

ART for Diffusion Sampling: A Reinforcement Learning Approach to Timestep Scheduling

Xunyu Zhou

(Joint with Yilie Huang and Wenpin Tang)

Columbia University

May 2026, *Foundations of Multi-Agent and Mean Field RL*

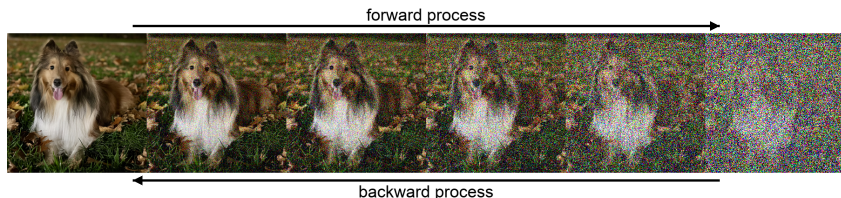
Outline

- 1 Diffusion models for GenAI and sampling efficiency
- 2 ART: learn a sampling clock
- 3 Theory: solving ART via ART-RL
- 4 Algorithm: actor-critic
- 5 Experiments: drop-in schedule replacement
- 6 Takeaways

- 1 Diffusion models for GenAI and sampling efficiency
- 2 ART: learn a sampling clock
- 3 Theory: solving ART via ART-RL
- 4 Algorithm: actor-critic
- 5 Experiments: drop-in schedule replacement
- 6 Takeaways

Diffusion models

- **Generative AI (GenAI)**: generate new samples similar to given samples from an *unknown* distribution.
- **Forward process**: gradually adds noise and transforms training samples into a simple distribution e.g. Gaussian (Ho et al., 2020; Song and Ermon, 2019; Song et al., 2021b).
- **Backward process**: starts from noise and iteratively denoises to generate new samples.



Forward blurring

- p_0 : an unknown target distribution in $\mathbb{R}^d$.
- Forward OU process:

$$d\bar{x}(\tau) = -f(\tau)\bar{x}(\tau) d\tau + g(\tau) dw(\tau), \quad \bar{x}(0) \sim p_0.$$

where w is d -dimensional Brownian motion independent of $\bar{x}(0)$, and both $f > 0$ and $g > 0$ are given and known.

- $\bar{x}(\tau) = e^{-\int_0^\tau f(s)ds} \bar{x}(0) + \int_0^\tau e^{-\int_s^\tau f(v)dv} g(s) dw(s), \quad \tau \in [0, T].$
- $p_\tau(\cdot)$: density function of $\bar{x}(\tau)$.
- $p_{\tau|0}(\cdot|\bar{x}(0))$: density of $\bar{x}(\tau)$ given $\bar{x}(0)$, which is Gaussian.

Backward denoising

- Backward process $\tilde{x}(\tau) := \bar{x}(T - \tau)$, $\tau \in [0, T]$.
- Under mild assumptions, it satisfies SDE (Anderson 1982, Haussmann and Pardoux 1986):

$$d\tilde{x}(\tau) = [f(T - \tau)\tilde{x}(\tau) + g(T - \tau)^2 \nabla_x \log p_{T-\tau}(\tilde{x}(\tau))]d\tau + g(T - \tau)dB(\tau),$$

where B is a d -dimensional Brownian motion.

- $\tilde{x}(0) \sim \nu := \mathcal{N}\left(0, \int_0^T e^{-2\int_t^T f(s)ds} (g(t))^2 dt \cdot I_d\right)$ ($\approx p_T \sim \bar{x}(T)$).
- $\tilde{x}(T) \sim p_0$, where p_0 is the target distribution we want to generate samples from.

Score matching

- *Score function* which is unknown

$$S(\tau, x) = \nabla_x \log p_\tau(x).$$

- Estimate it with a deep neural network s_θ , by

$$\min_{\theta} \mathbb{E}_{\tau \sim U[0, T]} \left[\lambda(\tau) \mathbb{E}_{\bar{x}(\tau)} \|s_\theta(\tau, \bar{x}(\tau)) - \nabla_x \log p_\tau(\bar{x}(\tau))\|^2 \right].$$

- Equivalent to *denoising score matching* (Vincent 2011, Song et al. 2021)

$$\min_{\theta} \mathbb{E}_{\tau \sim U[0, T]} \left[\lambda(\tau) \mathbb{E}_{\bar{x}(0)} \mathbb{E}_{\bar{x}(\tau) | \bar{x}(0)} \|s_\theta(\tau, \bar{x}(\tau)) - \nabla_x \log p_{\tau|0}(\bar{x}(\tau) | \bar{x}(0))\|^2 \right].$$

which is tractable as $p_{\tau|0}(\cdot | \bar{x}(0))$ is Gaussian.

- Loss function above can be approximated by Monte Carlo and minimized by SGD.

Probability flow ODE

- Song et al. (2021b) show that there exists an ODE:

$$\frac{d\tilde{x}(\tau)}{d\tau} = f(T - \tau)\tilde{x}(\tau) + \frac{1}{2}g(T - \tau)^2 S(T - \tau, \tilde{x}(\tau)), \quad \tilde{x}(0) \sim \nu$$

whose solution at time $\tau \in [0, T]$ is distributed according to $p_{T-\tau}$.

- In particular, $\tilde{x}(T) \sim p_0$.
- This ODE is called the *probability flow ODE*.

Sampling efficiency

- Diffusion models have become a central generative modeling paradigm:
 - ▶ image generation systems such as DALL-E 2 and Stable Diffusion (Ramesh et al., 2022; Rombach et al., 2022);
 - ▶ increasingly, video and language generation (OpenAI, 2024; Singer et al., 2023; Google, 2024; Khanna et al., 2025; Nie et al., 2025).
- High sample quality comes from the backward (reverse) process.
- Therefore, **sampling (inference) efficiency** directly controls **deployment cost**:
 - ▶ fewer network evaluations for each generated sample;
 - ▶ lower discretization errors;
 - ▶ lower computing cost at scale.

Sampling means choosing a time grid

- The backward process is governed by an SDE or ODE.
- Implementation needs to discretize the continuous trajectory:

$$0 = \tau_0 < \tau_1 < \cdots < \tau_K = T.$$

- Once K is fixed, the key numerical decision is how to place these τ_i 's.
- Strategic allocation:
 - ▶ smaller steps where the reverse dynamics are “delicate”;
 - ▶ big leaps where the reverse dynamics are “easy”.

Existing schedules

- **Uniform**: simple and easy to implement.
- **Hand-crafted schedules**: analytic prescriptions, but fixed once chosen (Song et al., 2021a; Karras et al., 2022; Lu et al., 2022; Chen et al., 2023).
 - ▶ EDM: a strong default schedule in the official EDM sampling pipeline (Karras et al., 2022).
 - ▶ DPM-Solver grid: a uniform log-SNR grid used as another strong baseline (Lu et al., 2022).
- These schedules are powerful baselines, but they do not learn the allocation *adaptively*.
- Question: Can we learn an “optimal” time grid?

Main contributions

- **Formulation**: formulate timestep allocation as Adaptive Reparameterized Time (ART), a control problem for the sampling clock.
- **Theory**: connect ART to a reinforcement learning problem, called ART-RL, and characterize policy evaluation, improvement, and gradient updates.
- **Algorithm**: develop ART-RL with Gaussian policies and actor-critic updates, building on continuous-time RL (Wang et al., 2020; Jia and Zhou, 2022a,b).
- **Evidence**: improve Uniform, DPM-Solver, and EDM schedules at matched budgets.

- 1 Diffusion models for GenAI and sampling efficiency
- 2 ART: learn a sampling clock
- 3 Theory: solving ART via ART-RL
- 4 Algorithm: actor-critic
- 5 Experiments: drop-in schedule replacement
- 6 Takeaways

Continuous-time score-based sampling

Forward diffusion

$$d\bar{x}(\tau) = -f(\tau)\bar{x}(\tau) d\tau + g(\tau) dw(\tau), \quad \bar{x}(0) \sim p_0.$$

Let p_τ be the law of $\bar{x}(\tau)$, and let

$$S(\tau, x) = \nabla_x \log p_\tau(x)$$

be the score.

Backward probability flow ODE

$$\frac{d\tilde{x}(\tau)}{d\tau} = f(T - \tau)\tilde{x}(\tau) + \frac{1}{2}g(T - \tau)^2 \hat{S}(T - \tau, \tilde{x}(\tau))$$

where $\hat{S}$ is the learned (pre-trained) score.

Euler discretization

On a grid $0 = \tau_0 < \tau_1 < \dots < \tau_K = T$, with $h_i = \tau_{i+1} - \tau_i$,

$$\tilde{x}_{i+1} = \tilde{x}_i + h_i \left[f(T - \tau_i) \tilde{x}_i + \frac{1}{2} g(T - \tau_i)^2 \hat{S}(T - \tau_i, \tilde{x}_i) \right].$$

- Uniform grid $\tau_i = iT/K$ treats all time regions equally.
- The reverse dynamics are not “equal” along the trajectory.
- Therefore, the same number of evaluations should be redistributed under the same terminal time.

ART: introduce a new clock

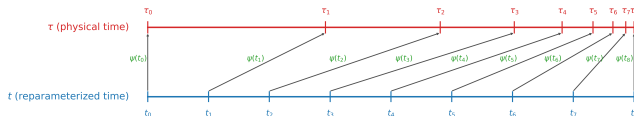
- Replace the original diffusion time τ by a learned clock map $\tau = \psi(t)$ with $\psi(0) = 0$ and $\psi(T) = T$.
- Track the backward state on this new clock:

$$x(t) = \tilde{x}(\psi(t)).$$

- Control the speed of the clock:

$$\theta(t) = \psi'(t), \quad \int_0^T \theta(t) dt = T.$$

- The time budget stays fixed: slowing down in one region must be compensated by speeding up elsewhere.



Controlled dynamics

By the chain rule, ART gives

$$\begin{cases} x'(t) = \theta(t)F(x(t), \psi(t)), \\ \psi'(t) = \theta(t), \end{cases}$$

where

$$F(x, \psi) = f(T - \psi)x + \frac{1}{2}g(T - \psi)^2\hat{S}(T - \psi, x).$$

- Large $\theta(t)$ moves faster in original diffusion time.
- Small $\theta(t)$ spends more grid resolution in that region.
- Thus, ART redistributes a fixed number of evaluations along the reverse trajectory.

Discretization error

Euler dismisses the second-order term $x''(t)$, which is

$$\frac{d}{dt}[\theta(t)F(x(t), \psi(t))] = \theta'(t)F(x(t), \psi(t)) + \theta(t)^2 Q(x(t), \psi(t)).$$

$$Q(x, \psi) = A(s, x)B(s, x) - g(s)g'(s)\hat{S}(s, x) - f'(s)x - \frac{1}{2}g(s)^2\partial_s\hat{S}(s, x),$$

$$A(s, x) = f(s)I_d + \frac{1}{2}g(s)^2\nabla_x\hat{S}(s, x),$$

$$B(s, x) = f(s)x + \frac{1}{2}g(s)^2\hat{S}(s, x), \quad s = T - \psi.$$

On $[t_i, t_{i+1})$, the implementation holds the control at a constant value θ_i . Thus within that interval the leading error is proportional to

$$|Q(x(t_i), \psi(t_i))|\theta_i^2.$$

ART control objective

Total-time budget. The clock must traverse exactly the original diffusion horizon:

$$\int_0^T \theta(t) dt = \psi(T) - \psi(0) = T.$$

Objective functional. Maximize accumulated negative Euler-error under this budget:

$$J^\theta(s, y, \phi) = \mathbb{E} \left[\int_s^T (-|Q(x(t), \psi(t))| \theta(t)^2 - \gamma \theta(t)) dt + \gamma T \mid x(s) = y, \psi(s) = \phi \right]$$

where γ is the Lagrange multiplier.

Optimal value function.

$$V(s, y, \phi) = \max_{\theta} J^\theta(s, y, \phi).$$

Drop-in schedule replacement

What changes?

Only the timestep list changes; the score model, backbone, solver, and sampler implementation stay fixed.

- **Uniform**: $\psi(t) = t$, a fixed evenly spaced clock.
- **EDM/DPM**: fixed hand-crafted reparameterizations.
- The learned grid can be inserted into EDM, EDM2, and other samplers without changing anything else.

- 1 Diffusion models for GenAI and sampling efficiency
- 2 ART: learn a sampling clock
- 3 Theory: solving ART via ART-RL
- 4 Algorithm: actor-critic
- 5 Experiments: drop-in schedule replacement
- 6 Takeaways

Why randomized control?

- ART, a deterministic optimal control problem, has no closed-form solution.
- Its HJB equation is high-dimensional in the sampling state x - *curse of dimensionality*.
- We introduce a randomized auxiliary problem with *Gaussian* policies.
- The randomization embeds ART into continuous-time RL which is solvable.

Gaussian policy

For each (t, x, ψ) , let

$$\pi^{(\lambda)}(\cdot \mid t, x, \psi) = \mathcal{N}\left(\mu(t, x, \psi), \frac{\lambda}{|Q(x, \psi)|}\right).$$

- The mean μ is the time-warping rate to be learned.
- The variance is smaller when the local error geometry is large.
- The parameter λ controls level of randomization in the auxiliary RL problem.

ART-RL auxiliary problem

Exploratory dynamics. Average the controlled dynamics under the Gaussian policy (Wang et al., 2020):

$$\begin{cases} \frac{dx^\pi(t)}{dt} = \int_{\mathbb{R}} \theta F(x^\pi(t), \psi^\pi(t)) \pi_t(\theta) d\theta, \\ \frac{d\psi^\pi(t)}{dt} = \int_{\mathbb{R}} \theta \pi_t(\theta) d\theta. \end{cases}$$

Objective and value function. Optimize the policy-averaged surrogate reward:

$$J^\pi(s, y, \phi) = \mathbb{E} \left[\gamma T + \lambda T + \int_s^T \int_{\mathbb{R}} \pi_t(\theta) (-|Q(x^\pi(t), \psi^\pi(t))| \theta^2 - \gamma \theta) d\theta dt \right. \\ \left. \mid x^\pi(s) = y, \psi^\pi(s) = \phi \right],$$

$$V^{(\lambda)}(s, y, \phi) = \max_{\pi \in \Pi(\lambda)} J^\pi(s, y, \phi).$$

HJB equations: ART versus ART-RL

Original ART HJB

$$V_t + \sup_{\theta} \{ (V_x^\top F(x, \psi) + V_\psi - \gamma)\theta - |Q(x, \psi)|\theta^2 \} = 0.$$

Exploratory ART-RL HJB

$$V_t^{(\lambda)} + \sup_{\mu} \left\{ (V_x^{(\lambda)\top} F(x, \psi) + V_\psi^{(\lambda)} - \gamma)\mu - |Q(x, \psi)| \left(\mu^2 + \frac{\lambda}{|Q(x, \psi)|} \right) \right\} = 0.$$

Terminal conditions: $V(T, x, \psi) = \gamma T$ and $V^{(\lambda)}(T, x, \psi) = (\gamma + \lambda)T$.

Theorem 1: value function shift

Theorem (Value function shift)

If V is a classical solution of the original ART HJB equation, then

$$V^{(\lambda)}(t, x, \psi) = V(t, x, \psi) + \lambda t$$

is a classical solution of the ART-RL exploratory HJB equation.

- The randomized ART-RL problem differs from ART only by a linear value shift.
- Therefore, the auxiliary RL formulation preserves the original schedule-learning objective.

Theorem 2: recovery of ART control

Theorem (Recovery of the optimal ART control)

Under the conditions of Theorem 1, if

$$\pi^{(\lambda)*}(\cdot \mid t, x, \psi) = \mathcal{N}\left(\mu^*(t, x, \psi), \frac{\lambda}{|Q(x, \psi)|}\right)$$

is optimal for ART-RL, then its mean μ^ is optimal for ART. Moreover,*

$$\mu^*(t, x, \psi) = \frac{V_x^\top F(x, \psi) + V_\psi - \gamma}{2|Q(x, \psi)|}.$$

Theorem 3: policy evaluation and improvement

Theorem (Policy evaluation and improvement)

A Gaussian policy $\pi^{(\lambda)}(\cdot \mid t, x, \psi) = \mathcal{N}\left(\mu(t, x, \psi), \frac{\lambda}{|Q(x, \psi)|}\right)$ has the value function $J^{\pi^{(\lambda)}}(t, x, \psi) = \bar{v}(t, x, \psi) + \lambda t$ where $\bar{v}$ satisfies certain PDE. Construct $\tilde{\pi}^{(\lambda)}(\cdot \mid t, x, \psi) = \mathcal{N}\left(\tilde{\mu}(t, x, \psi), \frac{\lambda}{|Q(x, \psi)|}\right)$ where

$$\tilde{\mu}(t, x, \psi) = \frac{\bar{v}_x(t, x, \psi)^\top F(x, \psi) + \bar{v}_\psi(t, x, \psi) - \gamma}{2|Q(x, \psi)|}.$$

Then $J^{\tilde{\pi}^{(\lambda)}}(t, x, \psi) \geq J^{\pi^{(\lambda)}}(t, x, \psi)$.

Theorem 4: trajectory-based policy gradient

Theorem (Policy evaluation and policy-gradient equations)

The critic is identified by the orthogonality condition

$$\mathbb{E} \left[\int_s^T \xi(t, x^\pi, \psi^\pi) \left(d\hat{V} - (|Q(x^\pi, \psi^\pi)|\theta^\pi(t)^2 + \gamma\theta^\pi(t))dt \right) \right] = 0.$$

The actor direction is identified by

$$\mathbb{E} \left[\int_s^T \eta(t, x^{\hat{\pi}}, \psi^{\hat{\pi}}) (\theta^{\hat{\pi}}(t) - \hat{\mu}(t, x^{\hat{\pi}}, \psi^{\hat{\pi}})) \left(dJ^\pi - (|Q|\theta^{\hat{\pi}}(t)^2 + \gamma\theta^{\hat{\pi}}(t))dt \right) \right] = 0.$$

The stochastic control term $\theta^{\hat{\pi}} - \hat{\mu}$ is what gives a nontrivial policy-improvement direction.

- 1 Diffusion models for GenAI and sampling efficiency
- 2 ART: learn a sampling clock
- 3 Theory: solving ART via ART-RL
- 4 Algorithm: actor-critic**
- 5 Experiments: drop-in schedule replacement
- 6 Takeaways

Actor-critic parameterization

We use two neural networks:

$$\hat{V}^{\vartheta_c}(t, x, \psi) = NN^{\vartheta_c}(t, x, \psi) + \lambda t,$$

$$\hat{\pi}^{\vartheta_a}(\cdot \mid t, x, \psi) = \mathcal{N}\left(NN^{\vartheta_a}(t, x, \psi), \frac{\lambda}{|Q(x, \psi)|}\right).$$

- Critic approximates the shifted value function.
- Actor approximates the policy mean.
- Lagrange multiplier enforces the terminal-time constraint.

Moment conditions and TD residual

For a trajectory $(x(t), \psi(t), \theta(t))$, the actor-critic moment conditions are

$$\mathbb{E} \left[\int_0^T \frac{\partial NN^{\vartheta_c}}{\partial \vartheta_c} \left(d\hat{V}^{\vartheta_c} - (|Q|\theta^2 + \gamma\theta)dt \right) \right] = 0,$$

$$\mathbb{E} \left[\int_0^T \frac{\partial \log \hat{\pi}^{\vartheta_a}(\theta(t) \mid t, x, \psi)}{\partial \vartheta_a} \left(d\hat{V}^{\vartheta_c} - (|Q|\theta^2 + \gamma\theta)dt \right) \right] = 0.$$

By stochastic approximation, we view these moment conditions as equations in $(\vartheta_c, \vartheta_a)$. On a grid $0 = t_0 < \dots < t_K = T$, define

$$D_{n,k} = \hat{V}_{k+1}^{\vartheta_{c,n}} - \hat{V}_k^{\vartheta_{c,n}} - \gamma_n \theta_n(t_k) \Delta t - |Q(x_n(t_k), \psi_n(t_k))| \theta_n(t_k)^2 \Delta t.$$

Stochastic approximation update rules

The discrete-time stochastic approximation updates are

$$\vartheta_{c,n+1} \leftarrow \vartheta_{c,n} + a_n \sum_{k=0}^{K-1} \frac{\partial NN^{\vartheta_{c,n}}}{\partial \vartheta_c}(t_k, \mathbf{x}_n(t_k), \psi_n(t_k)) D_{n,k},$$

$$\vartheta_{a,n+1} \leftarrow \vartheta_{a,n} + a_n \sum_{k=0}^{K-1} \frac{\partial \log \hat{\pi}_{t_k}^{\vartheta_{a,n}}(\theta_n(t_k))}{\partial \vartheta_a} D_{n,k}.$$

$$\gamma_{n+1} \leftarrow \gamma_n + a_n(\psi_n(T) - T).$$

These updates are the implementable actor, critic, and multiplier updates used inside ART-RL.

Time-discretized ART-RL actor-critic

```
1: for  $n = 1, \dots, N$  do
2:   Initialize  $x_n(t_0) \sim P_T$ ,  $\psi_n(t_0) = 0$ 
3:   for  $k = 0, \dots, K - 1$  do
4:     Sample  $\theta_n(t_k) \sim \hat{\pi}^{\vartheta_{a,n}}(\cdot \mid t_k, x_n(t_k), \psi_n(t_k))$ 
5:     Update  $(x_n(t_{k+1}), \psi_n(t_{k+1}))$  by ART dynamics
6:   end for
7:   Compute residuals  $D_{n,k}$ 
8:    $\vartheta_{c,n+1} \leftarrow \vartheta_{c,n} + a_n \sum_{k=0}^{K-1} \frac{\partial NN^{\vartheta_{c,n}}}{\partial \vartheta_c}(t_k, x_n(t_k), \psi_n(t_k)) D_{n,k}$ 
9:    $\vartheta_{a,n+1} \leftarrow \vartheta_{a,n} + a_n \sum_{k=0}^{K-1} \frac{\partial \log \hat{\pi}_{t_k}^{\vartheta_{a,n}}(\theta_n(t_k))}{\partial \vartheta_a} D_{n,k}$ 
10:   $\gamma_{n+1} \leftarrow \gamma_n + a_n(\psi_n(T) - T)$ 
11: end for
```

- 1 Diffusion models for GenAI and sampling efficiency
- 2 ART: learn a sampling clock
- 3 Theory: solving ART via ART-RL
- 4 Algorithm: actor-critic
- 5 Experiments: drop-in schedule replacement
- 6 Takeaways

Experimental design

- Controlled comparison: keep the model, solver, and sampler implementation fixed.
- Vary only the timestep schedule:
 - ▶ Uniform grid;
 - ▶ DPM-Solver timestep grid;
 - ▶ EDM timestep grid;
 - ▶ ART-RL learned grid.
- Evaluation depends on the setting:
 - ▶ W_2 for the one-dimensional analytical-score example;
 - ▶ FID (Fréchet Inception Distance) versus NFE (number of function evaluations) for image generation.

One-dimensional analytical-score testbed

Forward process:

$$dx(t) = \sqrt{2t} dW(t), \quad x(0) \sim \mathcal{N}(0, 1), \quad T = 3.$$

Then

$$x(t) \sim \mathcal{N}(0, 1 + t^2), \quad S(t, x) = -\frac{x}{1 + t^2}.$$

- Because the score is exact, score model error is removed from the comparison.
- Therefore, performance differences come only from the timestep schedule.

One-dimensional W_2 accuracy

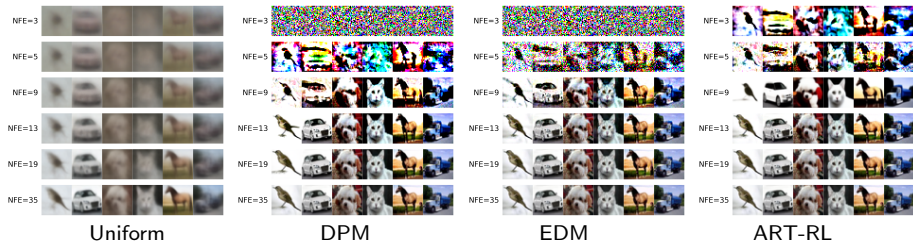
K	2	5	10	20	50	100
Uniform	.468	.215	.114	.060	.027	.016
DPM	.670	.401	.211	.113	.049	.027
EDM	.664	.319	.177	.094	.041	.023
ART-RL	.345	.149	.079	.042	.020	.013

CIFAR-10 under the EDM pipeline

NFE	3	5	9	13	19	35
Uniform	280.29	254.47	213.13	191.69	168.87	118.02
DPM	465.83	244.50	52.29	8.67	2.76	1.89
EDM	465.83	305.15	35.54	6.79	2.54	1.85
ART-RL	152.86	130.48	32.13	5.44	2.45	1.82

- Controlled setting: the score model, solver, and EDM implementation are fixed.
- Intervention: ART-RL replaces only the timestep grid.
- Outcome: ART-RL is consistently best across all tested NFEs.

CIFAR-10 sample visualization



Heun sampler and robustness

- Main image results use the default EDM Heun sampler.
- This tests whether an Euler-trained schedule still works under a stronger practical solver.
- The learned grid remains effective under Heun; a multi-seed check at $\text{NFE} = 35$ gives

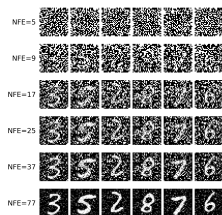
ART-RL: 1.82, 1.79, 1.82, EDM: 1.85, 1.83, 1.85.

MNIST small-model test with RK4

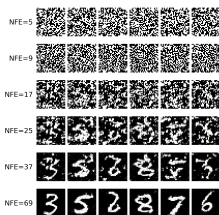
NFE	5	9	17	25	37	69
Uniform	365.10	305.91	278.79	256.35	208.55	111.11
DPM	345.09	322.41	246.46	238.82	41.41	4.66
EDM	345.09	313.65	73.91	32.36	5.37	1.01
ART-RL	261.63	118.92	69.48	28.07	5.16	0.75

- Purpose: test a deliberately **small score model**, not a large pretrained EDM checkpoint.
- Outcome: learned timestep allocation remains useful beyond modern large-model pipelines.

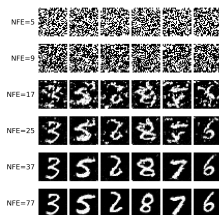
MNIST sample visualization



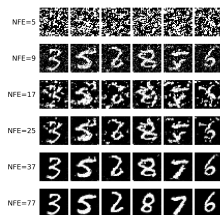
Uniform



DPM



EDM



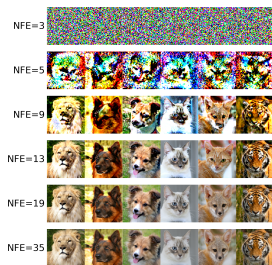
ART-RL

Transfer and amortization

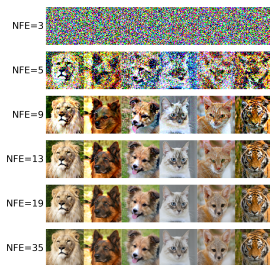
- Training phase: learn ART-RL once on CIFAR-10.
- Distillation phase: convert the policy to a fixed time-only schedule.
- Transfer phase: apply the same schedule
 - ▶ across AFHQv2, FFHQ, ImageNet-64, and ImageNet-512 without retraining;
 - ▶ from pixel-space EDM to [latent-space EDM2](#).
- Training is offline and short, about 1–2 hours on a Colab T4, so reuse amortizes the cost.

AFHQv2 transfer

Method	3	5	9	13	19	35
DPM	375.76	321.59	67.64	9.77	3.44	2.15
EDM	375.76	266.02	27.88	7.56	2.99	2.11
ART-RL	243.48	194.79	20.48	6.12	2.85	2.07



DPM



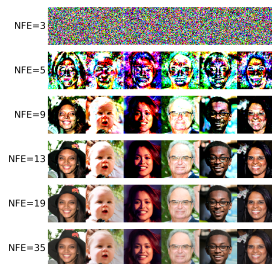
EDM



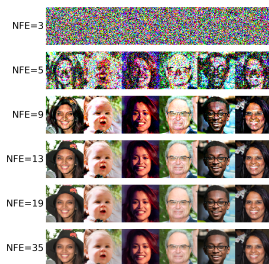
ART-RL

FFHQ transfer

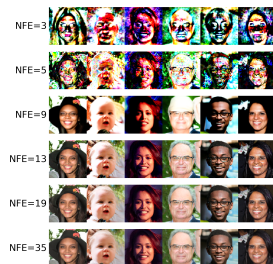
Method	3	5	9	13	19	35
DPM	466.76	340.51	113.87	15.94	5.25	2.66
EDM	466.76	344.76	57.13	15.87	5.26	2.73
ART-RL	305.97	240.38	35.73	11.08	4.31	2.57



DPM



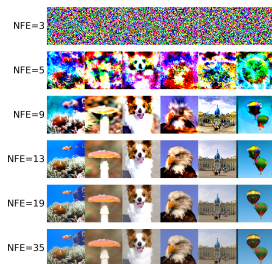
EDM



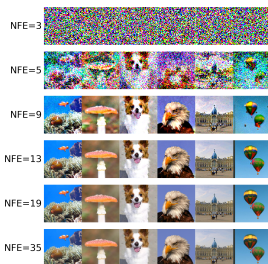
ART-RL

ImageNet transfer

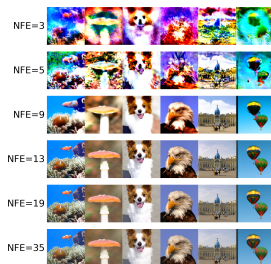
Method	3	5	9	13	19	35
DPM	437.42	233.35	60.48	12.31	4.46	2.66
EDM	437.42	248.32	35.32	8.18	3.68	2.57
ART-RL	147.21	108.47	29.49	7.01	3.62	2.56



DPM

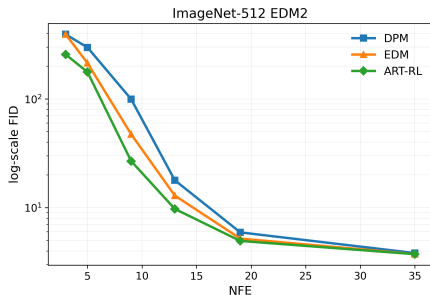


EDM



ART-RL

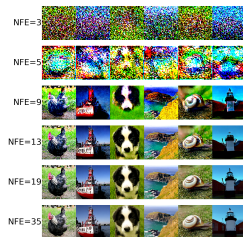
ImageNet-512: transfer beyond EDM



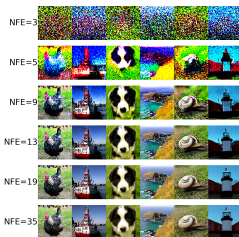
- The CIFAR-10 schedule is reused directly; only the timestep grid changes.
- This is a stronger transfer test because EDM2 changes the **backbone and pipeline**.
- It also moves from pixel-space EDM to **latent-space** diffusion.
- ART-RL still improves over both DPM and EDM at matched budgets.

ImageNet-512 samples under EDM2

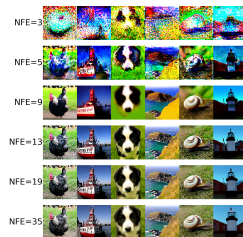
Method	3	5	9	13	19	35
DPM	392.19	297.26	99.38	17.86	5.92	3.82
EDM	392.19	213.45	47.33	12.91	5.19	3.74
ART-RL	256.13	176.50	26.78	9.73	4.94	3.73



DPM



EDM



ART-RL

- 1 Diffusion models for GenAI and sampling efficiency
- 2 ART: learn a sampling clock
- 3 Theory: solving ART via ART-RL
- 4 Algorithm: actor-critic
- 5 Experiments: drop-in schedule replacement
- 6 Takeaways

What the experiments show

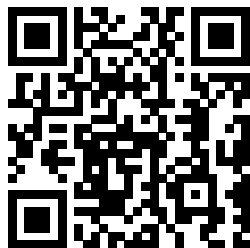
- Accuracy: ART-RL consistently improves over Uniform, DPM, and EDM at matched budgets.
- Efficiency: the largest gains occur at low and moderate NFEs.
- Robustness: the learned schedule works under Heun, RK4, EDM, and EDM2 settings.
- Reuse: a single distilled schedule transfers across datasets, pipelines, and representation spaces.

Conclusions

- We introduce **ART**: timestep allocation as continuous-time control of the sampling clock.
- Inherent high dimensional - curse of dimensionality.
- Solve the problem by RL and, in particular, by randomizing controls.
- Here randomization is not for conceptual reason (exploration), but for *technical* one (policy improvement).
- Distillation and transfer learning.

ART for Diffusion Sampling

Adaptive Reparameterized Time for Timestep Scheduling



<https://arxiv.org/abs/2601.18681>

Thank you
Questions and discussion

References I

Chen, H., Lee, H., and Lu, J. (2023). Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *ICML*, pages 4735–4763.

Google (2024). State-of-the-art video and image generation with veo 2 and imagen 3. <https://blog.google/technology/google-labs/video-image-generation-update-december-2024/>. Accessed: 2025-09-17.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Neurips*, volume 33, pages 6840–6851.

Jia, Y. and Zhou, X. Y. (2022a). Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *J. Mach. Learn. Res.*, 23(154):1–55.

Jia, Y. and Zhou, X. Y. (2022b). Policy gradient and actor-critic learning in continuous time and space: Theory and algorithms. *J. Mach. Learn. Res.*, 23(154):1–55.

References II

- Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. In *Neurips*, volume 35, pages 26565–26577.
- Khanna, S., Kharbanda, S., Li, S., Varma, H., Wang, E., Birnbaum, S., Luo, Z., Miraoui, Y., Palrecha, A., and Ermon, S. (2025). Mercury: Ultra-fast language models based on diffusion. arXiv:2506.17298.
- Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. (2022). Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787.
- Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. (2025). Large language diffusion models. arXiv:2502.09992.
- OpenAI (2024). Sora: Creating video from text. <https://openai.com/sora>. Accessed: 2025-09-17.

References III

- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. arXiv:2204.06125.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695.
- Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., and Gafni, O. (2023). Make-a-video: Text-to-video generation without text-video data. In *ICLR*.
- Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. In *ICLR*.
- Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Neurips*, volume 32, page 11918–11930.

References IV

- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021b). Score-based generative modeling through stochastic differential equations. In *ICLR*.
- Wang, H., Zariphopoulou, T., and Zhou, X. Y. (2020). Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21(198):1–34.