Reward Score Matching

Unifying reward-based fine-tuning for flow and diffusion models

Jeongjae Lee*, Jinho Chang*, Jeongsol Kim†, Jong Chul Ye†

KAIST AI
Kim Jaechul Graduate School

ICML 2026 SPIGM Workshop

Flow-based models

$p_1(\boldsymbol{x}_1)$ $p_0(\boldsymbol{x}_0)$

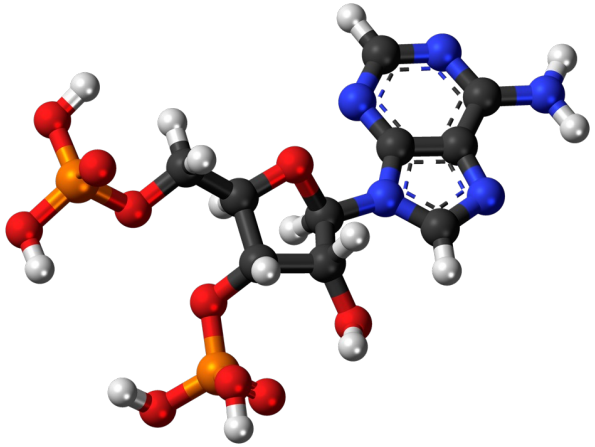
$t = 1$ $t = 0$

Reward model

Introduces an additional objective/constraint:

$$r : \mathbb{R}^d \to \mathbb{R}, \boldsymbol{x}_0 \mapsto r(\boldsymbol{x}_0)$$

QED
Drug likeliness of molecule



Aesthetic Score
Aesthetic quality of image

Reward alignment (fine-tuning)

Given: Pretrained generative model p^{ref}

Obtain: Finetuned generative model $p^{\star}$

$$p^{\star}(\boldsymbol{x}_0) = \arg\max_{q} \mathbb{E}_{\boldsymbol{x}_0 \sim q}\Big[r(\boldsymbol{x}_0)\Big]$$

Reward alignment (fine-tuning)

Given: Pretrained generative model p^{ref}

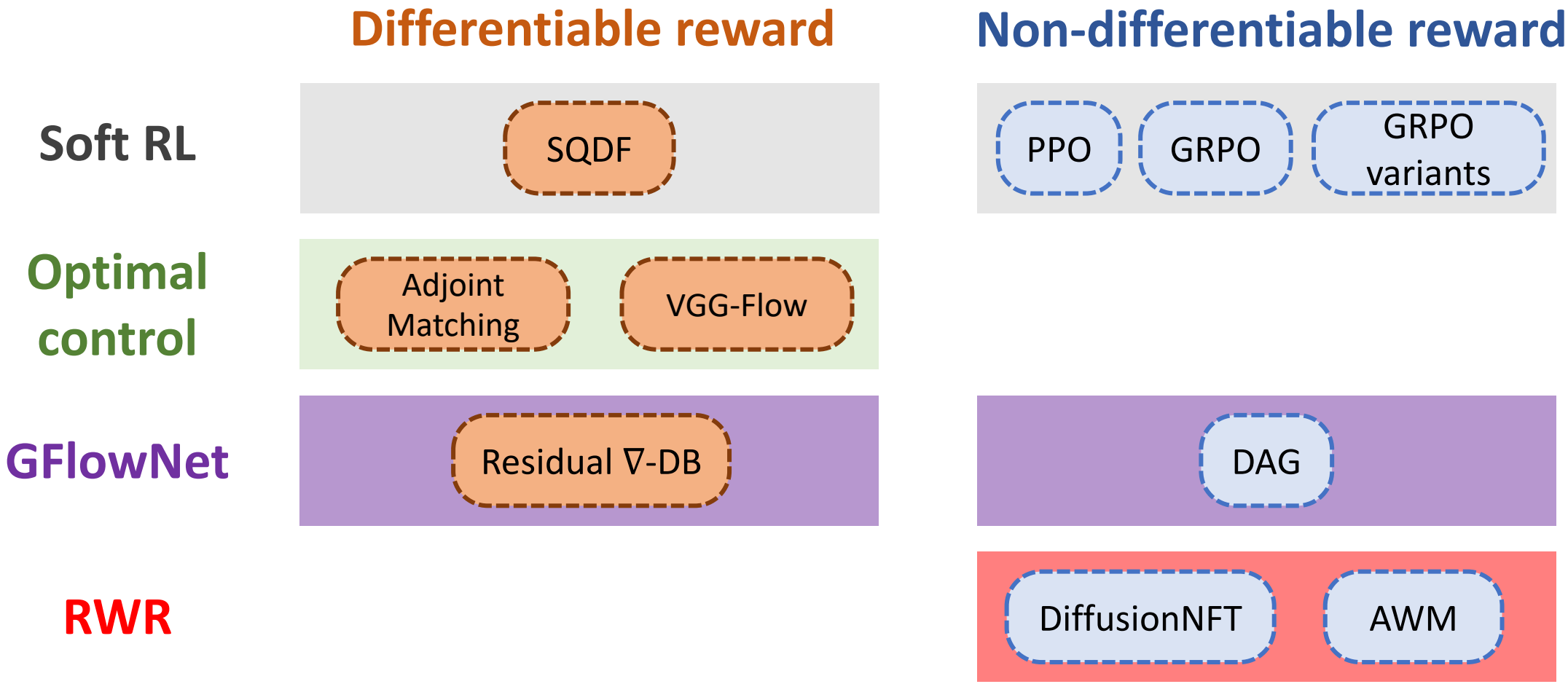
Obtain: Finetuned generative model $p^{\star}$

$$p^{\star}(\boldsymbol{x}_0) = \arg\max_{q} \mathbb{E}_{\boldsymbol{x}_0 \sim q}\Big[r(\boldsymbol{x}_0)\Big] - \alpha \mathcal{D}_{\mathrm{KL}}\Big(q \| p^{\mathrm{ref}}\Big)$$

$$= \frac{p^{\mathrm{ref}}(\boldsymbol{x}_0)}{Z} \exp\left(\frac{r(\boldsymbol{x}_0)}{\alpha}\right)$$

Reward-based fine-tuning ➔ Reward-tilted distribution

Fragmented literature



One problem, Many methods.

RL fine-tuning

$$p^\star(\boldsymbol{x}_t) = \frac{p^{\mathrm{ref}}(\boldsymbol{x}_t)}{Z} \exp\left(\frac{V_t(\boldsymbol{x}_t)}{\alpha}\right)$$

RL fine-tuning

$$p^\star(\boldsymbol{x}_t) = \frac{p^{\mathrm{ref}}(\boldsymbol{x}_t)}{Z} \exp\left(\frac{V_t(\boldsymbol{x}_t)}{\alpha}\right)$$

$$s^\star(\boldsymbol{x}_t) = s^{\mathrm{ref}}(\boldsymbol{x}_t) + \frac{1}{\alpha}\nabla_{\boldsymbol{x}_t} V_t(\boldsymbol{x}_t)$$

RL fine-tuning as score matching

$$p^\star(\boldsymbol{x}_t) = \frac{p^{\mathrm{ref}}(\boldsymbol{x}_t)}{Z} \exp\left(\frac{V_t(\boldsymbol{x}_t)}{\alpha}\right)$$

$$s^\star(\boldsymbol{x}_t) = s^{\mathrm{ref}}(\boldsymbol{x}_t) + \frac{1}{\alpha}\nabla_{\boldsymbol{x}_t} V_t(\boldsymbol{x}_t)$$

$$\mathcal{L}_{\mathrm{RSM}}(\theta) = \mathbb{E}_{t_i, \boldsymbol{x}_{t_i}, \boldsymbol{\epsilon}}\Big[C_1(t_i)\Big(\underbrace{\|\boldsymbol{s}^\theta_{t_i} - (\boldsymbol{s}^{\mathrm{ref}}_{t_i} + \gamma(t_i)\hat{\boldsymbol{\Psi}}_{t_i})\|^2}_{\text{Guidance}} + C_2(t_i)\underbrace{\|\boldsymbol{s}^\theta_{t_i} - \boldsymbol{s}^{\theta_{\mathrm{old}}}_{t_i}\|^2}_{\text{Regularization}}\Big)\Big]$$

Many methods = Score matching against value-tilted score

Flow-GRPO

```
ratio = torch.exp(log_prob - sample["log_probs"][:, j])
```

Flow-GRPO (Liu et. al., NeurIPS 2025)

$$p(\boldsymbol{x}) = (\text{const}) \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{\mu}\|^2}{(\text{const})}\right)$$

$$\log p(\boldsymbol{x}) = (\text{const}) + (\text{const})\|\boldsymbol{x}-\boldsymbol{\mu}\|^2$$

Flow-GRPO

```
ratio = torch.exp(log_prob - sample["log_probs"][:, j])
```

Flow-GRPO (Liu et. al., NeurIPS 2025)

$\widehat{1 + \log \rho_t(\theta)}$ is a better estimator of $\rho_t(\theta)$ than $\widehat{\rho_t(\theta)}$

Avoid exponentiation $\Longrightarrow$ Much lower variance

PCPO (Lee and Ye, ICLR 2026)

Flow-GRPO is also L_2 regression

```
ratio = torch.exp(log_prob - sample["log_probs"][:, j])
```

Flow-GRPO (Liu et. al., NeurIPS 2025)

$\widehat{1 + \log \rho_t(\theta)}$ is a better estimator of $\rho_t(\theta)$ than $\widehat{\rho_t(\theta)}$

Avoid exponentiation $\Longrightarrow$ Much lower variance

PCPO (Lee and Ye, ICLR 2026)

$$\mathcal{L}_{\mathrm{RSM}}(\theta) = \mathbb{E}_{t_i, \boldsymbol{x}_{t_i}, \boldsymbol{\epsilon}} \left[C_1(t_i) \Big(\underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - (\boldsymbol{s}^{\mathrm{ref}}_{t_i} + \gamma(t_i) \hat{\boldsymbol{\Psi}}_{t_i}) \|^2}_{\text{Guidance}} + C_2(t_i) \underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - \boldsymbol{s}^{\theta_{\mathrm{old}}}_{t_i} \|^2}_{\text{Regularization}} \Big) \right]$$

Flow-GRPO is also L_2 regression against the value-tilted score.

Three design knobs

$$\mathcal{L}_{\mathrm{RSM}}(\theta) = \mathbb{E}_{t_i, \boldsymbol{x}_{t_i}, \boldsymbol{\epsilon}} \Big[C_1(t_i) \Big(\underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - (\boldsymbol{s}^{\mathrm{ref}}_{t_i} + \gamma(t_i) \hat{\boldsymbol{\Psi}}_{t_i}) \|^2}_{\text{Guidance}} + C_2(t_i) \underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - \boldsymbol{s}^{\theta_{\mathrm{old}}}_{t_i} \|^2}_{\text{Regularization}} \Big) \Big]$$

$\hat{\boldsymbol{\Psi}}_{t_i}$	value guidance estimator
$C_1(t_i), \gamma(t_i)$	guidance strength
$C_2(t_i)$, clipping	regularization

These three knobs explain differences among methods.

Knob 1: Estimator design

$$\hat{\Psi}_{t_i}^{\mathrm{LA},1} = \frac{\sigma_{t_i}^2}{\alpha\Omega(t_i)} \frac{1}{K_i} \sum_{k=1}^{K_i} \nabla_{\boldsymbol{x}_{t_{i-1}}^{(k)}} r(\hat{\boldsymbol{x}}_{0|t_j}^{(k)}) \approx \frac{\sigma_{t_i}^2}{\alpha\Omega(t_i)} \mathbb{E}_{\boldsymbol{x}_{t_{i-1}:t_j}} \left[\nabla_{\boldsymbol{x}_{t_{i-1}}} r(\hat{\boldsymbol{x}}_{0|t_j}) \right]$$

$$\hat{\Psi}_{t_i}^{\mathrm{LA},0} = \frac{\sigma_{t_i}}{\alpha\Omega(t_i)} \frac{1}{K_i} \sum_{k=1}^{K_i} r(\hat{\boldsymbol{x}}_{0|t_j}^{(k)}) \boldsymbol{\epsilon}_{t_i}^{(k)} \approx \frac{\sigma_{t_i}}{\alpha\Omega(t_i)} \mathbb{E}_{\boldsymbol{x}_{t_{i-1}:t_j}} \left[r(\hat{\boldsymbol{x}}_{0|t_j}) \boldsymbol{\epsilon}_{t_i} \right]$$

$\mathbb{E}_{\boldsymbol{x}}[\nabla r(\boldsymbol{x})]$ vs $\mathbb{E}_{\boldsymbol{x},\epsilon}[r(\boldsymbol{x})\epsilon]$ — first-order vs zeroth-order

$\mathbb{E}_{\boldsymbol{x}} \Rightarrow \frac{1}{K_i} \sum_{k=1}^{K_i} \boldsymbol{x}^{(k)} \ldots$ — number of particles

$\mathbb{E}_{\boldsymbol{x}_0}[r(\boldsymbol{x}_0)\cdots]$ vs $\mathbb{E}_{\boldsymbol{x}_{t_j}}[r(\hat{\boldsymbol{x}}_{0|t_j})\cdots]$ — rollout depth

Knob 1: Estimator design (Part a)

$$\hat{\Psi}_{t_i}^{\mathrm{LA},1} \approx (\text{coefficient})\mathbb{E}_{\boldsymbol{x}}\left[\nabla_{\boldsymbol{x}} r(\boldsymbol{x})\right]$$

$$\hat{\Psi}_{t_i}^{\mathrm{LA},0} \approx (\text{coefficient})\mathbb{E}_{\boldsymbol{x}}\left[r(\boldsymbol{x})\boldsymbol{\epsilon}_{t_i}\right]$$

$\mathbb{E}_{\boldsymbol{x}}[\nabla r(\boldsymbol{x})]$ vs $\mathbb{E}_{\boldsymbol{x},\epsilon}[r(\boldsymbol{x})\epsilon]$

first-order vs zeroth-order

Knob 1: Estimator design (Part b)

$$\hat{\Psi}_{t_i}^{\mathrm{LA},1} = (\text{coefficient})\frac{1}{K_i}\sum_{k=1}^{K_i} \nabla_{\boldsymbol{x}^{(k)}} r(\hat{\boldsymbol{x}}^{(k)}) \approx (\text{coefficient})\mathbb{E}_{\boldsymbol{x}}\left[\nabla_{\boldsymbol{x}} r(\boldsymbol{x})\right]$$

$$\hat{\Psi}_{t_i}^{\mathrm{LA},0} = (\text{coefficient})\frac{1}{K_i}\sum_{k=1}^{K_i} r(\hat{\boldsymbol{x}}^{(k)})\,\boldsymbol{\epsilon}^{(k)} \approx (\text{coefficient})\mathbb{E}_{\boldsymbol{x}}\left[r(\boldsymbol{x})\,\boldsymbol{\epsilon}_{t_i}\right]$$

$\mathbb{E}_{\boldsymbol{x}}[\nabla r(\boldsymbol{x})]$ vs $\mathbb{E}_{\boldsymbol{x},\epsilon}[r(\boldsymbol{x})\epsilon]$

first-order vs zeroth-order

$\mathbb{E}_{\boldsymbol{x}} \Rightarrow \frac{1}{K_i}\sum_{k=1}^{K_i} \boldsymbol{x}^{(k)} \ldots$

number of particles

Knob 1: Estimator design (Part c)

$$\hat{\Psi}^{\mathrm{LA},1}_{t_i} = (\text{coefficient})\frac{1}{K_i}\sum_{k=1}^{K_i} \nabla_{\boldsymbol{x}^{(k)}_{t_{i-1}}} r(\hat{\boldsymbol{x}}^{(k)}_{0|t_j}) \approx (\text{coefficient})\mathbb{E}_{\boldsymbol{x}_{t_{i-1}:t_j}}\left[\nabla_{\boldsymbol{x}_{t_{i-1}}} r(\hat{\boldsymbol{x}}_{0|t_j})\right]$$

$$\hat{\Psi}^{\mathrm{LA},0}_{t_i} = (\text{coefficient})\frac{1}{K_i}\sum_{k=1}^{K_i} r(\hat{\boldsymbol{x}}^{(k)}_{0|t_j})\,\boldsymbol{\epsilon}^{(k)}_{t_i} \approx (\text{coefficient})\mathbb{E}_{\boldsymbol{x}_{t_{i-1}:t_j}}\left[r(\hat{\boldsymbol{x}}_{0|t_j})\,\boldsymbol{\epsilon}_{t_i}\right]$$

$\mathbb{E}_{\boldsymbol{x}}[\nabla r(\boldsymbol{x})]$ vs $\mathbb{E}_{\boldsymbol{x},\epsilon}[r(\boldsymbol{x})\epsilon]$ — first-order vs zeroth-order

$\mathbb{E}_{\boldsymbol{x}} \Rightarrow \frac{1}{K_i}\sum_{k=1}^{K_i} \boldsymbol{x}^{(k)} \ldots$ — number of particles

$\mathbb{E}_{\boldsymbol{x}_0}[r(\boldsymbol{x}_0)\cdots]$ vs $\mathbb{E}_{\boldsymbol{x}_{t_j}}[r(\hat{\boldsymbol{x}}_{0|t_j})\cdots]$ — rollout depth

Knob 1: Estimator design

$$\hat{\Psi}_{t_i}^{\mathrm{LA},1} = \frac{\sigma_{t_i}^2}{\alpha\Omega(t_i)} \frac{1}{K_i} \sum_{k=1}^{K_i} \nabla_{\boldsymbol{x}_{t_{i-1}}^{(k)}} r(\hat{\boldsymbol{x}}_{0|t_j}^{(k)}) \approx \frac{\sigma_{t_i}^2}{\alpha\Omega(t_i)} \mathbb{E}_{\boldsymbol{x}_{t_{i-1}:t_j}} \left[\nabla_{\boldsymbol{x}_{t_{i-1}}} r(\hat{\boldsymbol{x}}_{0|t_j}) \right]$$

$$\hat{\Psi}_{t_i}^{\mathrm{LA},0} = \frac{\sigma_{t_i}}{\alpha\Omega(t_i)} \frac{1}{K_i} \sum_{k=1}^{K_i} r(\hat{\boldsymbol{x}}_{0|t_j}^{(k)}) \boldsymbol{\epsilon}_{t_i}^{(k)} \approx \frac{\sigma_{t_i}}{\alpha\Omega(t_i)} \mathbb{E}_{\boldsymbol{x}_{t_{i-1}:t_j}} \left[r(\hat{\boldsymbol{x}}_{0|t_j}) \boldsymbol{\epsilon}_{t_i} \right]$$

$\mathbb{E}_{\boldsymbol{x}}[\nabla r(\boldsymbol{x})]$ vs $\mathbb{E}_{\boldsymbol{x},\epsilon}[r(\boldsymbol{x})\epsilon]$ First-order vs Zeroth-order

$\mathbb{E}_{\boldsymbol{x}} \Rightarrow \frac{1}{K_i} \sum_{k=1}^{K_i} \boldsymbol{x}^{(k)} \cdots$ More particles = More compute, Less variance

$\mathbb{E}_{\boldsymbol{x}_0}[r(\boldsymbol{x}_0) \cdots]$ vs $\mathbb{E}_{\boldsymbol{x}_{t_j}}[r(\hat{\boldsymbol{x}}_{0|t_j}) \cdots]$

More rollout depth = More compute, Less bias

Three design knobs

$$\mathcal{L}_{\mathrm{RSM}}(\theta) = \mathbb{E}_{t_i, \boldsymbol{x}_{t_i}, \boldsymbol{\epsilon}} \Big[C_1(t_i) \Big(\underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - (\boldsymbol{s}^{\mathrm{ref}}_{t_i} + \gamma(t_i) \hat{\boldsymbol{\Psi}}_{t_i}) \|^2}_{\text{Guidance}} + C_2(t_i) \underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - \boldsymbol{s}^{\theta_{\mathrm{old}}}_{t_i} \|^2}_{\text{Regularization}} \Big) \Big]$$

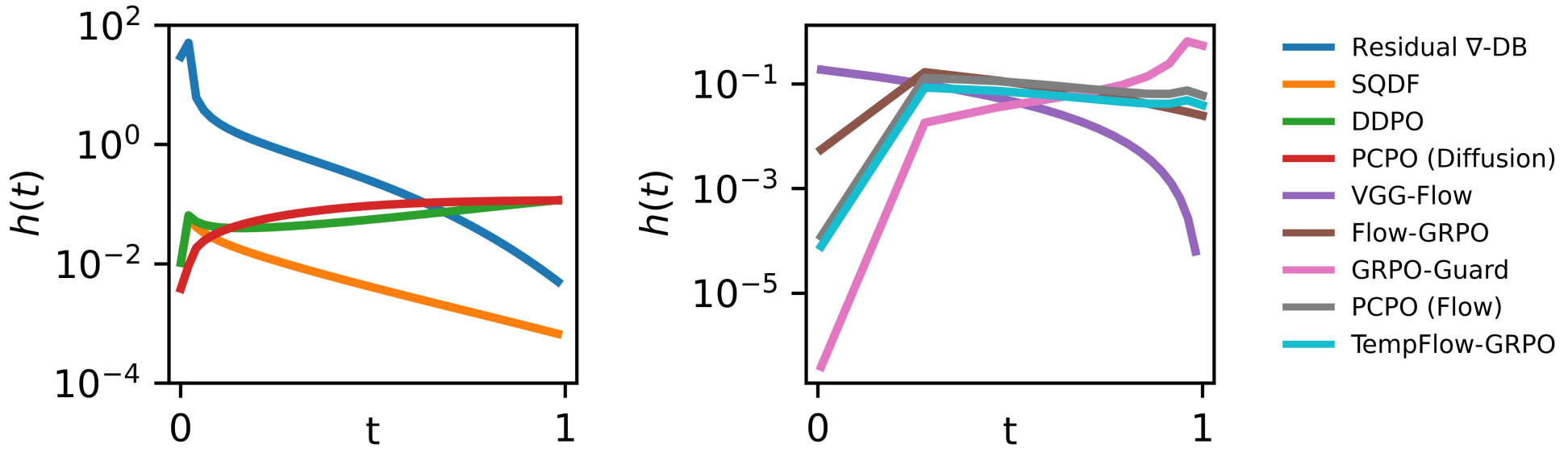
✅ $\hat{\boldsymbol{\Psi}}_{t_i}$ value guidance estimator

➡️ $C_1(t_i), \gamma(t_i)$ guidance strength

$C_2(t_i)$, clipping regularization

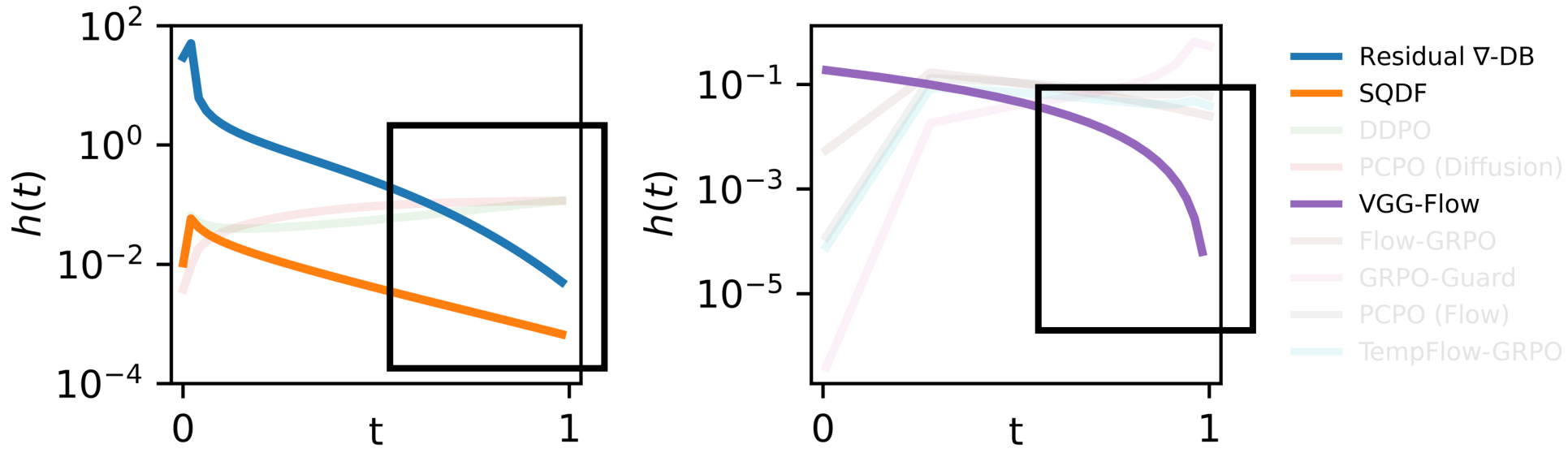
These three knobs explain differences among methods.

Knob 2: Guidance strength $h(t)$



$$h(t_i) := \boxed{C_1(t_i)\gamma(t_i)} \frac{\delta(t_i)\sigma_{t_i}^2}{\alpha\Omega(t_i)}$$

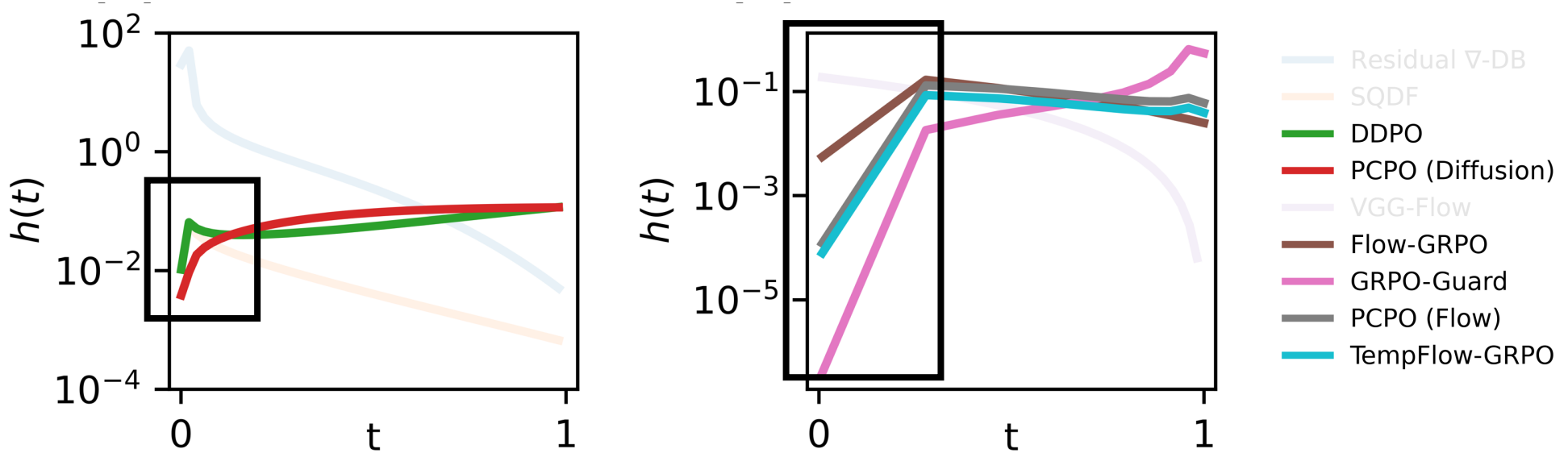
Knob 2: Guidance strength $h(t)$



$$h(t_i) := \boxed{C_1(t_i)\gamma(t_i)} \frac{\delta(t_i)\sigma_{t_i}^2}{\alpha\Omega(t_i)}$$

Existing methods suppress guidance at unreliable timesteps.

Knob 2: Guidance strength $h(t)$



$$h(t_i) := \boxed{C_1(t_i)\gamma(t_i)} \frac{\delta(t_i)\sigma_{t_i}^2}{\alpha\Omega(t_i)}$$

Existing methods suppress guidance at unreliable timesteps.

Three design knobs

$$\mathcal{L}_{\mathrm{RSM}}(\theta) = \mathbb{E}_{t_i, \boldsymbol{x}_{t_i}, \boldsymbol{\epsilon}} \Big[C_1(t_i) \Big(\underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - (\boldsymbol{s}^{\mathrm{ref}}_{t_i} + \gamma(t_i) \hat{\boldsymbol{\Psi}}_{t_i}) \|^2}_{\text{Guidance}} + C_2(t_i) \underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - \boldsymbol{s}^{\theta_{\mathrm{old}}}_{t_i} \|^2}_{\text{Regularization}} \Big) \Big]$$

✅ $\hat{\boldsymbol{\Psi}}_{t_i}$ value guidance estimator

✅ $C_1(t_i), \gamma(t_i)$ guidance strength

➡️ $C_2(t_i)$, clipping regularization

These three knobs explain differences among methods.

Knob 3: Regularization

if (PPO-Clip condition):

$$\gamma(t_i) \leftarrow 0, C_2(t_i) \leftarrow 0$$

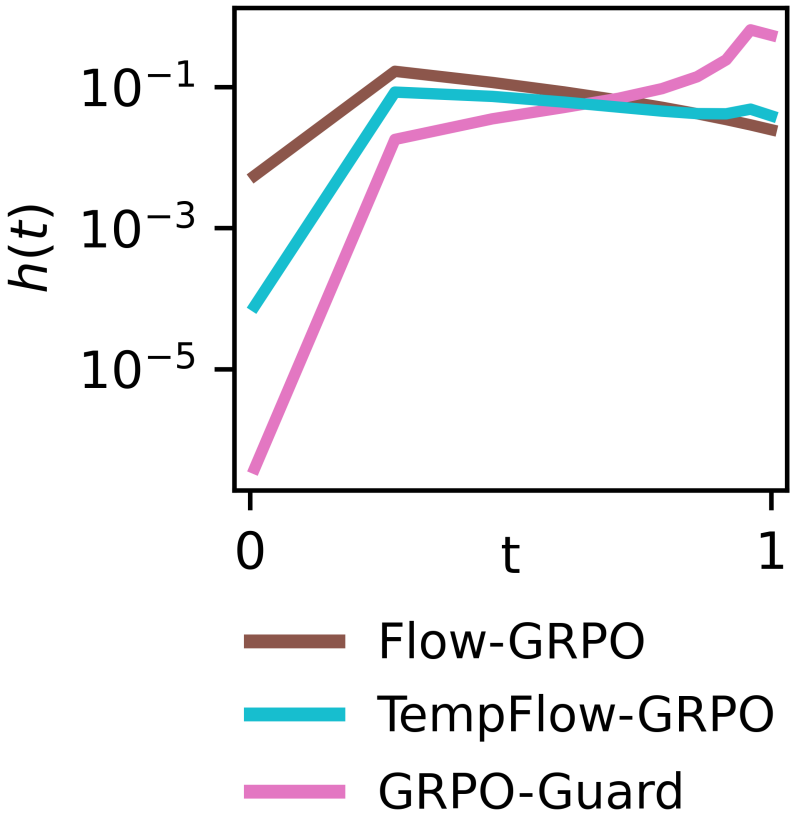
$$\mathcal{L}(\theta) = \mathbb{E}_{t_i, \boldsymbol{x}_{t_i}, \boldsymbol{\epsilon}} \left[C_1(t_i) \Big(\underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - \boldsymbol{s}^{\text{ref}}_{t_i} \|^2}_{\text{Anchor}} \Big) \right]$$

else:

$$\mathcal{L}(\theta) = \mathbb{E}_{t_i, \boldsymbol{x}_{t_i}, \boldsymbol{\epsilon}} \left[C_1(t_i) \Big(\underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - (\boldsymbol{s}^{\text{ref}}_{t_i} + \gamma(t_i) \hat{\boldsymbol{\Psi}}_{t_i}) \|^2}_{\text{Guidance}} + C_2(t_i) \underbrace{\| \boldsymbol{s}^{\theta}_{t_i} - \boldsymbol{s}^{\theta_{\text{old}}}_{t_i} \|^2}_{\text{Anchor}} \Big) \right]$$

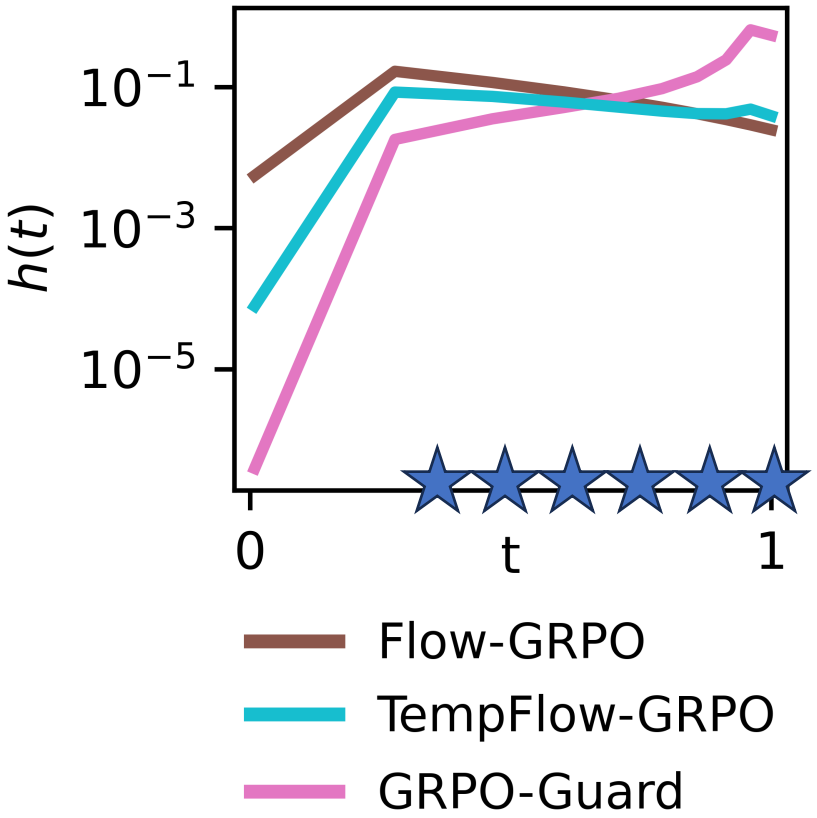
Off-policy updates are heuristically regularized.

Case Study: GRPO with GenEval



TempFlow-GRPO suppresses guidance at $t_i \approx 0$, but maintains $K_i = \text{const}$

Case Study: GRPO with GenEval

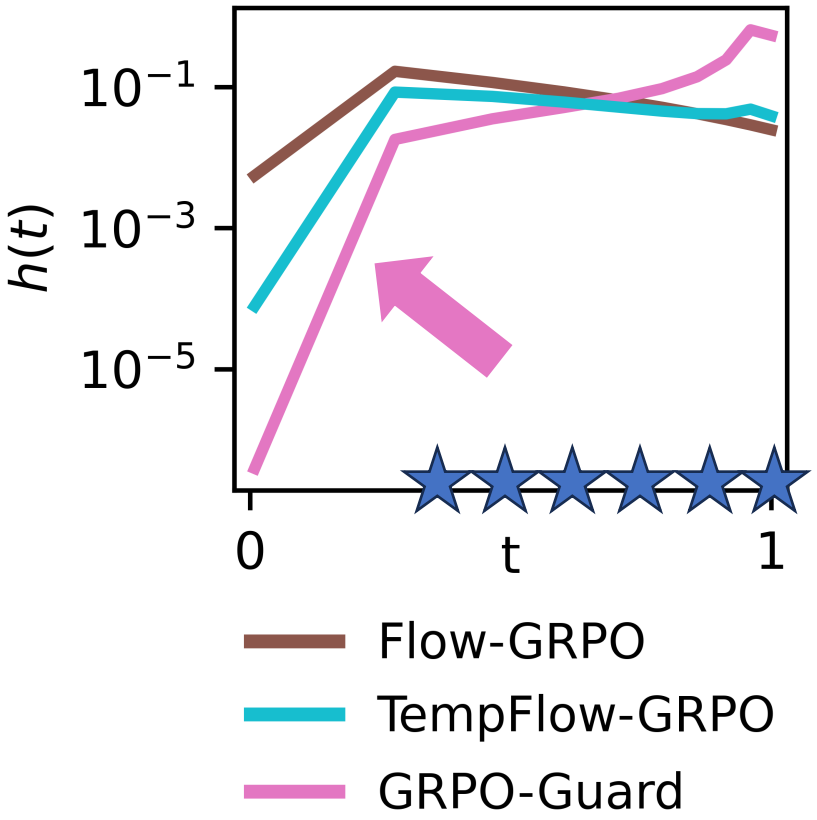


TempFlow-GRPO suppresses guidance at $t_i \approx 0$, but maintains $K_i = \text{const}$

Easy win

1. Concentrate $\{K_i\}$ on $\{t_i\}$ that matters

Case Study: GRPO with GenEval

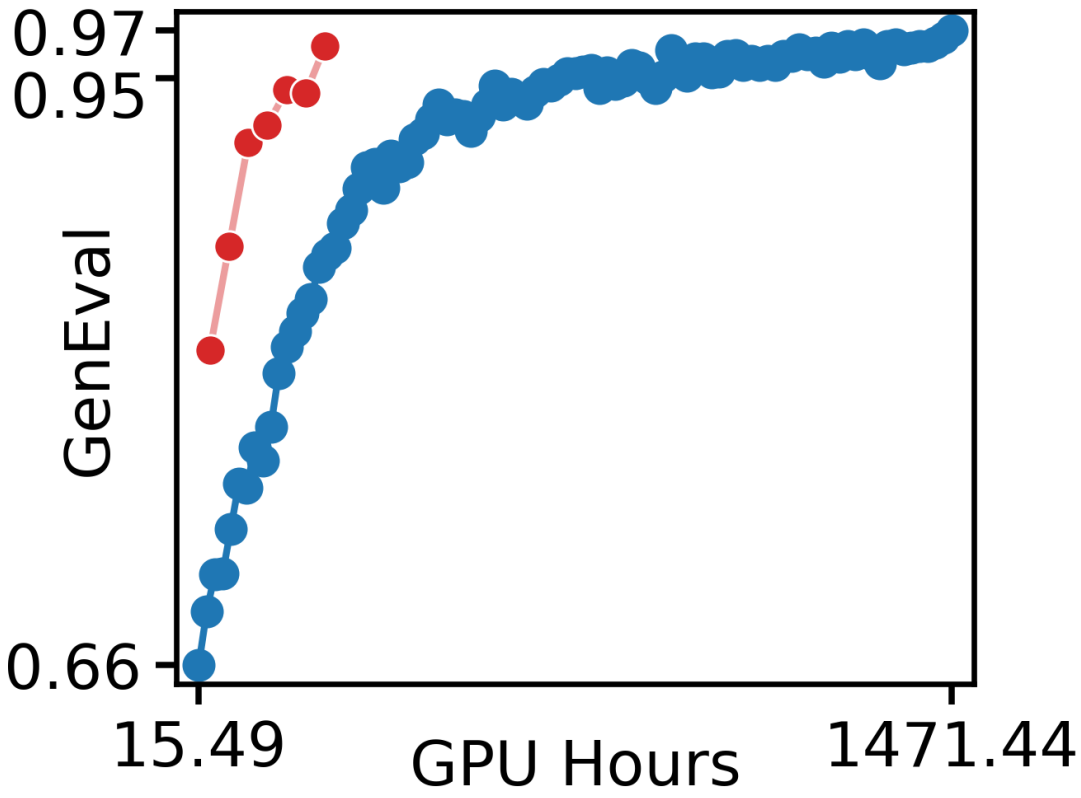


TempFlow-GRPO suppresses guidance at $t_i \approx 0$, but maintains $K_i = \text{const}$

Easy win

1. Concentrate $\{K_i\}$ on $\{t_i\}$ that matters
2. Double down on $h(t_i)$

Case Study: GRPO with GenEval



5×

wall-clock speedup over TempFlow-GRPO

GenEval = 0.97

RSM ⇒ straightforward redesigns ⇒ improve efficiency

Reward-weighted Regression

"RWR is an approximation to Reward-Weighted MLE"

Diffusion RL tutorial (Uehara et. al., 2024)

$$\mathcal{L}_{\mathrm{RSM}}(\theta) = \mathbb{E}\Big[C_1(t_i)\underbrace{\|\boldsymbol{s}^{\theta}_{t_i} - \boldsymbol{s}^{\star}_{t_i}\|^2}_{\text{Guidance}} + C_1(t_i)C_2(t_i)\underbrace{\|\boldsymbol{s}^{\theta}_{t_i} - \boldsymbol{s}^{\theta_{\mathrm{old}}}_{t_i}\|^2}_{\text{Regularization}}\Big] \qquad \boldsymbol{s}^{\star}_{t_i} \equiv \boldsymbol{s}^{\mathrm{ref}}_{t_i} + \gamma(t_i)\hat{\boldsymbol{\Psi}}_{t_i}$$

$$\mathcal{L}_{\mathrm{NFT}}(\theta) \equiv \mathbb{E}\Big[\underbrace{r(\boldsymbol{x}_0)\|\boldsymbol{v}^{\theta}_{t_i} - \boldsymbol{v}^{\mathrm{fwd}}_{t_i}\|^2}_{\text{RWR}} + (\beta - r(\boldsymbol{x}_0))\underbrace{\|\boldsymbol{v}^{\theta}_{t_i} - \boldsymbol{v}^{\theta_{\mathrm{old}}}_{t_i}\|^2}_{\text{Regularization}}\Big] \qquad \boldsymbol{v}^{\mathrm{fwd}}_{t_i} \equiv (\boldsymbol{\epsilon} - \boldsymbol{x}_0)$$

DiffusionNFT, AWM, RAM, ⋯ are closely related.

Takeaways

$$\arg\max_{q} \mathbb{E}_{x_0 \sim q}\Big[r(x_0)\Big] - \alpha \mathcal{D}_{\mathrm{KL}}\Big(q \| p^{\mathrm{ref}}\Big)$$

$$\mathcal{L}_{\mathrm{RSM}}(\theta) = \mathbb{E}_{t_i, \boldsymbol{x}_{t_i}, \boldsymbol{\epsilon}}\Big[C_1(t_i)\Big(\underbrace{\|\boldsymbol{s}^{\theta}_{t_i} - (\boldsymbol{s}^{\mathrm{ref}}_{t_i} + \gamma(t_i)\hat{\boldsymbol{\Psi}}_{t_i})\|^2}_{\text{Guidance}} + C_2(t_i)\underbrace{\|\boldsymbol{s}^{\theta}_{t_i} - \boldsymbol{s}^{\theta_{\mathrm{old}}}_{t_i}\|^2}_{\text{Regularization}}\Big)\Big]$$

Same problem, Same L_2 regression.

$\hat{\boldsymbol{\Psi}}_{t_i}$

$C_1(t_i), \gamma(t_i)$

$C_2(t_i)$, clipping

Understand three axes $\Rightarrow$ Simple & effective redesigns.




{jaylee2000, jinhojsk515,
jeongsol, jong.ye}@kaist.ac.kr

Project Page