From Embedding to Control: Representations for Stochastic Multi-Object Systems

Xiaoyuan Cheng[1], Yiming Yang[1], Jiang Wei[2], Chenyang Yuan[3], Zhuo Sun[4], Yukun Hu[1,*]

[1] Dynamic Systems Lab, University College London, United Kingdom; [2] Independent Researcher, Hong Kong, China

[3] Electrical and Electronic Engineering, University of Sheffield, United Kingdom; [4] Statistics and Data Science, Shanghai University of Finance and Economics, China

Motivation

- **Lack Theoretical Foundation:** no formal guarantee for embedding and controlling stochastic multi-object dynamics;
- **Unrealistic relations:** assume the uniform interaction among all neighbors;
- **Limited scalability**: lack generalization to large-scale or random graphs.

Main Idea – Graph Controllable Embedding

Hilbert Space Embedding of Distributions

Features in Hilbert spaces:

Observation: $o_t \mapsto \psi_t^o$

history: $h_t \mapsto \psi_t^h$

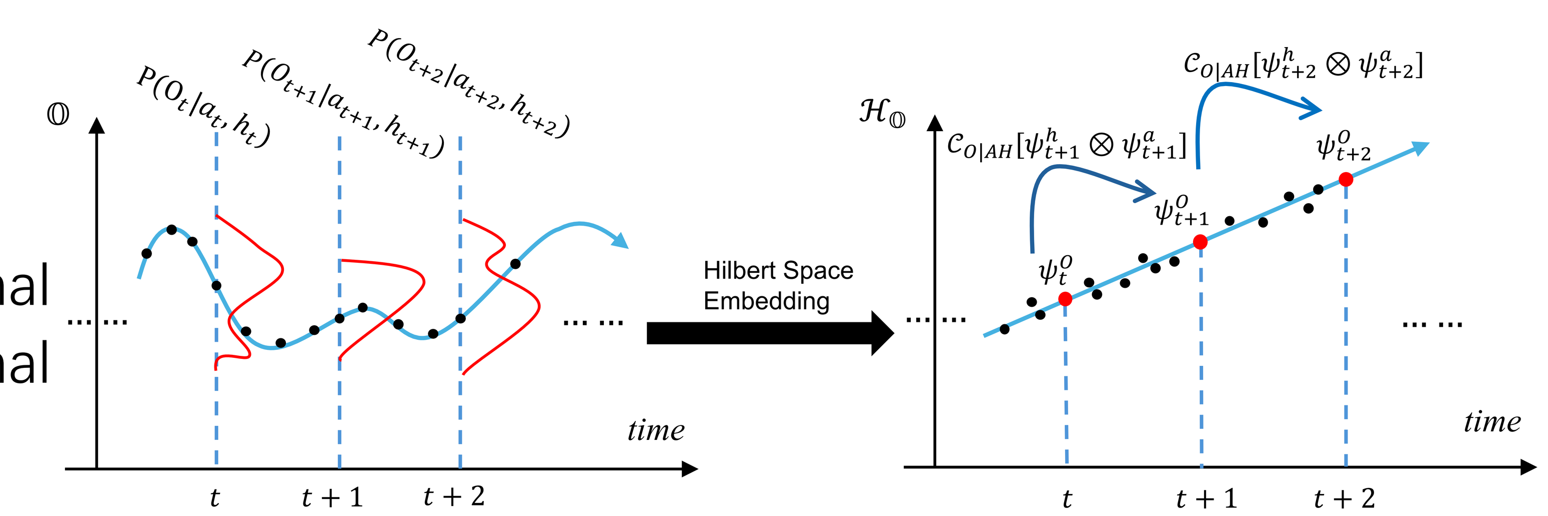
Action: $a_t \mapsto \psi_t^a$

and

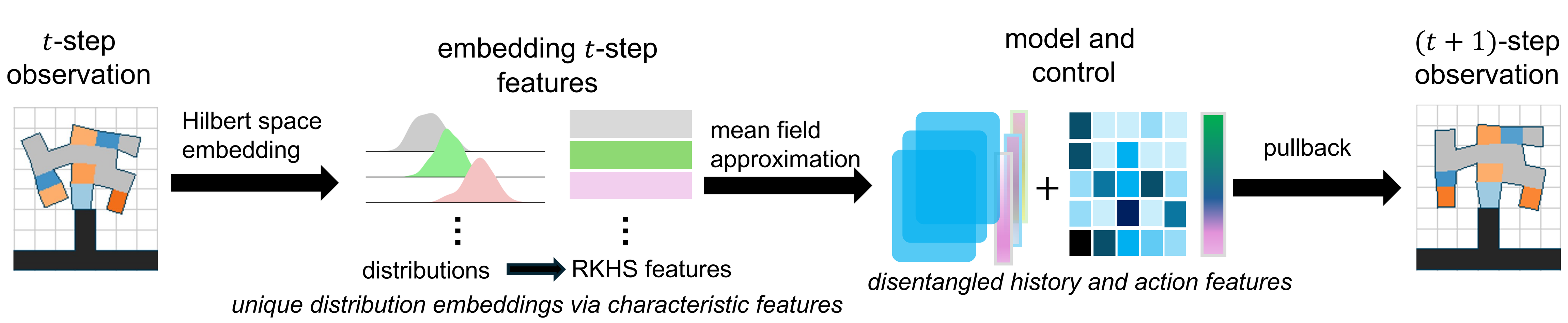
$$\mathbb{E}[\psi_t^o | a_t, h_t] = \mathcal{C}_{O|AH}[\psi_t^h \otimes \psi_t^a],$$

Here, $\mathcal{C}_{O|AH}$ is the conditional embedding operator for conditional distributions $p(O_t|a_t, h_t)$.

Figure. Left: stochastic nonlinear dynamics evolves following conditional distribution $p(O_t|a_t,h_t)$ (red curves: probability distributions, black dots: realizations). *Right*: After embedding into RKHS, dynamics become linear under operator $\mathcal{C}_{O|AH}$ (red dots: expectations).



Method



- **Loss Function**:

$$\mathbb{E}_\mu \left\| \sum_{t=1}^{M} \left[\begin{bmatrix} \psi_t^{o,1} \\ \vdots \\ \psi_t^{o,N} \end{bmatrix} - \begin{bmatrix} \hat{\mathcal{C}}_{O^1|H} \\ \vdots \\ \hat{\mathcal{C}}_{O^N|H} \end{bmatrix} \odot \begin{bmatrix} \sum_j \alpha_t^{1,j} \psi_t^{h,j} \\ \vdots \\ \sum_j \alpha_t^{N,j} \psi_t^{h,j} \end{bmatrix} - \begin{bmatrix} \hat{\mathcal{C}}_{O^1|A^1} & \cdots & \hat{\mathcal{C}}_{O^1|A^N} \\ \vdots & \ddots & \vdots \\ \hat{\mathcal{C}}_{O^N|A^1} & \cdots & \hat{\mathcal{C}}_{O^N|A^N} \end{bmatrix} \begin{bmatrix} \psi_t^{a,1} \\ \vdots \\ \psi_t^{a,N} \end{bmatrix} \right] \right\|_{HS}$$

where $\mathcal{C}_{O^i|H}$ and $\mathcal{C}_{O^i|A^j}$ are the decomposed tractable operators for $\mathcal{C}_{O^i|A^jH^j}$, quantifying the influence from j-object to i-object. $\alpha_t^{i,j}$ is the Boltzmann-Gibbs weight estimating from kernel functions.

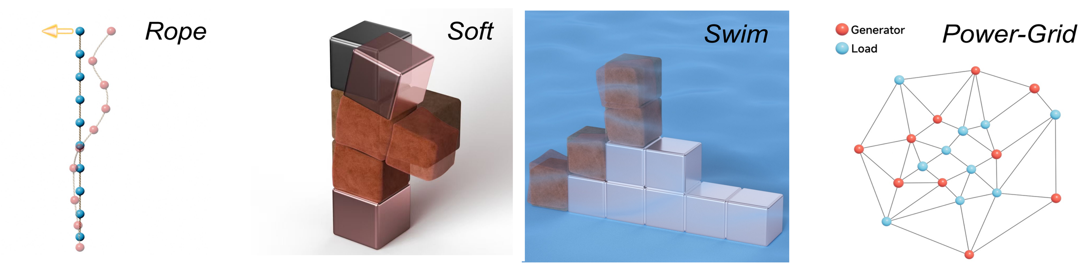
- **Simple control**: $\min_{\{\psi_t^a\}_{t=1}^M} \mathbb{E}[\sum_{t=1}^{M} \|[\hat{\psi}_t^{O,1}, \ldots, \hat{\psi}_t^{O,N}]^\top - [\psi_*^{o,1}, \cdots, \psi_*^{o,N}]^\top\|_{Q_1}^2 + [\psi_t^{a,1}, \ldots, \psi_t^{a,N}]^\top\|_{Q_2}^2]$

solving with simple and efficient large-scale linear quadratic regulator.

† Equal contribution; Corresponding Author: Yukun Hu (yukun.hu@ucl.ac.uk)

Experiments

Experiments on physical systems, robotics, and power grids with random graphs



Q1. Necessary to design specific controllable embeddings? *Yes—VAE and PCC capture trajectories but lack control-ready structure. GraphODE adds relations but has no explicit control design and relies on local linearization, so performance suffers.*

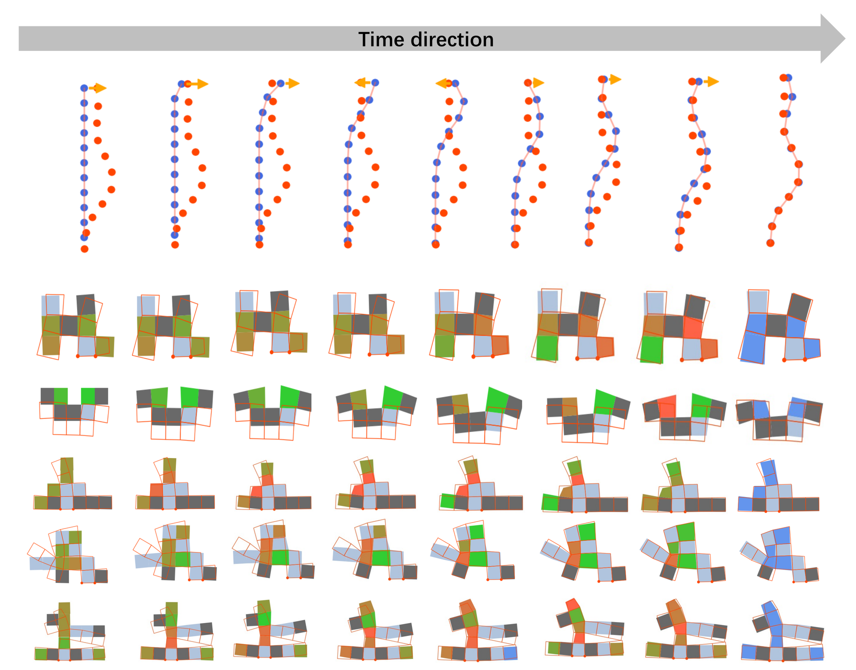


Table: Performance comparison for in-distribution and few-shot validation. Control cost and error are shown as mean ± standard deviation, with the best and second-best results highlighted in green and blue, respectively.

		In-Distribution Validation		Few-Shot Validation	
Methods	Environments	Control cost	Control error	Control cost	Control error
VAE		573.1 ± 108.7	0.73 ± 0.19	835.4 ± 113.2	0.92 ± 0.15
PCC		513.3 ± 92.5	0.68 ± 0.15	732.8 ± 94.5	0.80 ± 0.12
GraphODE		417.8 ± 87.9	0.52 ± 0.17	693.5 ± 58.2	0.58 ± 0.09
KPM		385.5 ± 75.2	0.44 ± 0.06	523.4 ± 22.8	0.61 ± 0.11
CKO	Swim	389.1 ± 76.9	0.42 ± 0.13	421.0 ± 70.0	0.44 ± 0.08
Ours (vMF)		392.7 ± 73.1	0.45 ± 0.09	452.3 ± 62.9	0.43 ± 0.15
Ours (Laplace)		403.1 ± 68.3	0.46 ± 0.13	435.7 ± 74.4	0.45 ± 0.10
Ours (Gaussian)		**383.7 ± 77.8**	**0.41 ± 0.08**	**404.3 ± 74.2**	**0.41 ± 0.09**

Q2. Can the non-uniform weighting enhance control performance? *Yes — Both prediction and control performance can be significantly improved.*

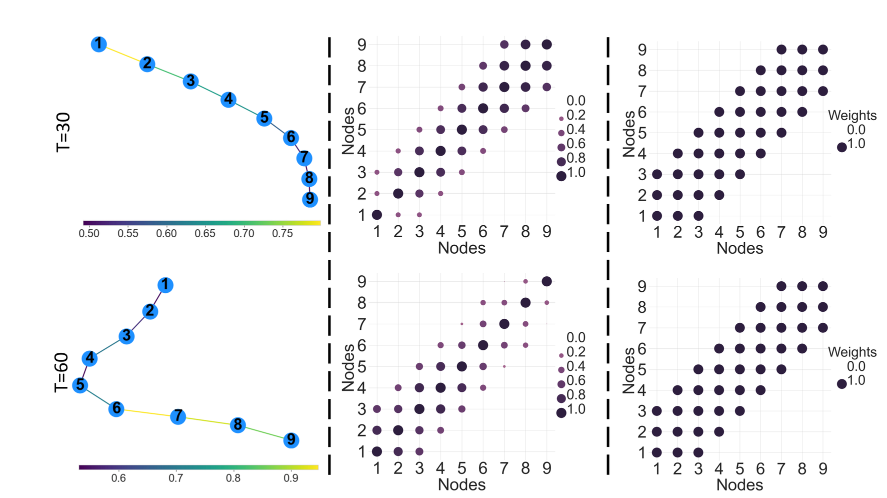
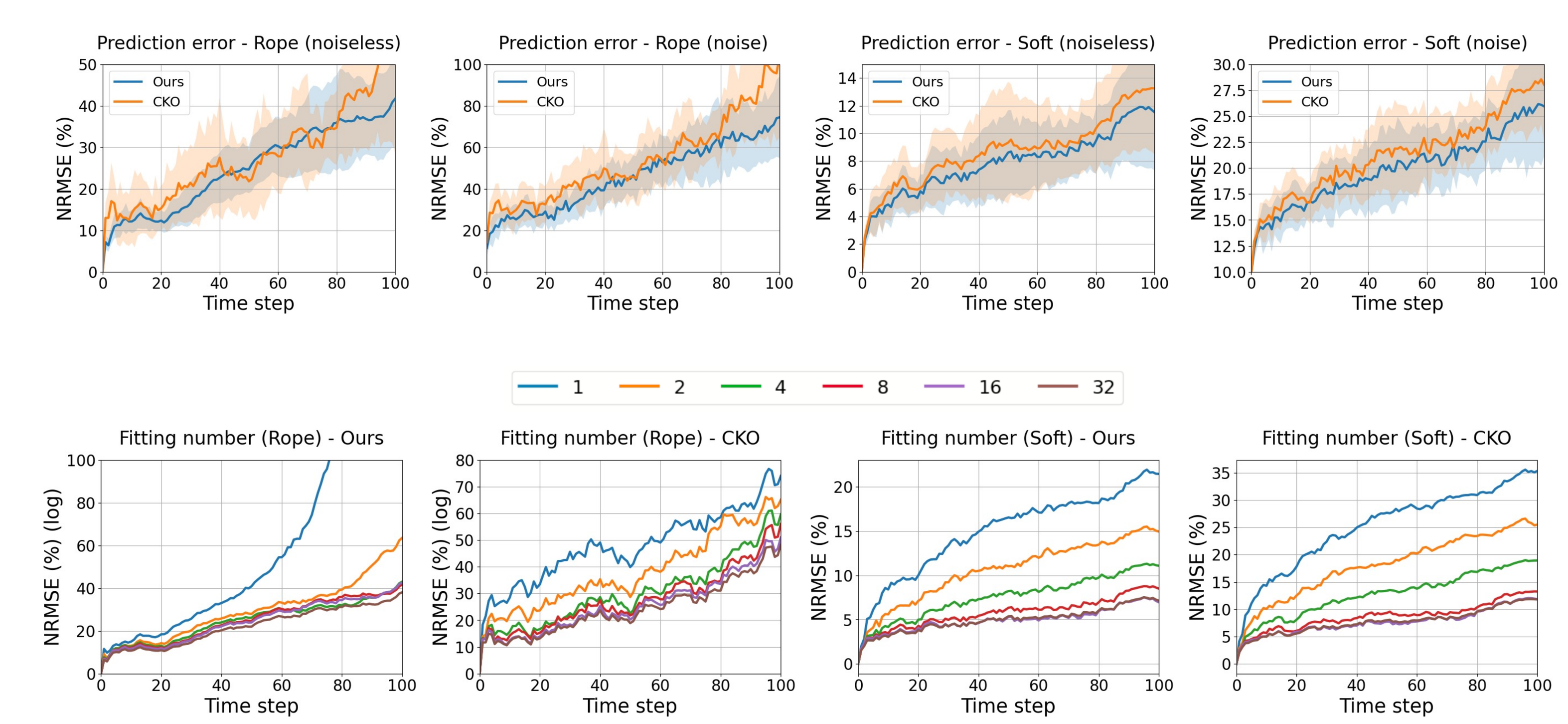
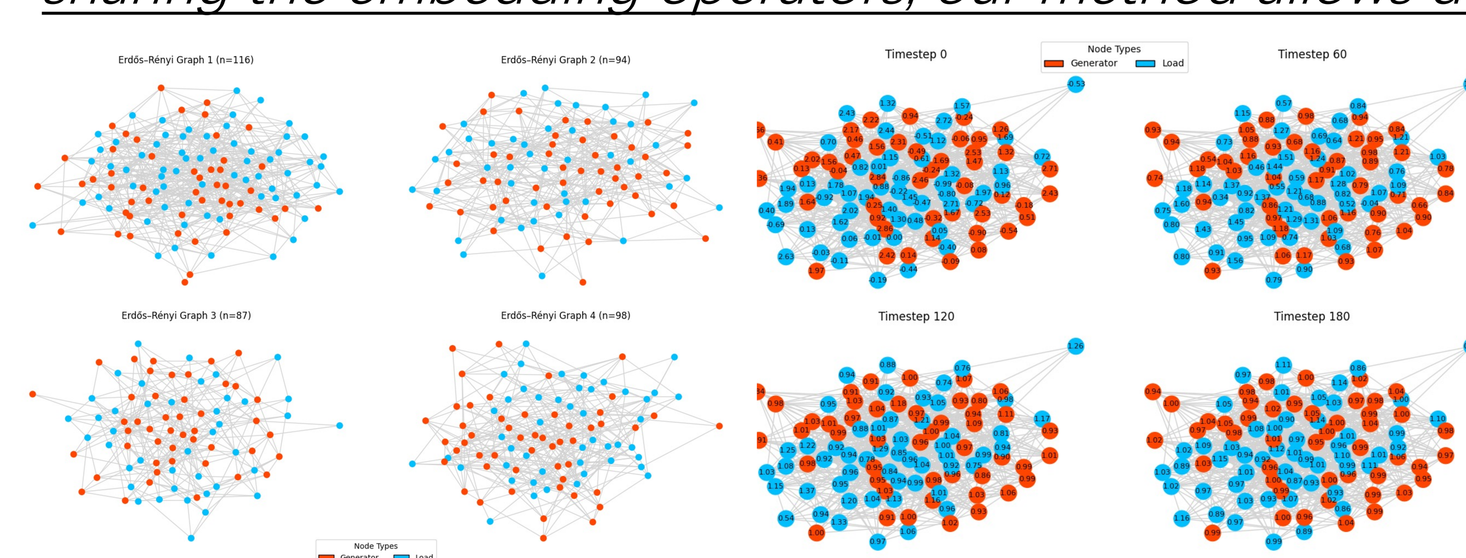


Figure: 9-object rope visualization. Left: observations at t=30,60 with colors showing pairwise L2 distances. Middle: our adaptive (unnormalized) Boltzmann–Gibbs weights, showing uneven neighbor influence. Right: uniform weights, where all neighbors affect equally.



Q3. Can our method be extended to the large-scale even random graphs? *Yes — by sharing the embedding operators, our method allows diverse graphs with strong performance.*



	Method	Noiseless	2%	5%	10%
Control Error	GraphODE	0.58 ± 0.043	0.62 ± 0.073	NaN	NaN
	KPM	0.42 ± 0.028	0.50 ± 0.022	NaN	NaN
	CKO	0.47 ± 0.031	0.48 ± 0.034	0.51 ± 0.027	0.65 ± 0.051
	Ours (Gaussian)	**0.21 ± 0.005**	**0.27 ± 0.018**	**0.39 ± 0.024**	**0.63 ± 0.037**
Control Cost	GraphODE	13.29 ± 0.95	14.93 ± 1.06	NaN	NaN
	KPM	9.19 ± 1.66	11.70 ± 1.68	NaN	NaN
	CKO	11.04 ± 0.82	11.22 ± 0.84	11.69 ± 1.67	12.42 ± 2.27
	Ours (Gaussian)	**4.77 ± 0.61**	**4.99 ± 0.69**	**5.72 ± 0.91**	**7.81 ± 1.13**

Future Direction

- **Our work focuses on pair-wise relations:** extending it to richer relational structures, such as hypergraphs.