DoF : A Diffusion Factorization Framework for Offline Multi-Agent Reinforcement Learning

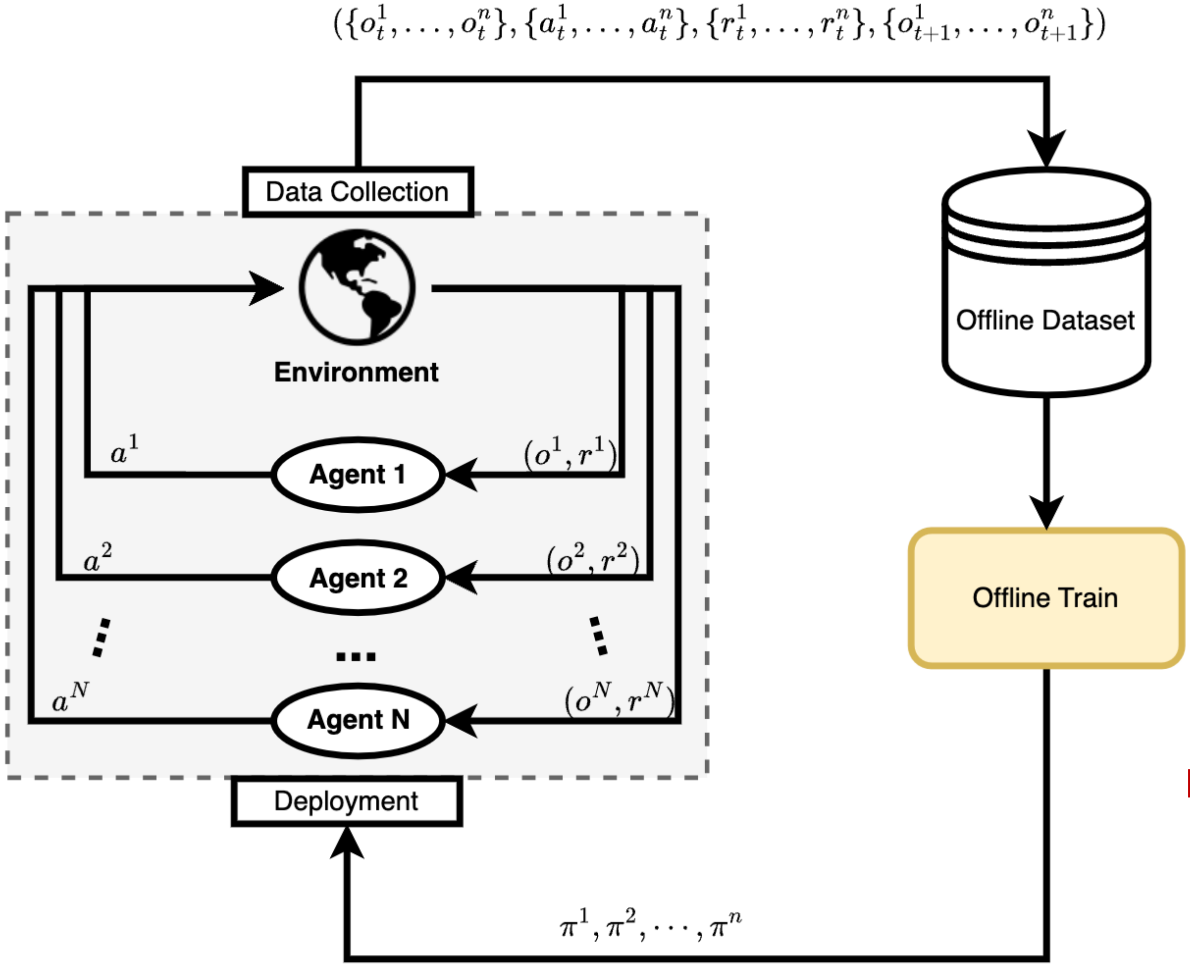
Chao Li[ab*], Ziwei Deng[ab*], Chenxing Lin[ab], Wenqi Chen[e],
Yongquan Fu[c], Weiquan Liu[abd], Chenglu Wen[ab],
Cheng Wang[ab], Siqi Shen[ab†]

chaoli@stu.xmu.edu.cn, dengziwei@stu.xmu.edu.cn, lincx1123@stu.xmu.edu.cn,
siqishen@xmu.edu.cn

[a] Xiamen University
[b] National University of Defense Technology
[c] Jimei University
[d] University of Electronic Science and Technology of China

ICLR 2025

Offline Multi-Agent Reinforcement Learning



Learning Objective:

In Offline Dataset $\mathcal{D}$

⬇

policy $\boldsymbol{\pi} = (\pi^1, \pi^2, \cdots, \pi^n)$

⬇

Maximize long-term return

$$J(\pi) = \mathbb{E}_{\pi,\mathcal{D}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \mathbf{a}_t)\right]$$

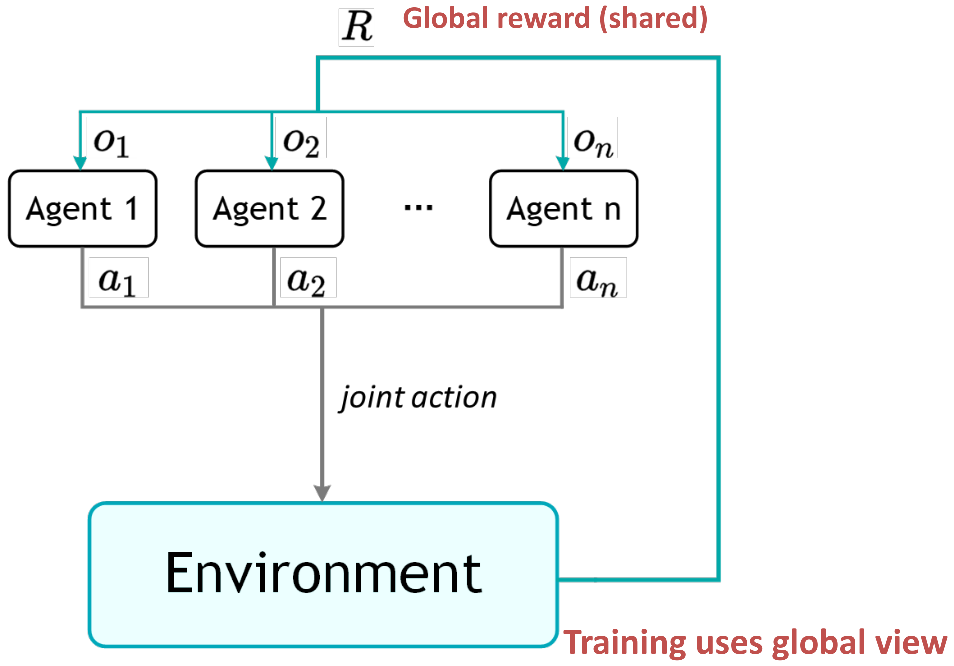
IGM principle and CTDE

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

$$\arg\max_{\mathbf{u}} Q_{jt}(\tau, u) = \begin{pmatrix} \arg\max_{\mathbf{u}_1} Q_1(\tau_1, u_1) \\ \vdots \\ \arg\max_{\mathbf{u}_n} Q_n(\tau_n, u_n) \end{pmatrix}$$

This principle ensures that the global joint action can be decomposed into local greedy actions over individual Q-functions.

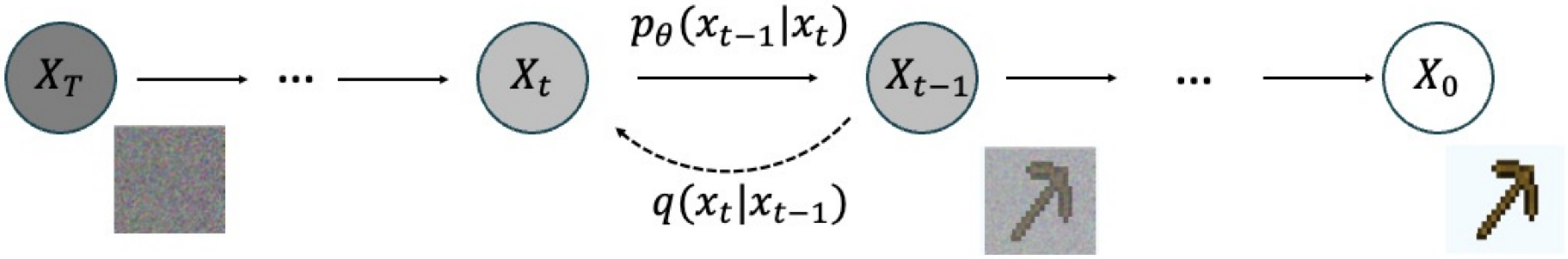
- ***Centralized Training with Decentralized Execution paradigm (CTDE)***



CTDE enables centralized value learning during training, while allowing fully decentralized execution during deployment.

Diffusion Model

A **diffusion model** generates data by learning to reverse a step-by-step noising process, starting from pure noise and recovering meaningful samples.



Forward Diffusion Process

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) := \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}), \quad q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) := \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t \boldsymbol{I}\right).$$

Reverse Denoising Process

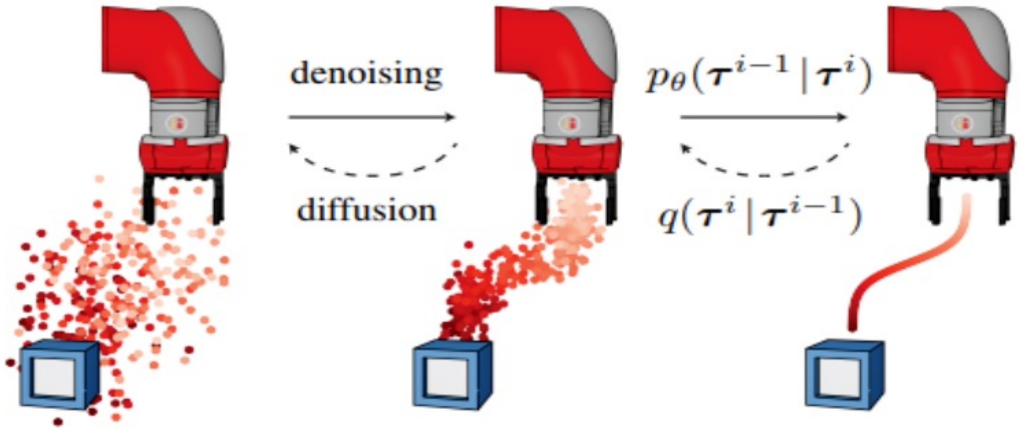
$$p_\theta(\boldsymbol{x}_{0:T}) := \mathcal{N}(\boldsymbol{x}_T; \mathbf{0}, \boldsymbol{I}) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$$

Ho et al. Denoising diffusion probabilistic models. In NeurIPS, 2020.

Diffusion Model in RL

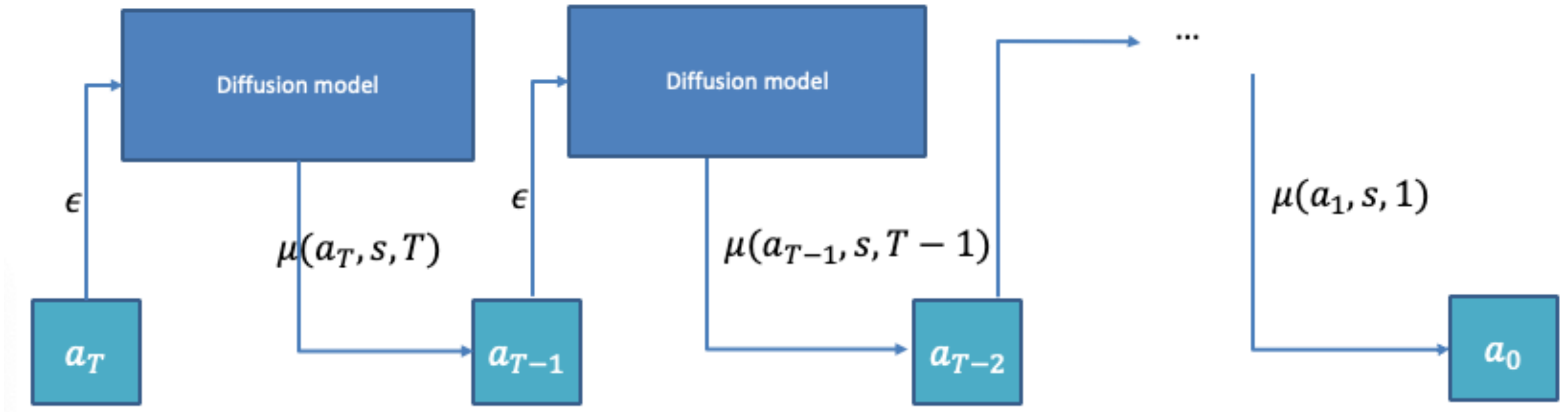
- **Trajectory-based Diffusion Models**

  Model entire trajectories as sequences to imitate expert behavior; good for capturing long-term dependencies but hard to scale in multi-agent settings.



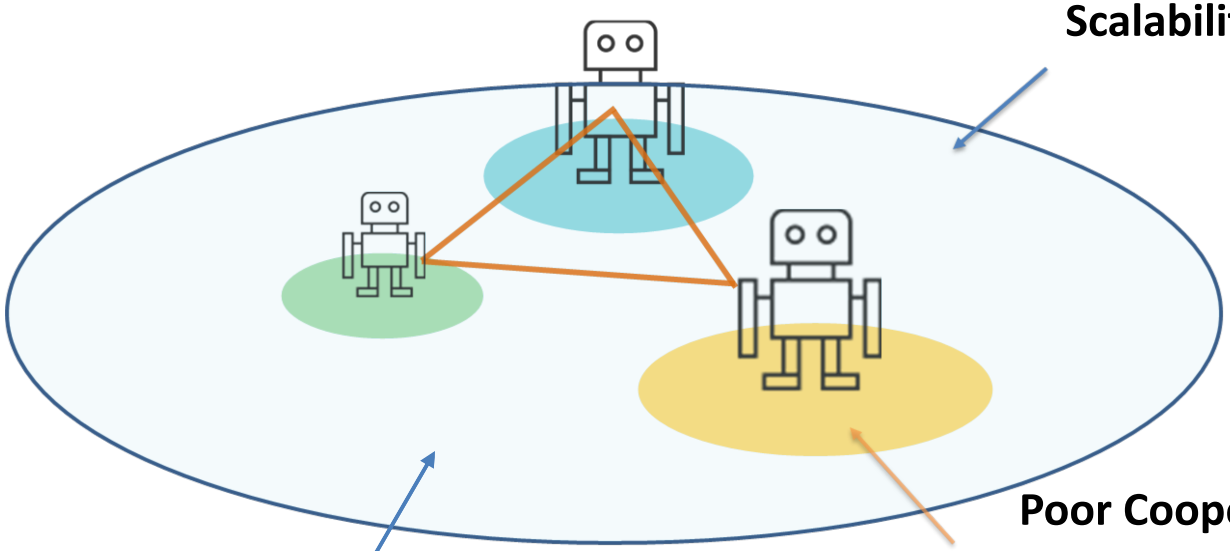
- **Policy-based Diffusion Models**

  Generate actions step-by-step conditioned on states; more scalable and flexible, but slower due to iterative sampling.



Janner, Michael, et al. Planning with diffusion for flexible behavior synthesis. ICML 2022
Zhendong Wang, et al. Diffusion policies as an expressive policy class for offline reinforcement learning. ICLR 2023

Challenge for Diffusion Model in MARL



Scalability Issue

The joint state-action space grows exponentially with the number of agents, making centralized modeling in offline MARL computationally infeasible.

Poor Cooperation Modeling

Independent policy learning fails to capture inter-agent dependencies, leading to suboptimal or conflicting behaviors in cooperative tasks.

Theoretical Vacuum in Diffusion-based MARL

IGM works for value-based methods, but not for diffusion models.
Is there an equivalent principle tailored for diffusion models in multi-agent settings?

Motivation

- How to design a **diffusion-based framework** that：
  - Enables decentralized execution
  - Encourages cooperation
  - Has **theoretical foundation**

DoF: A Diffusion Factorization Framework built on the **IGD Principle**

IGD principle

Individual-Global-identically-Distributed (IGD) principle:

Definition 2 (IGD). *For a joint total distribution* $p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^0) := \int p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^{0:K}) d\boldsymbol{x}_{tot}^{1:K}$*. which is called the reverse process, defined as a Markov chain* $p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^{0:K}) := p(\boldsymbol{x}_{tot}^K) \prod_{k=1}^K p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^{k-1} | \boldsymbol{x}_{tot}^k)$ *with learned Gaussian distribution starting as* $p(\boldsymbol{x}_{tot}^K) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \in \mathcal{R}^{N \times d}$*, where* x_{tot} *is the generated data,* N *is the number of agent,* d *is data dimension,* K *is the diffusion steps. After* $p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^0)$ *is learned to model ground truth distribution, if there exists a joint individual distribution functions* $[p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^0) := \int p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^{0:K}) d\boldsymbol{x}_i^{1:K}]_{i=1}^N$*, where* $\boldsymbol{x}_i^k \in \mathcal{R}^d$ *is the data generated by agent* i*,* $\boldsymbol{x}_i^K \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$*, such that the following conditions are satisfied.*

the joint distribution from individual diffusion models equals that of the centralized model.

$$\prod_{i=1}^N p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^0) = p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^0) \quad \boldsymbol{\theta}_i \subset \boldsymbol{\theta}_{tot} \tag{3}$$

It indicates that the collection of generated samples $\boldsymbol{x}_i^0$*,* identically distributed *as* $\boldsymbol{x}_{tot}^0$*. We can state that* $[p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^0)]_{i=1}^N$ *satisfy IGD for* $p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^0)$ *and the diffusion model* $p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^0)$ *is generatively factorized by diffusion models* $[p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i)]_{i=1}^N$*.*

The IGD principle is a generalization of the IGM principle

Replace with action u

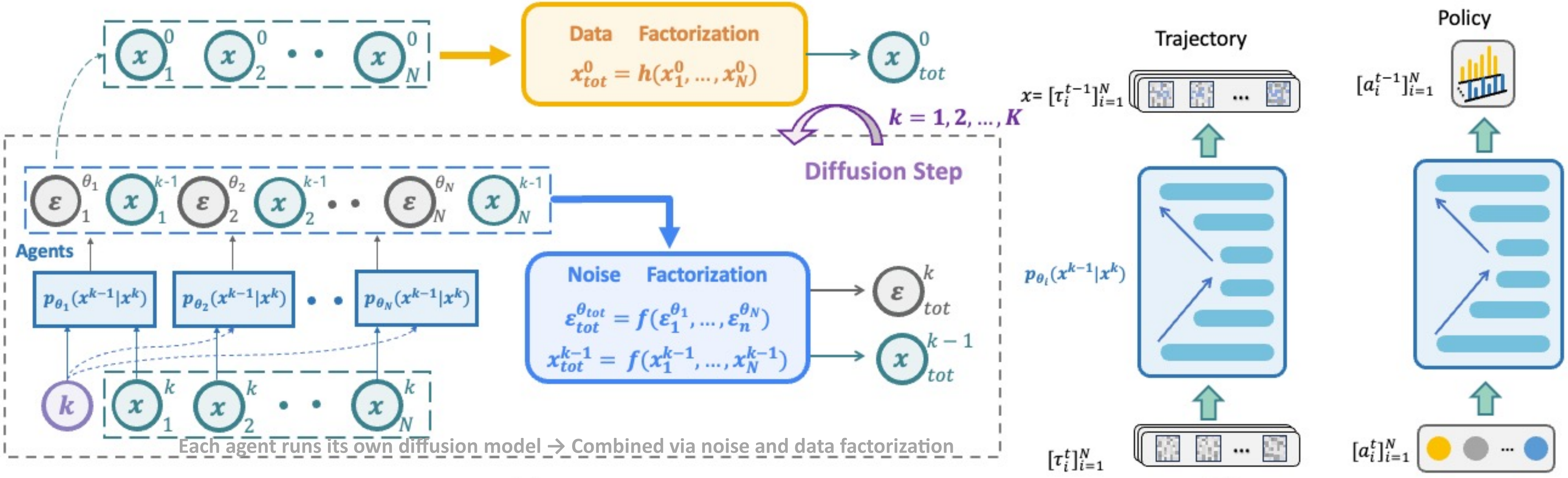
$$\prod_{i=1}^{N} p_{\theta_i}(x_i^0) = p_{\theta_{\text{tot}}}(x_{\text{tot}}^0)$$

$$\prod_{i=1}^{N} p\left(\arg\max_{u_i} Q_i(\tau_i, u_i)\right) = p\left(\arg\max_{u_{\text{tot}}} Q_{\text{tot}}(\tau_{\text{tot}}, u_{\text{tot}})\right)$$

$$\arg\max_{u_i} Q_i(\tau_i, u_i) \qquad \arg\max_{u_{\text{tot}}} Q_{\text{tot}}(\tau_{\text{tot}}, u_{\text{tot}})$$

DoF Framework

How Does DoF Work? A Two-Stage Diffusion Factorization Framework



(a) Factorized Diffusion Process

(b) Trajectory-Level Generation

(c) Policy-Level Generation

Decentralized models + factorization = Centralized output

Concat: A Simple Yet Effective Noise Factorization

Theorem 1. *A multi-agent diffusion model* $p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^0)$

$$p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^0) := \int p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^{0:K})\, d\boldsymbol{x}_{tot}^{1:K} \tag{8}$$

$$\boldsymbol{\epsilon}_{tot}^k = \oplus[\boldsymbol{\epsilon}_i^k]_{i=1}^N \quad \boldsymbol{\epsilon} \in \mathcal{N}(\mu, \sigma) \quad 0 \leq k \leq K \tag{9}$$

$$\boldsymbol{x}_{tot}^k = \oplus[\boldsymbol{x}_i^k]_{i=1}^N \quad 0 \leq k \leq K \tag{10}$$

$$\boldsymbol{\epsilon}_{tot}^{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^k, k) = \oplus[\boldsymbol{\epsilon}_i^{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^k, k)]_{i=1}^N \tag{11}$$

is generatively factorized by $[p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i)]_{i=1}^N$. $\oplus$ *is the Concat function.* $p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^0) := \int p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^{0:K})\, d\boldsymbol{x}_i^{1:K}$. $\boldsymbol{\epsilon}_i^t$ *is the noise added during the forward process.* $\boldsymbol{\epsilon}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^k, k)$ *is used for the denoising process to predict the source noise* $\boldsymbol{\epsilon}_i^0 \sim \mathcal{N}(0, I)$ *that determines* $\boldsymbol{x}_i^k$ *from* $\boldsymbol{x}_i^0$.

Concat directly stacks individual agent noises and data to construct the centralized diffusion process.

✓ Concat satisfies IGD by aligning local and global distributions via direct concatenation.

✓ Easy to implement, but assumes aligned feature spaces across agents.

WConcat: Weighted Noise Factorization for Multi-Agent Diffusion

Theorem 2. *A multi-agent diffusion model* $p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^0)$

$$p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^0) := \int p_{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^{0:K})\, d\boldsymbol{x}_{tot}^{1:K} \tag{B.25}$$

$$\boldsymbol{\epsilon}_{tot}^k = \uplus[\boldsymbol{\epsilon}_i^k]_{i=1}^N \quad \boldsymbol{\epsilon} \in \mathcal{N}(\mu, \sigma) \quad 0 \leq k \leq K \tag{B.26}$$

$$\boldsymbol{x}_{tot}^k = \uplus[\boldsymbol{x}_i^k]_{i=1}^N \quad 0 \leq k \leq K \tag{B.27}$$

$$\boldsymbol{\epsilon}_{tot}^{\boldsymbol{\theta}_{tot}}(\boldsymbol{x}_{tot}^k, k) = \uplus[\boldsymbol{\epsilon}_i^{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^k, k)]_{i=1}^N \tag{B.28}$$

$$\boldsymbol{\theta}_{tot} = \oplus[\boldsymbol{\theta}_i]_{i=1}^N \tag{B.29}$$

is generatively factorized by $[p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i)]_{i=1}^N$. $\uplus$ *is the WConcat function, and* $\oplus$ *is the Concat function.* $p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^0) := \int p_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^{0:K})\, d\boldsymbol{x}_i^{1:K}$. $\boldsymbol{\epsilon}_i^t$ *is the noise added during the forward process.* $\boldsymbol{\epsilon}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i^k, k)$ *is used for the denoising process to predict the source noise* $\boldsymbol{\epsilon}_i^0 \sim \mathcal{N}(0, I)$ *that determines* $\boldsymbol{x}_i^k$ *from* $\boldsymbol{x}_i^0$.

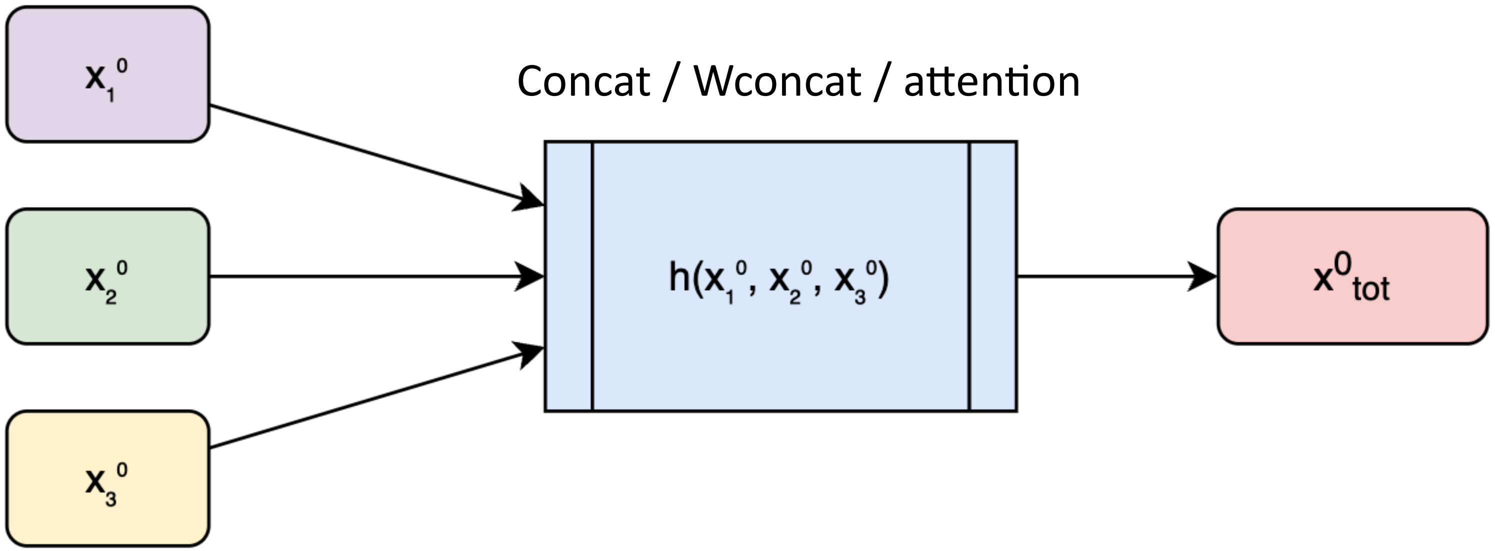
Adds learnable weights to agent-wise features before fusion.

✔ WConcat improves over Concat by enabling learnable importance weighting.

✔ Better handles heterogeneous or imbalanced agents.

Data Factorization

Combine individual agent outputs to reconstruct a consistent global trajectory.



Data Factorization aggregates per-agent predictions to reconstruct the global trajectory under the IGD principle.

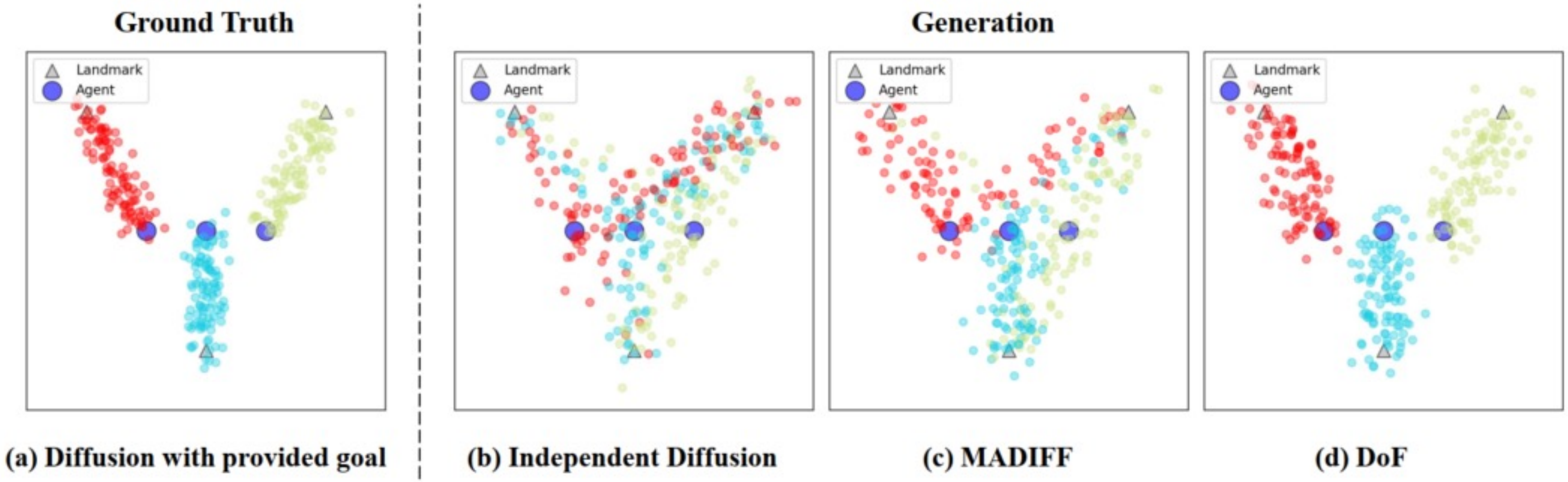
✓ Ensures IGD holds at the data level

✓ Compatible with various aggregation forms: concat, wconcat, attention

✓ Enables joint behavior from decentralized generation

Motivation Example: Why We Need DoF

(a) A Landmark covering game



Human-like Cooperation

No Coordination

Partial Centralization, Still Weak

Structured Coordination via Factorization

! Independent Diffusion： Each agent acts alone, causing collisions and failed coverage

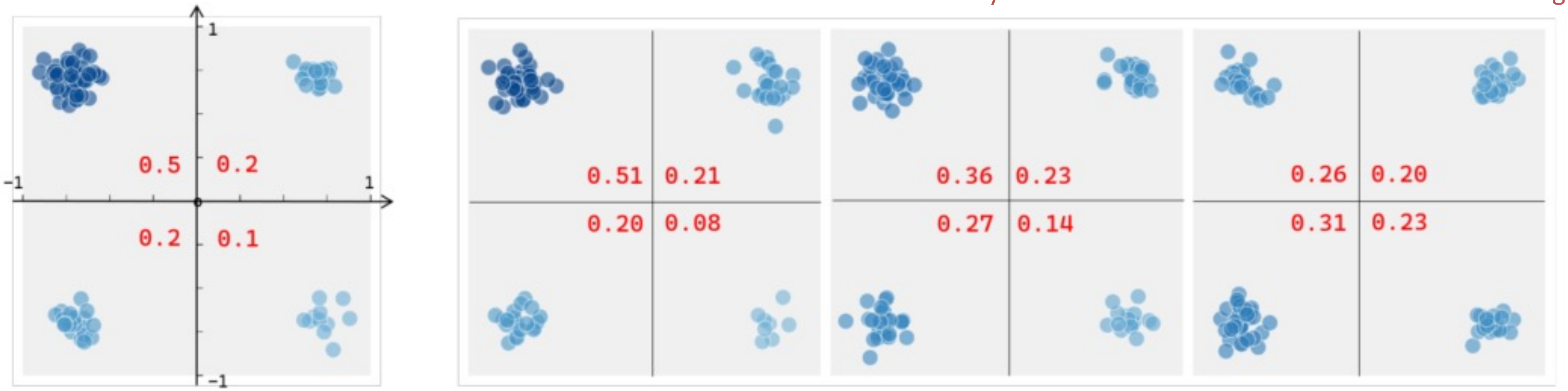
! MADIFF improves coordination slightly, but lacks structural factorization.

✓ DoF recovers optimal cooperation by combining local models under the IGD principle

Motivation Example: Why We Need DoF

(b) A matrix game generating two dimensional data

Only DoF maintains distributional structure across agents.



(a) Ground Truth (b) DoF (c) MADIFF (d) Independent Diffusion

(c) Q value generating Game

u_1 \ u_2	A	B
A	1.0	0.0
B	18.0	1.0

(a) Game Payoff Matrix 1

Q1 \ Q2	A	B
A	0.9	0.0
B	17.9	1.2

(b) DoF

u_1 \ u_2	A	B
A	4.0	0.0
B	14.0	2.0

(c) Game Payoff Matrix 2

Q1 \ Q2	A	B
A	4.0	0.0
B	13.9	2.1

(d) DoF

DoF accurately reconstructs the joint value function across both agents

Experiment —— SMAC

SMAC

DoF consistently ranks **1st or 2nd** across all maps and data qualities.

Maps	Data	QMIX	MABCQ	MACQL	MAICQ	MADT	MADIFF	DoF
3m	Good	3.6±0.8	3.7±1.1	19.1±0.1	18.7±0.7	19.0±0.3	19.3±0.5	**19.8±0.2**
	Medium	4.4±1.4	4.0±1.0	13.7±0.3	13.9±0.8	15.8±0.5	16.4±2.6	**18.6±1.2**
	Poor	5.0±1.1	3.4±1.0	4.2±0.1	8.4±2.6	4.2±0.1	10.3±6.1	**10.9±1.1**
8m	Good	5.4±1.1	4.8±0.6	5.4±0.9	**19.6±0.2**	18.5±0.4	18.9±1.1	**19.6±0.3**
	Medium	6.1±1.2	5.6±0.6	4.5±1.5	17.9±0.5	18.2±0.1	16.8±1.6	**18.6±0.8**
	Poor	4.0±0.6	3.6±0.8	3.5±1.0	11.2±1.3	4.8±0.1	9.8±0.9	**12.0±1.2**
5m_vs_6m	Good	3.4±0.2	2.4±0.4	7.4±0.6	11.0±0.6	16.8±0.1	16.5±2.8	**17.7±1.1**
	Medium	3.6±0.4	3.8±0.5	8.1±0.2	10.6±0.6	16.1±0.2	15.2±2.6	**16.2±0.9**
	Poor	3.5±0.7	3.3±0.5	6.8±0.1	6.6±0.2	7.6±0.3	8.9±1.3	**10.8±0.3**
2s3z	Good	7.3±0.3	7.7±0.9	17.4±0.3	18.3±0.2	18.1±0.1	15.9±1.2	**18.5±0.8**
	Medium	7.5±0.7	7.6±0.7	15.6±0.4	17.0±0.1	15.1±0.2	15.6±0.3	**18.1±0.9**
	Poor	7.1±1.0	6.6±0.2	8.4±0.8	9.9±0.6	8.9±0.3	8.5±1.3	**10.0±1.1**
3s5z_vs_3s6z	Good	5.8±0.4	5.9±0.3	7.8±0.5	**13.5±0.6**	12.8±0.2	7.1±1.5	12.8±0.8
	Medium	5.5±0.3	6.5±0.5	8.5±0.6	11.5±0.2	11.6±0.3	5.7±0.6	**11.9±0.7**
	Poor	6.3±0.5	6.1±0.6	5.9±0.4	**7.9±0.2**	5.6±0.3	4.7±0.6	7.5±0.2
2c_vs_64zg	Good	6.3±0.2	10.1±0.2	12.9±0.2	14.2±0.3	13.8±0.3	14.7±2.2	**16.1±0.8**
	Medium	5.9±0.1	9.9±0.2	11.6±0.1	12.0±0.1	11.8±0.2	12.8±1.2	**13.9±0.9**
	Poor	5.2±0.3	9.0±0.2	10.2±0.1	9.8±0.3	10.1±0.5	10.8±1.1	**11.5±1.1**

Experiment —— SMACv2 and MPE

SMACv2

Map	Data	BC	QMIX	MABCQ	MACQL	MAICQ	MADIFF	DoF
terran_5_vs_5	replay	7.3±1.0	10.3±1.2	13.8±4.4	11.8±0.9	13.7±1.7	13.3±1.8	**15.4±1.3**
Zerg_5_vs_5	replay	6.8±0.6	10.1±2.4	10.3±1.2	10.3±3.4	10.6±0.7	10.2±1.1	**12.0±1.1**
terran_10_vs_10	replay	7.4±0.5	9.9±2.4	12.7±2.0	11.8±2.0	14.4±0.7	13.8±1.3	**14.6±1.1**

MPE

Dataset	Task	MAICQ	MA-TD3+BC	MACQL	OMAR	MADIFF	DoF
Expert	Spread	101.4±3.4	110.3±3.3	85.3±4.6	113.9±2.6	120.1±6.3	**126.4±3.9**
	Tag	95.2±10.1	113.1±11.6	84.3±10.2	115.8±13.6	120.8±11.3	**125.6±8.6**
	World	98.5±21.8	95.3±18.3	65.4±20.2	113.4±23.1	124.7±20.1	**135.2±19.1**
Medium	Spread	29.3±5.5	32.3±3.8	35.3±10.3	45.0±18.8	67.5±8.5	**75.6±8.7**
	Tag	58.3±18.0	63.3±25.6	62.3±27.8	55.3±16.7	78.6±12.3	**86.3±10.6**
	World	69.9±20.1	72.4±9.3	56.4±6.4	69.2±21.5	80.1±13.4	**85.2±11.2**
Md-Replay	Spread	13.7±5.6	14.4±5.8	19.2±6.4	35.3±14.0	48.1±3.6	**57.4±6.8**
	Tag	29.5±21.8	25.7±20.1	23.9±16.2	52.4±18.3	57.4±13.4	**65.4±12.5**
	World	12.0±9.1	15.4±8.1	21.3±10.3	42.6±28.2	51.6±12.1	**58.6±10.4**
Random	Spread	5.3±3.4	8.8±4.4	20.5±5.8	30.4±8.2	20.6±7.6	**35.9±6.8**
	Tag	2.2±2.6	3.7±3.5	2.7±4.4	10.9±3.8	13.3±3.4	**16.5±6.3**
	World	1.0±2.2	2.8±3.5	2.4±3.2	9.2±3.6	6.1±2.2	**13.1±2.1**

✓ DoF consistently outperforms prior methods in both adversarial and cooperative settings.
✓ Scales to **high-dimensional, hard exploration** tasks (e.g., MPE-World).
✓ Learns robust strategies **even from non-expert data**

Experiments —— ablations

We study how different noise factorization functions f affect the performance of DoF on SMAC maps.

DoF with different Noise Factorization Function f

Maps	Dataset	Decentralized			Centralized		MADIFF	DoF+MADIFF
		Concat	WConcat	Dec-Atten	QMix	Atten		
3m	Good	19.7±0.6	19.8±0.5	4.3±2.3	3.8±1.3	19.8±0.4	19.3±0.5	19.7±0.4
	Medium	17.8±2.1	18.0±1.0	4.5±1.8	4.2±1.5	18.0±1.4	16.4±2.6	18.2±1.1
	Poor	10.6±1.6	11.4±0.7	3.2±1.5	3.5±1.4	11.3±1.3	10.3±1.5	10.8±1.2
5m_vs_6m	Good	16.7±1.4	17.0±0.8	3.6±1.5	4.1±1.2	17.1±0.8	16.5±2.8	16.7±1.2
	Medium	15.6±1.1	15.9±1.2	2.5±1.6	2.9±1.4	15.9±0.6	15.2±2.6	15.7±0.9
	Poor	9.8±1.1	10.7±0.8	2.9±1.4	2.3±1.1	10.2±0.7	8.9±1.3	10.0±0.8

✓ WConcat is the most effective noise factorization function in DoF.

Experiments —— ablations

DoF with different Data Factorization Function h

Maps	Dataset	h = Concat	h = WConcat	h = Atten
3m	Good	19.7±0.6	19.8±0.2	19.9±0.1
	Medium	17.8±2.1	18.6±1.2	18.7±1.0
	Poor	10.6±1.6	10.9±1.1	10.8±0.9
5m_vs_6m	Good	15.8±1.4	17.7±1.1	18.2±0.9
	Medium	14.9±1.1	16.2±0.9	16.8±0.8
	Poor	9.8±1.1	10.8±0.3	11.0±0.5
3s5z_vs_3s6z	Good	11.3±0.9	12.8±0.8	15.2±0.7
	Medium	9.4±0.7	11.9±0.7	12.8±0.5
	Poor	6.8±0.3	7.5±0.2	8.2±0.3
2s3z	Good	15.5±1.0	18.5±0.8	19.5±0.3
	Medium	14.8±0.8	18.1±0.9	18.5±0.3
	Poor	9.6±1.1	10.0±1.1	10.2±0.7

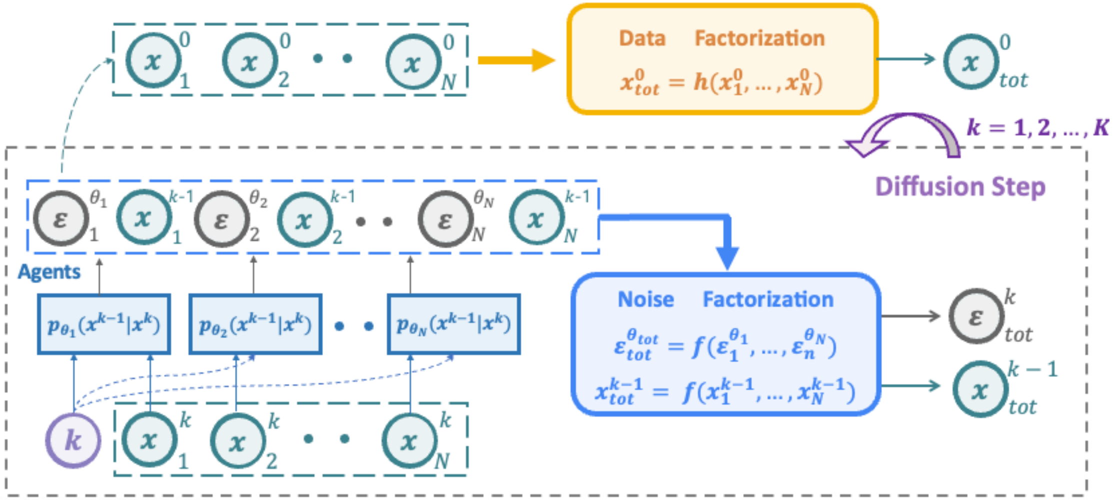
✓ Learnable attention is the best data aggregation method for global trajectory recovery.

Summary

- IGD principle, a generalization of IGM tailored for diffusion-based multi-agent modeling.
- DoF, a diffusion factorization framework satisfying IGD for offline Multi-Agent Reinforcement Learning.
- Through extensive experiments, we show that DoF achieves state-of-the-art performance across SMAC, SMACv2, and MPE.

For more details, please check our project page:
https://github.com/xmu-rl-3dv/DoF

Contact us:
chaoli@stu.xmu.edu.cn
dengziwei@stu.xmu.edu.cn
siqishen@xmu.edu.cn



Thanks for your attention!