# Evolution and Application of Diffusion Models (DMs) in Reinforcement Learning (RL)

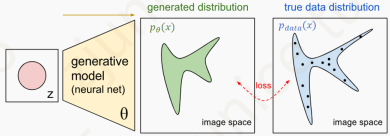
Networked Intelligence for Comprehensive Efficiency (NICE) Lab
College of Information Science and Electronic Engineering
Zhejiang University
http://nice.rongpeng.info/


Sep. 18, 2025

# Content

# Research Background & Preliminaries

# Deep Generative Models[1]




- **Maximizing Log-Likelihood**

$$\begin{aligned}\theta^* &= \arg\min_\theta D_{\mathrm{KL}}(p_{\mathrm{data}}(x) \,\|\, p_\theta(x)) \\ &= \arg\min_\theta \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}\left[\log p_{\mathrm{data}}(x) - \log p_\theta(x)\right] \\ &= \arg\max_\theta \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}\left[\log p_\theta(x)\right] \\ &\approx \arg\max_\theta \textstyle\sum_{i=1}^N \log p_\theta(x_i)\end{aligned}$$

- $p(x) = \int p(x,z)\, dz$: the marginalization integral over the latent variable $z$ is intractable in complex models
- $p(x) = \frac{p(x,z)}{q(z|x)}$: the posterior distribution is difficult to obtain directly

- **Optimizing Evidence Lower Bound (ELBO)**

$$\begin{aligned}\log p(x) &= \int q_\phi(z|x) \log p(x)\, dz && \text{(Bring Evidence into Integral)} \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log p(x)\right] && \text{(Definition of Expectation)} \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{p(z|x)}\right] && \text{(Apply Bayes Rule)} \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{p(z|x)}\frac{q_\phi(z|x)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p(z|x)}\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right] + D_{\mathrm{KL}}\left(q_\phi(z|x) \,\|\, p(z|x)\right) \\ &\geq \boxed{\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right]} && \text{(KL Divergence} \geq 0\text{)}\end{aligned}$$

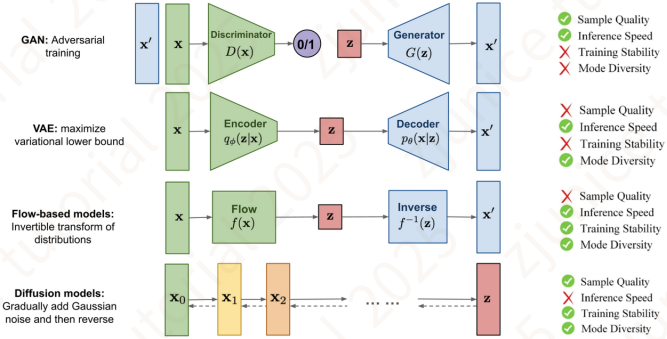
[1] L. Ruthotto and E. Haber, "An introduction to deep generative modeling," *GAMM-Mitteilungen*, vol. 44, no. 2, e202100008, 2021 .

# Deep Generative Models[2]

$$\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] \quad \text{(Chain Rule of Probability)}$$

$$= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] \quad \text{(Split the Expectation)}$$

$$= \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,\|\, p(\boldsymbol{z}))}_{\text{prior matching term}} \quad \text{(Definition of KL Divergence)}$$

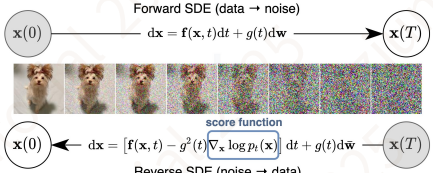



[2] Z. Ahmad et al., "Understanding gans: Fundamentals, variants, training challenges, applications, and open problems," *Multimedia Tools and Applications*, vol. 84, no. 12, pp. 10 347–10 423, 2025 .

DDIM

# Diffusion Models[3,4]

## Score-Based




- Forward SDE:

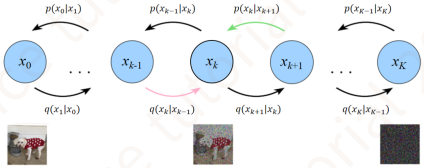
$$\mathrm{d}\boldsymbol{x}_t = f(\boldsymbol{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{w}_t$$

- Reverse SDE:

$$\mathrm{d}\boldsymbol{x}_t = \left[f(\boldsymbol{x}_t, t) - g^2(t)\,\boxed{\nabla \log p_t(\boldsymbol{x}_t)}\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\boldsymbol{w}}_t$$

$$\theta^* = \arg\min_\theta \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_t}\left[\lambda(t)\,\|\mathbf{s}_\theta(\boldsymbol{x}_t, t) - \nabla \log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2\right]$$

## DDPM




- Forward diffusion process:

$$q(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) \sim \mathcal{N}\left(\sqrt{\alpha_k}\,\boldsymbol{x}_{k-1},\ (1-\alpha_k)\boldsymbol{I}\right)$$
$$q(\boldsymbol{x}_k \mid \boldsymbol{x}_0) \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_k}\,\boldsymbol{x}_0,\ (1-\bar{\alpha}_k)\boldsymbol{I}\right)$$

- Reverse denoising process:

$$q(\boldsymbol{x}_{k-1} \mid \boldsymbol{x}_k, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}, \boldsymbol{x}_0)\,q(\boldsymbol{x}_{k-1} \mid \boldsymbol{x}_0)}{q(\boldsymbol{x}_k \mid \boldsymbol{x}_0)}$$
$$\sim \mathcal{N}\left(\frac{1}{\sqrt{\alpha_k}}\boldsymbol{x}_k - \frac{1-\alpha_k}{\sqrt{\alpha_k}\sqrt{1-\bar{\alpha}_k}}\,\boxed{\boldsymbol{\epsilon}_0}, \sigma_k^2\boldsymbol{I}\right)$$

$$\theta^* = \arg\min_\theta \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_k}\left[\lambda(k)\,\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_k, k) - \boldsymbol{\epsilon}_0\|_2^2\right]$$

[3] Y. Song et al., "Score-based generative modeling through stochastic differential equations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Virtual Edition, 2021.

[4] J. Ho et al., "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, Virtual Edition, 2020.

# Diffusion Models[5]

- **Feature 1: Reparameterization**

  Rewrite a random variable as a deterministic function of a noise variable. This enables gradient-based optimization of non-stochastic terms.

$$z \sim \mathcal{N}(z; \mu_\theta, \sigma_\theta^2 I)$$

$$\Downarrow$$

$$z = \mu_\theta + \sigma_\theta \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

- **Feature 2: $x_k$ can be directly represented by $x_0$**

  Forward Process:

$$\begin{aligned} x_k &= \sqrt{\alpha_k} x_{k-1} + \sqrt{1-\alpha_k}\,\epsilon_{k-1} \\ &= \sqrt{\alpha_k \alpha_{k-1}} x_{k-2} + \sqrt{1-\alpha_k \alpha_{k-1}}\,\epsilon_{k-2} \\ &= \cdots \\ &= \sqrt{\bar{\alpha}_k}\, x_0 + \sqrt{1-\bar{\alpha}_k}\,\epsilon_0 \quad \bar{\alpha}_k = \textstyle\prod_{i=1}^{k} \alpha_i \end{aligned}$$

  Reconstruct:

$$x_0 = \frac{x_k - \sqrt{1-\bar{\alpha}_k}\,\epsilon_0}{\sqrt{\bar{\alpha}_k}}$$

[5] J. Ho et al., "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, Virtual Edition, 2020.

# Denoising Diffusion Implicit Model (DDIM)[6]


Figure 1: Graphical models for diffusion (left) and non-Markovian (right) inference models.

## Non-Markovian sampling process

$$q_\sigma(x_{k-1} \mid x_k, x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{k-1}}x_0 + \sqrt{1-\bar{\alpha}_{k-1}-\sigma_k^2}\,\frac{x_k - \sqrt{\bar{\alpha}_k}\,x_0}{\sqrt{1-\bar{\alpha}_k}}, \sigma_k^2 \boldsymbol{I}\right)$$

The mean is chosen to ensure that

$$q_\sigma(x_k \mid x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_k}\,x_0,\ (1-\bar{\alpha}_k)\,\boldsymbol{I}\right)$$

- When $\sigma \to 0$, $x_{k-1}$ is uniquely determined by $x_0$ and $x_k$.
- When $\sigma_k = \sqrt{\frac{1-\bar{\alpha}_{k-1}}{1-\bar{\alpha}_k}(1-\alpha_k)}$, the generalized sampler is equivalent to DDPM.

## Generalized generative process

Reconstruct: $\hat{x}_0 = \dfrac{x_k - \sqrt{1-\bar{\alpha}_k}\,\epsilon_\theta(x_k, k)}{\sqrt{\bar{\alpha}_k}}$

$$x_{k-1} = \sqrt{\bar{\alpha}_{k-1}}\,\frac{x_k - \sqrt{1-\bar{\alpha}_k}\,\epsilon_\theta(x_k,k)}{\sqrt{\bar{\alpha}_k}} + \sqrt{1-\bar{\alpha}_{k-1}-\sigma_k^2}\,\epsilon_\theta(x_k,k) + \sigma_k\,\epsilon_k$$

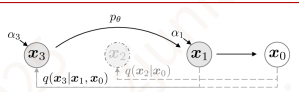
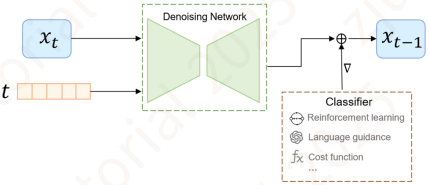



Figure 2: Graphical model for accelerated generation, where $\tau = [1, 3]$.

[6] J. Song et al., "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Virtual Edition, 2021.

# Conditional Diffusion Model[7,8]

## Classifer Guidance

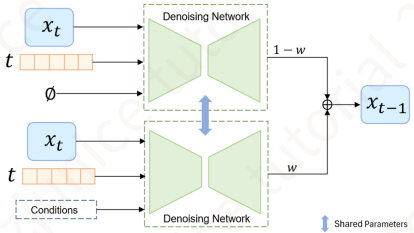



$$\nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \nabla \log p_t(\boldsymbol{x}_t) + \nabla \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$$

$$\boldsymbol{x}_{t-1} \sim \mathcal{N}\left(\mu_\theta(\boldsymbol{x}_t, t) + w \boxed{\nabla_{\boldsymbol{x}_t} \log p_\phi(\boldsymbol{y}|\boldsymbol{x}_t, t)}, \sigma_t^2 \boldsymbol{I}\right)$$

- Stronger control
- Higher sample quality

## Classifier-free Guidance



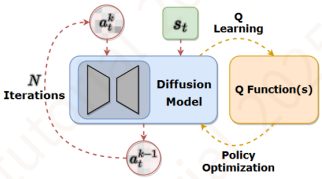

$$\tilde{\boldsymbol{\epsilon}}_t = w\,\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,\ t,\ \boldsymbol{y}) + (1-w)\,\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,\ t,\ \varnothing)$$

- No need for extra classifier
- More stable and flexible
- Preserves sample diversity

[7] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, Virtual Edition, 2021.

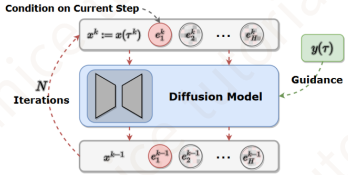
[8] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022 .

# Different Roles of DMs in RL[9]




**DM as policy** ⟺ **MFRL**

- Expressive multimodal policy parameterization
- Improved exploration and diversity
- Robustness to perturbations through iterative denoising

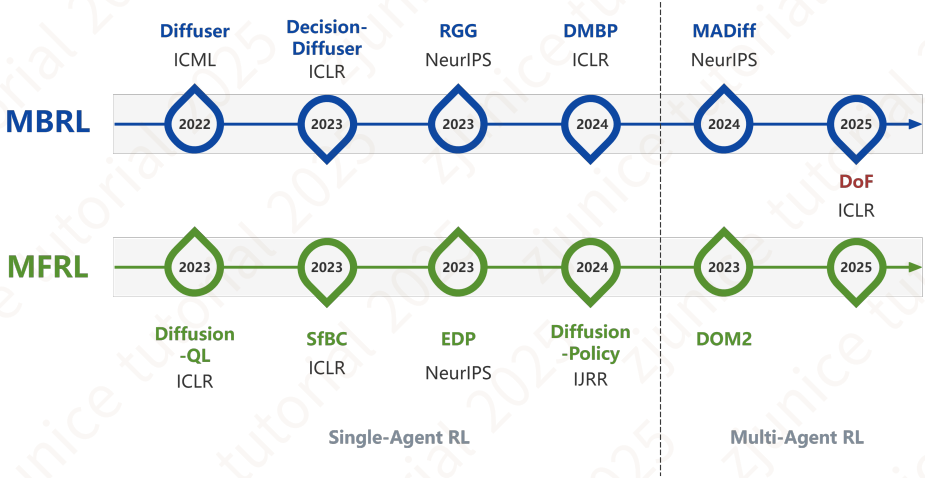



**DM as planner** ⟺ **MBRL**

- Global optimization from a long-term perspective
- Flexibility in multimodal trajectory generation
- Enhanced robustness through iterative refinement

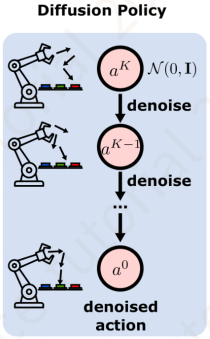
[9] Z. Zhu et al., "Diffusion models for reinforcement learning: A survey," *arXiv preprint arXiv:2311.01223*, 2023 .

# Algorithmic Timeline of DMs in RL



# DMs in Single-Agent RL

# Diffusion-QL[10]




- Behavior-cloning loss

$$\mathcal{L}_d(\theta) = \mathbb{E}_{k,\,\epsilon,\,(s,a)}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_k}\,\boxed{a} + \sqrt{1-\bar{\alpha}_k}\,\epsilon,\ \boxed{s},\ k\right)\right\|^2\right]$$

- Q-value function loss

$$\mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{D},\, a^0_{t+1}\sim\pi_{\theta'}}\left[\left\|\left(r(s_t,a_t)+\gamma \min_{i=1,2} Q_{\phi_i'}(s_{t+1}, a^0_{t+1})\right) - Q_{\phi_i}(s_t,a_t)\right\|^2\right]$$

$$\pi = \arg\min_{\pi_\theta} \mathcal{L}(\theta) = \mathcal{L}_d(\theta) + \mathcal{L}_q(\theta)$$
$$= \underbrace{\mathcal{L}_d(\theta)}_{\text{policy regularization}} - \alpha \cdot \underbrace{\mathbb{E}_{s\sim D,\, a^0\sim\pi_\theta}\left[Q_\phi(s, a^0)\right]}_{\text{policy improvement}}$$

**$\frac{\partial Q}{\partial a}$ is backpropagated through the whole diffusion chain**

[10] Z. Wang et al., "Diffusion policies as an expressive policy class for offline reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Kigali, Rwanda, 2023.

# Efficient Diffusion Policy (EDP)[11]

- Action reconstruction

  **Forward:** $\boldsymbol{x}_k = \sqrt{\bar{\alpha}_k}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}_0$

  $$\hat{a}_0 = \frac{1}{\sqrt{\bar{\alpha}_k}}\, a_k - \frac{\sqrt{1-\bar{\alpha}_k}}{\sqrt{\bar{\alpha}_k}}\, \epsilon_\theta(a_k,\ k,\ s)$$

- Policy improvement

  $$L_\pi(\theta) = -\,\mathbb{E}_{s\sim\mathcal{D},\,\hat{a}_0}\left[\, Q_\phi(s,\ \hat{a}_0)\,\right]$$

**Direct policy optimization:**

$$\nabla_\theta L_\pi(\theta) = -\,\frac{\partial Q_\phi(s,a)}{\partial a}\,\frac{\partial a}{\partial \theta}$$

$\frac{\partial a}{\partial \theta}$ is tractable

**Likelihood-based policy optimization:**

$\log \pi_\theta(a|s)$ is intractable

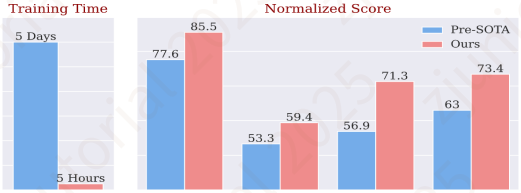
$$\max_\theta\ \mathbb{E}_{(s,a)\sim D}\left[\, f(Q_\phi(s,a))\,\log \pi_\theta(a\mid s)\,\right]$$

$$\Leftrightarrow\ \mathbb{E}_{k,\varepsilon,(a,s)}\left[\frac{\beta^k\, f(Q_\phi(s,a))}{2\,\alpha^k(1-\bar{\alpha}^{k-1})}\,\|\,\varepsilon-\varepsilon_\theta(a^k,k;\,s)\,\|^2\right]$$

Optimizing the ELBO in DDPM

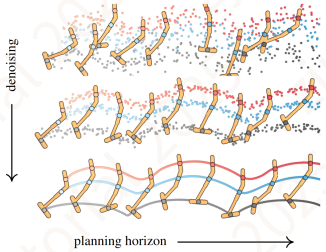
$$\Leftrightarrow\ \mathbb{E}_{k,\varepsilon,(a,s)}\left[\, f(Q_\phi(s,a))\,\|\,a-\hat{a}^0\|^2\right]$$

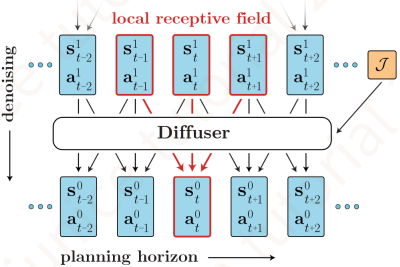
Policy approximation $\hat{\pi}_\theta(a|s) \triangleq \mathcal{N}(\hat{a}^0, I)$




[11] B. Kang et al., "Efficient diffusion policies for offline reinforcement learning," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, New Orleans, Louisiana, USA, 2023.

# Diffuser[12]




- Diffusing over state-action trajectory

$$\boldsymbol{x}_k = \begin{bmatrix} s_0 & s_1 & \cdots & s_H \\ a_0 & a_1 & \cdots & a_H \end{bmatrix}$$

- Guiding with classifier guidance

$$p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k,\ \boldsymbol{y}) \approx \mathcal{N}(\boldsymbol{x}_{k-1};\ \boldsymbol{\mu}_\theta + \boldsymbol{\Sigma}\boxed{\boldsymbol{g}},\ \boldsymbol{\Sigma})$$

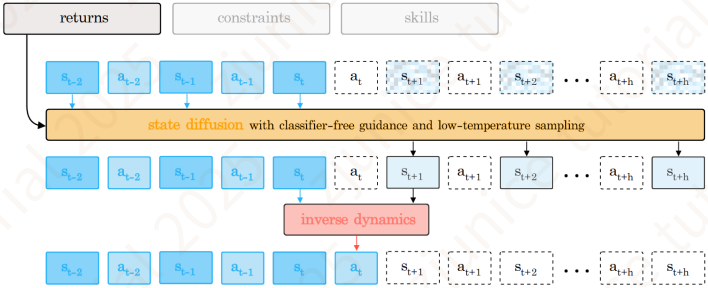
$$\boldsymbol{g} = \nabla_{\boldsymbol{\tau}} \log p(\boldsymbol{y} \mid \boldsymbol{\tau})\Big|_{\boldsymbol{\tau}=\boldsymbol{\mu}_\theta} = \nabla \mathcal{J}(\boldsymbol{\mu}_\theta)$$

## Limitations

- ▷ Actions are harder to predict and model
- ▷ Requires estimating a Q-function

[12] M. Janner et al., "Planning with diffusion for flexible behavior synthesis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Baltimore, Maryland, USA, 2022.

# Decision-Diffuser[13]




- Diffusing over state trajectory

$$\boldsymbol{x}_k(\tau) = \left(s_t,\ s_{t+1},\ \ldots,\ s_{t+H-1}\right)_k$$

- Acting with inverse-dynamics

$$a_t = f_\phi\left(s_t,\ s_{t+1}\right)$$

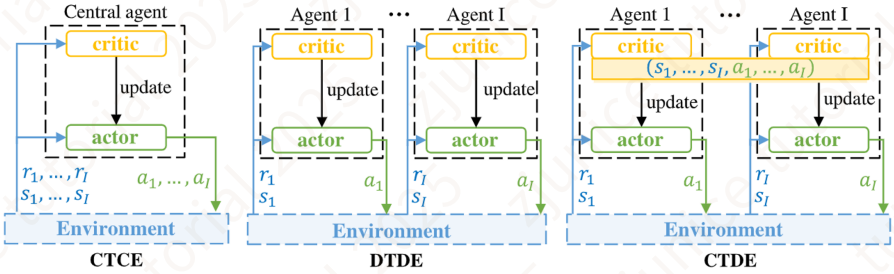
- Guiding with classifier-free guidance

$$\hat{\epsilon} = \epsilon_\theta\left(\boldsymbol{x}_k(\tau),\ \varnothing,\ k\right) + \omega\Big(\epsilon_\theta\left(\boldsymbol{x}_k(\tau),\ \boldsymbol{y}(\tau),\ k\right) - \epsilon_\theta\left(\boldsymbol{x}_k(\tau),\ \varnothing,\ k\right)\Big)$$

[13] A. Ajay et al., "Is conditional generative modeling all you need for decision-making?" In *Proc. Int. Conf. Learn. Represent. (ICLR)*, Kigali, Rwanda, 2023.

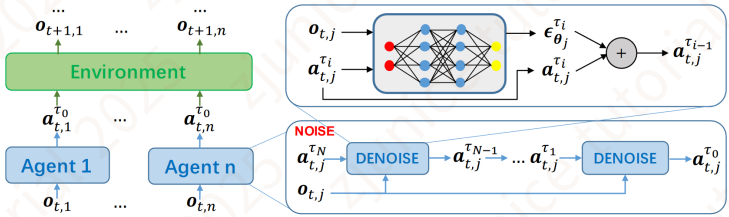
# DMs in Multi-Agent RL

# DMs in Multi-Agent RL




- MADiff-C
- DOM2
- MAD2RL
- MADiff-D
- DoF

# DOM2[14]




## Diffusion-QL-based policy update

- DTDE: agent $j$ learns and runs its own policy $\theta_j$

$$\mathcal{L}_{\text{bc}}(\theta_j) = \mathbb{E}_{(o_{t,j}, a^0_{t,j}),\ \epsilon,\ k}\left[\left\|\epsilon - \epsilon_{\theta_j}\left(a^k_{t,j},\ o_{t,j},\ k\right)\right\|_2^2\right]$$

- Sample based on 1-order DPM-solver (DDIM)

$$a^{k-1}_{t,j} = \frac{\bar{\alpha}_{k-1}}{\bar{\alpha}_k} a^k_{t,j} - \sigma_k\left(\frac{\bar{\alpha}_k \sigma_{k-1}}{\bar{\alpha}_{k-1}\sigma_k} - 1\right)\epsilon_{\theta_j}$$

## CQL-based Q-value update

$$\mathcal{L}(\phi_j) = \mathbb{E}_{(o_j, a_j)\sim\mathcal{D}_j}[(Q_{\phi_j}(o_j, a_j) - y_j)^2]$$

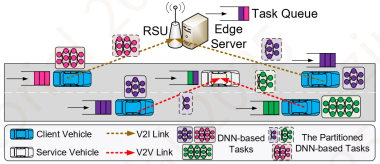
$$+ \zeta\mathbb{E}_{(o_j, a_j)\sim\mathcal{D}_j}[\log \sum_{\tilde{a}_j} \exp(Q_{\phi_j}(o_j, \tilde{a}_j)) - Q_{\phi_j}(o_j, a_j)]$$

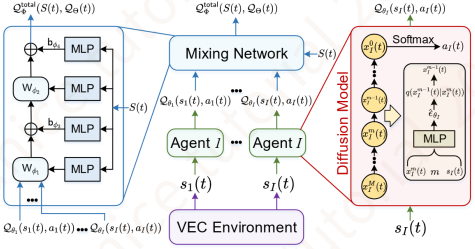
suppress the unseen state-action pairs

encourage the state-action pairs from the dataset

[14] Z. Li et al., "Beyond conservatism: Diffusion policies in offline multi-agent reinforcement learning," *arXiv preprint arXiv:2307.01472, 2023* .

# MAD2RL[15]




Incorporate DMs to determine the optimal DNN partitioning and task offloading decisions in Vehicular Edge Computing (VEC).

- DM-based action making

$$Q_{\theta_i}^{u}\big(s_i(t)\big) = \frac{e^{x_i^{0,u}(t)}}{\sum_{\nu} e^{x_i^{0,\nu}(t)}}, \qquad a_i(t) = \arg\max_{u} \; Q_{\theta_i}^{u}\big(s_i(t)\big)$$

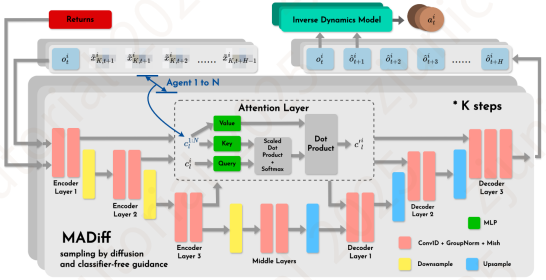
- QMIX-based policy improvement

$$\boldsymbol{Q}_{\Theta}(t) = \Big\{ Q_{\theta_1}\big(s_1(t), a_1(t)\big), \ldots, Q_{\theta_N}\big(s_N(t), a_N(t)\big) \Big\}$$

$$\boldsymbol{Q}_{\Phi}^{\text{total}} = \text{MixingNET}\big(\boldsymbol{S}(t),\ \boldsymbol{Q}_{\Theta}(t)\big)$$

[15] Z. Liu et al., "Dnn partitioning, task offloading, and resource allocation in dynamic vehicular networks: A lyapunov-guided diffusion-based reinforcement learning approach," *IEEE Trans. Mob. Comput.*, pp. 1–17, 2024 .

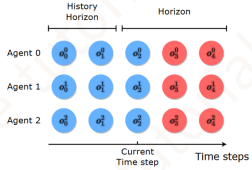
# MADiff[16]



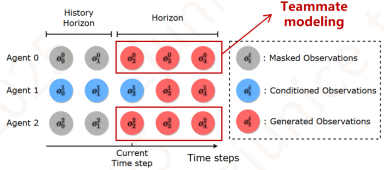

- **Information interchange**
  Multi-head attention on the skip-connected feature from symmetric $l$-th encoder layer of agent $i$: $\mathbf{c}_l^i$.
- **Centralized training**

$$\mathcal{L}(\theta,\phi) := \sum_i \mathbb{E}_{(s^i,a^i,s'^i)\in\mathcal{D}}[\|a^i - I_\phi^i(o^i,o'^i)\|^2] \quad \text{individual inverse dynamics model}$$
$$+ \mathbb{E}_{k,\boldsymbol{\tau}_0\in\mathcal{D},\beta}[\|\epsilon - \epsilon_\theta(\hat{\boldsymbol{\tau}}_k,(1-\beta)\boldsymbol{y}(\boldsymbol{\tau}_0)+\beta\emptyset,k)\|^2] \quad \text{unified diffusion noise model}$$

- **Execution**



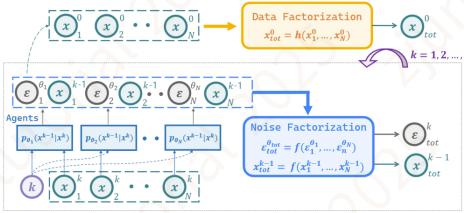

(a) MADIFF in centralized control.




(b) MADIFF in decentralized execution.

[16] Z. Zhu et al., "Madiff: Offline multi-agent learning with diffusion models," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, Vancouver, CANADA, 2024.

# DoF[17]




## Diffusion factorization

Factorize the joint noise and joint data as

$$\begin{aligned}
\boldsymbol{\epsilon}^k_{tot} &= f(\boldsymbol{\epsilon}^k_1, ..., \boldsymbol{\epsilon}^k_N) \quad 0 \le k \le K \\
\boldsymbol{\epsilon}^{\theta_{tot}}_{tot}(\boldsymbol{x}^k_{tot}, k) &= f(\boldsymbol{\epsilon}^{\theta_1}_1(\boldsymbol{x}^k_1, k), ..., \boldsymbol{\epsilon}^{\theta_N}_N(\boldsymbol{x}^k_N, k)) \quad 0 \le k \le K \\
\boldsymbol{x}^k_{tot} &= f(\boldsymbol{x}^k_1, ..., \boldsymbol{x}^k_N) \quad 1 \le k \le K \\
\boldsymbol{x}^0_{tot} &= h(\boldsymbol{x}^0_1, ..., \boldsymbol{x}^0_N)
\end{aligned}$$

- **Individual-Global-Max (IGM)**

$$\arg\max_{\boldsymbol{u}} Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{u}) = \begin{pmatrix} \arg\max_{u_1} Q_1(\tau_1, u_1) \\ \vdots \\ \arg\max_{u_N} Q_N(\tau_N, u_N) \end{pmatrix}$$

- **Individual-Global-Identically-Distributed (IGD)**

Assume individual marginals $p_{\theta_i}(x^0_i) = \int p_{\theta_i}(x^{0:K}_i)\, \mathrm{d}x^{1:K}_i$, such that

$$\prod_{i=1}^N p_{\theta_i}(x^0_i) = p_{\theta_{\text{tot}}}(x^0_{\text{tot}}), \quad \theta_i \subset \theta_{\text{tot}}.$$

Then the generated samples $\{x^0_i\}_{i=1}^N$ share the same distribution as $x^0_{\text{tot}}$.

[17] C. Li et al., "Dof: A diffusion factorization framework for offline multi-agent reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Singapore, 2025.

# DoF

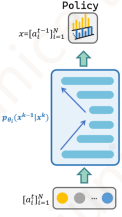
- **Concat**
  $x_{\text{tot}}\big[(i-1)\times d : i\times d\big] = x_i$
- **WConcat**
  $x_{\text{tot}}\big[(i-1)\times d : i\times d\big] = k_i\, x_i$
- **Atten**
  $x_{\text{tot}}\big[(i-1)\times d : i\times d\big] = \sum_{j=1}^{N} w_i^j\, x_j$
- **QMIX**
  $x_{tot} = h(x_1, x_2, \cdots, x_n)$

**Algorithm 1** Centralized Training

**repeat**
  $\mathbf{x}_{\text{tot}}^0 \sim q(\mathbf{x}_{\text{tot}})$ *(Sample global data)*
  $k \sim \text{Uniform}(\{1, \dots, K\})$ *(Diffusion step)*
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \in \mathbb{R}^{d\times N}$ *(Sample global noise)*
  $\mathbf{x}_{\text{tot}}^k = \sqrt{\bar{\alpha}^k}\mathbf{x}_{\text{tot}}^{k-1} + \sqrt{1-\bar{\alpha}^k}\epsilon$
  $\mathbf{x}_i^k = \mathbf{x}_{\text{tot}}^k[(i-1)\times d : i\times d], i \in [1, \dots, N]$
  $\epsilon_{\text{tot}} = f(\epsilon_{\theta_1}^1(\mathbf{x}_1^k, k), \dots, \epsilon_{\theta_N}^N(\mathbf{x}_N^k, k))$
  Take gradient descent step on:
$$\nabla_\theta \|\epsilon - \epsilon_{\text{tot}}\|^2$$
**until** convergence

**Algorithm 2** Decentralized Execution

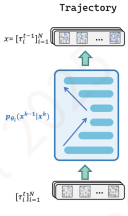
$\mathbf{x}_i^K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ *(Initialize for each agent i)*
**for** $k = K, \dots, 1$ **do**
  $\epsilon_\theta^i(\mathbf{x}_i^k, k)$ *(Noise prediction by each agent i)*
  Update state for each agent $i$:
$$\mathbf{x}_i^{k-1} = \frac{1}{\sqrt{\alpha_k}}\left(\mathbf{x}_i^k - \frac{1-\alpha_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon_\theta^i(\mathbf{x}_i^k, k)\right) + \sigma_k \mathbf{z},$$
  where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $k > 1$, else $\mathbf{z} = \mathbf{0}$.
**end for**
**return** $\mathbf{x}_i^0$ *(Final trajectory or action for each agent i)*




## DoF as policy

Follow the design of Diffusion-QL:

$$\mathcal{L} = \mathcal{L}_{\text{diff}}(\theta) + \mathcal{L}_{\text{pg}}(\theta)$$
$$= \mathbb{E}\left[\left\|\epsilon - \epsilon_\theta^{\text{tot}}\right\|^2\right] - \alpha\, \mathbb{E}\left[Q_\phi\left(s, \boldsymbol{u}^0\right)\right]$$




## DoF as planner

Follow the design of Decision-Diffuser:

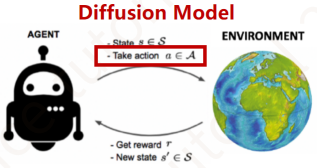
$$\mathcal{L} = \mathbb{E}\left[\left\|\epsilon - \epsilon_\theta^{\text{tot}}(\boldsymbol{x}_k, (1-\beta)\boldsymbol{y} + \beta\varnothing, k)\right\|^2\right]$$
$$+ \mathbb{E}\left[\left\|\boldsymbol{u} - D_\phi(\boldsymbol{o}, \boldsymbol{o}')\right\|^2\right]$$

# Conclusion & Future Prospect

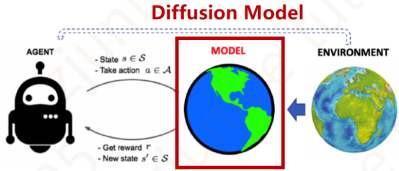
# Conclusion

I have talked about

- Fundamentals of Diffusion Models
  - Superior generative modeling power over other deep generative models
  - Mathematical foundations of diffusion models
  - Sampling acceleration techniques for diffusion models
  - Conditional generative modeling with diffusion models
- Applications of Diffusion Models in Reinforcement Learning
  - Diffusion models as policy




  - Diffusion models as planner




  - Diffusion models as a unified framework

# Future Prospect

- **Fast Inference and Online Adaptation of DMs**
  DMs require multi-step iterative denoising, which leads to low inference efficiency and limited real-time scalability.
  DDIM; DPM-Solver; DPM-Solver++; Flow; Consistency Model ...
- **Optimize the Architecture of the Noise Model**
  Most current noise models adopt simple backbones like MLPs or U-Nets, which struggle to capture the complex interactions among agents.
  **Attention mechanism** (enhance inter-agent interaction);
  **LSTM** (better historical insight) ...
- **Generalization to Multi-Task Scenarios**
  DMs can be post-trained to adapt to varying tasks and environments, enabling better generalization in multi-task settings.
  Importance sampling; beam search; conditional guidance; RL-based fine-tuning ...

# Thank you

Networked Intelligence for Comprehensive Efficiency (NICE) Lab
College of Information Science and Electronic Engineering
Zhejiang University
https://nice.rongpeng.info