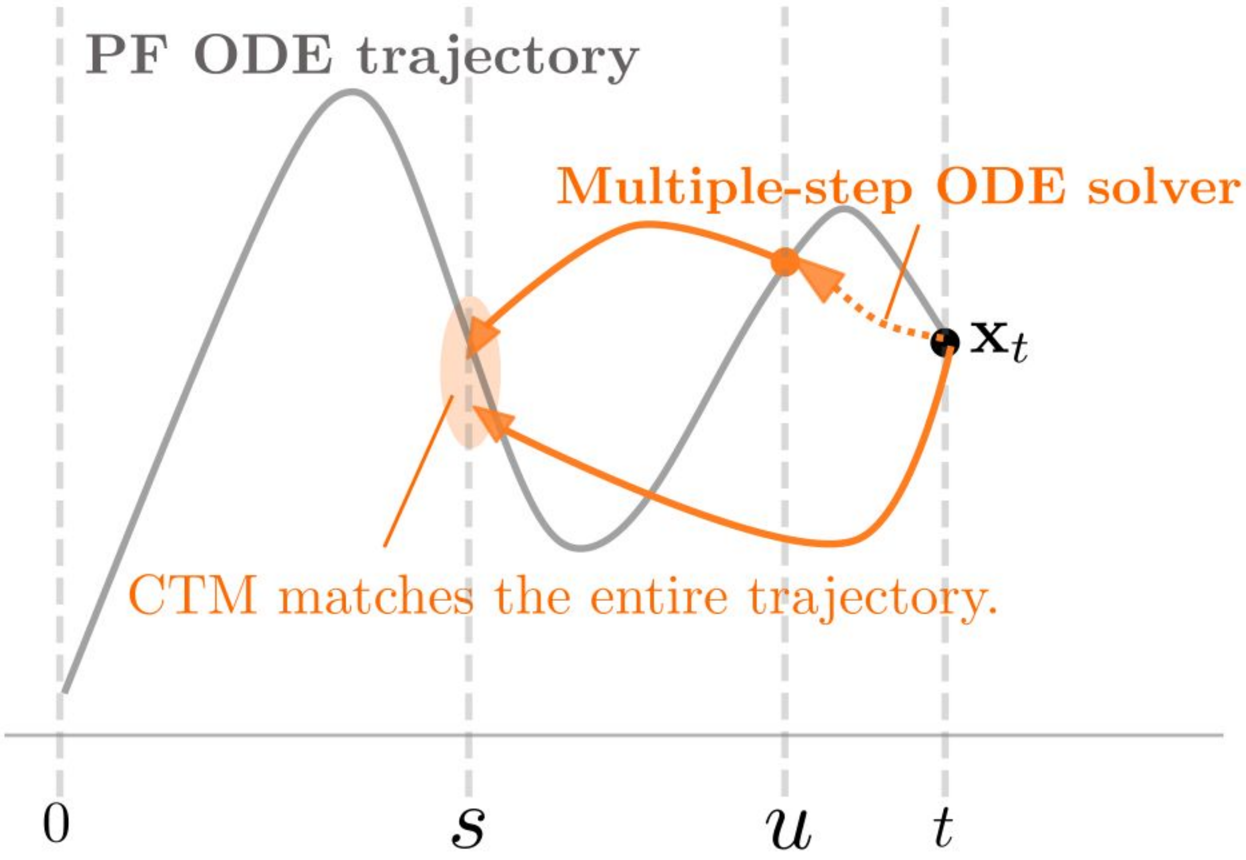


$$\mathcal{L}_{CD}^{N}(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}; \boldsymbol{\phi}) := \mathbb{E}[\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^{-}}(\hat{\mathbf{x}}_{t_n}^{\phi}, t_n))]$$

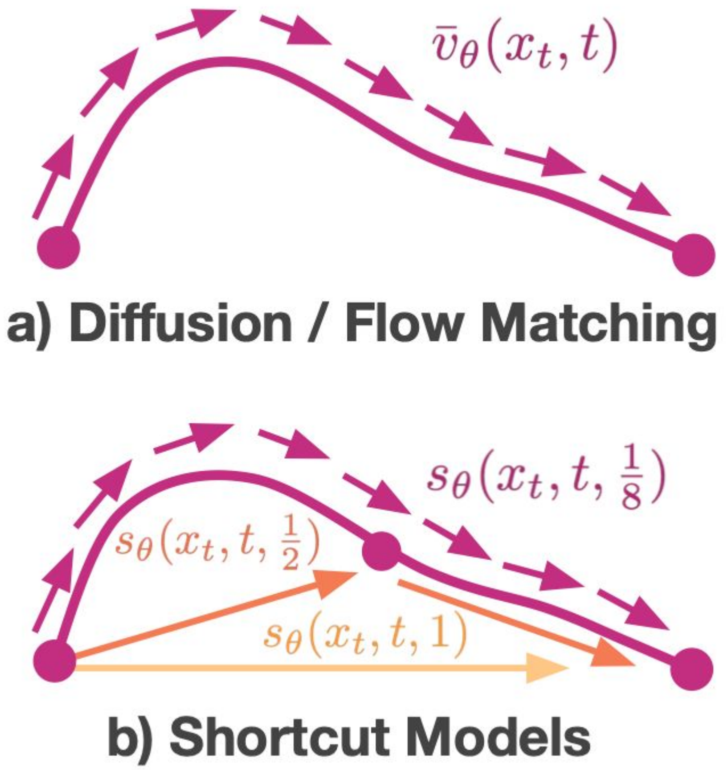
Consistency trajectory models



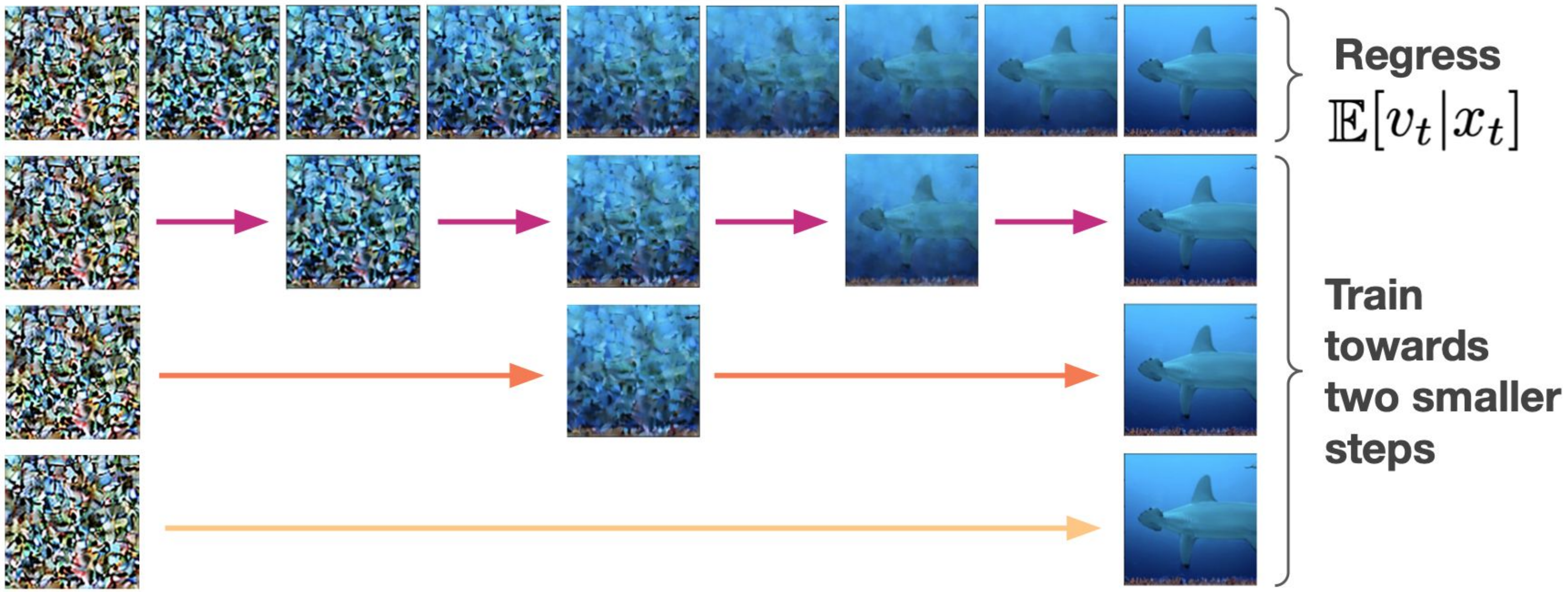
Consistency trajectory models



Shortcut models



Shortcut models



Shortcut models

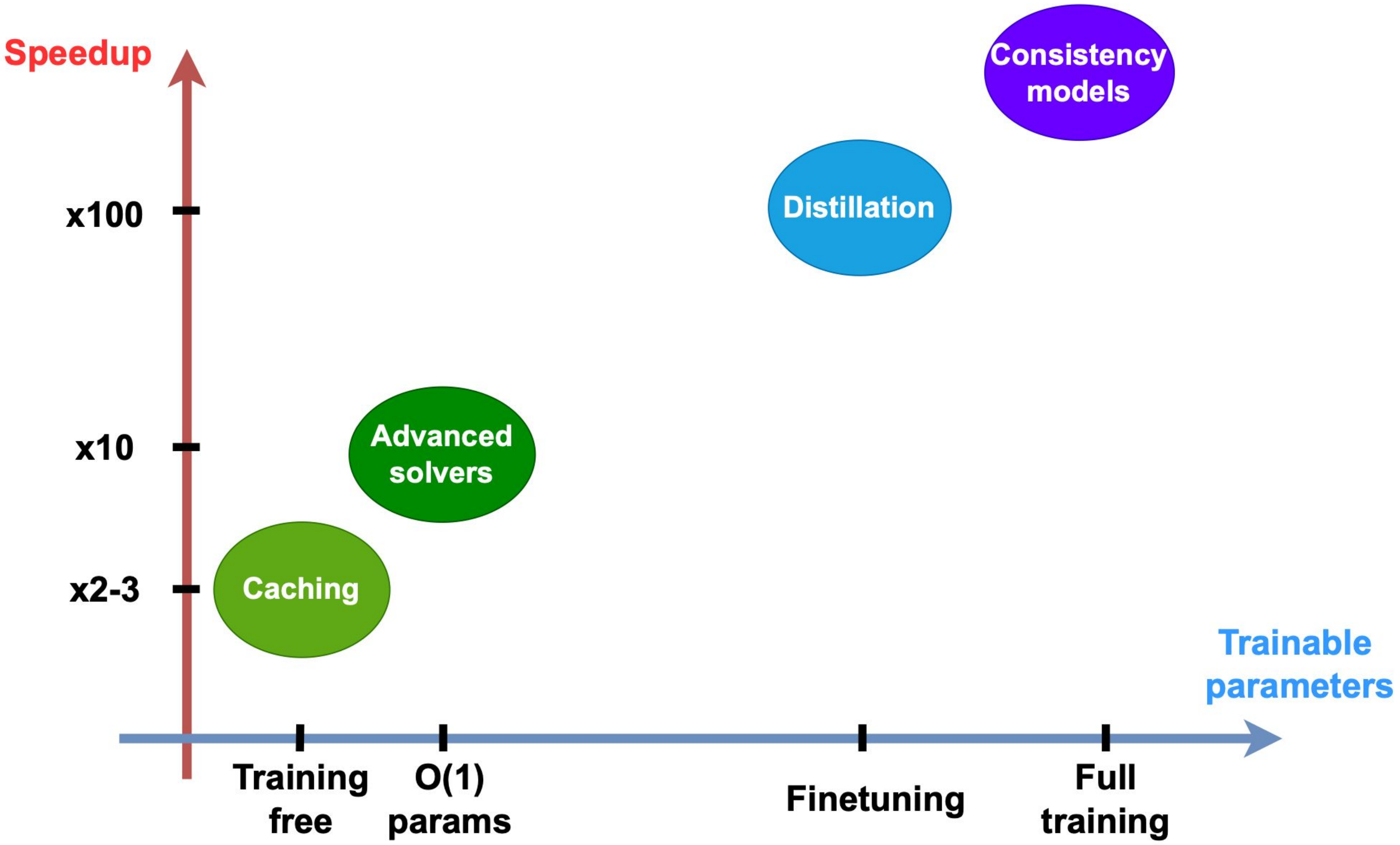
$$\mathcal{L}^{\mathrm{S}}(\theta) = E_{x_0 \sim \mathcal{N},\, x_1 \sim D,\, (t,d) \sim p(t,d)} \Big[\underbrace{\| s_\theta(x_t, t, 0) - (x_1 - x_0) \|^2}_{\text{Flow-Matching}} + \underbrace{\| s_\theta(x_t, t, 2d) - s_{\text{target}} \|^2}_{\text{Self-Consistency}} \Big],$$

$$\text{where} \quad s_{\text{target}} = s_\theta(x_t, t, d)/2 + s_\theta(x'_{t+d}, t, d)/2 \quad \text{and} \quad x'_{t+d} = x_t + s_\theta(x_t, t, d)d.$$

Further developments of CM

- **Consistency FM**
- **MeanFlow**
- **Aligh Your Flow**

Conclusions



Contacts

Head of the Center of Deep Learning and Bayesian Methods HSE University

Lead of Controllable Generative AI at AIRI

Email: **alanov.aibek@gmail.com**

@AIBRAIN