CL-DiffPhyCon: Closed-loop Diffusion Control of Complex Physical Systems

[1]Long Wei* , [1]Haodong Feng*, [2]Yuchen Yang#, [1]Ruiqi Feng, [3]Peiyan Hu#, [4]Xiang Zheng#, [1]Tao Zhang, [1]Dixia Fan, [1]Tailin Wu†

[1]Westlake University, [2]Nankai University, [3]Academy of Mathematics and Systems Science, CAS, [4]South China University of Technology

(* equal contributions; # intern at Westlake University, † corresponding author)

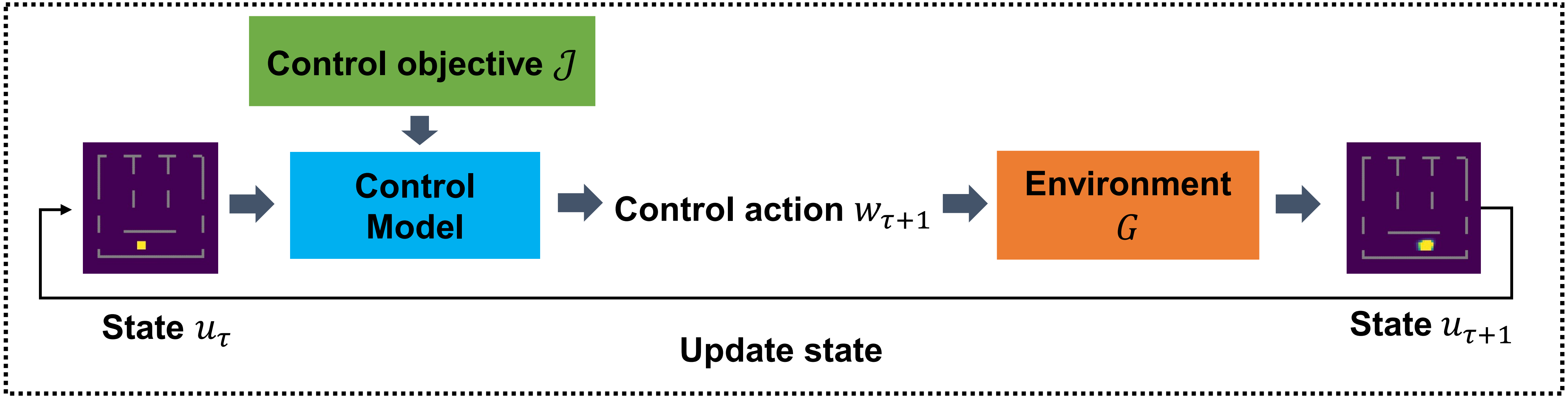
Contact: {weilong,fenghaodong,wutailin}@westlake.edu.cn

Outline

- **Introduction**
  - Problem setup, Applications, Challenges, Previous diffusion control approaches
- **Contributions**
- **Approach**
  - Notations, Learning, Inference algorithm, Efficiency
- **Results**
  - 1D Burgers' equation control
  - 2D Navier-Stokes equation control (indirect smoke control)

Introduction

- **Problem setup.** ***Closed-loop* control task**: Given a control objective $\mathcal{J}$, an initial state u_0, and a system dynamics G, find the optimal policy π to optimize $\mathcal{J}$:
  - $\min_{\pi} \mathbb{E}_{w_{\tau+1}\sim\pi(w_{\tau+1}|u_\tau)}[\mathcal{J}(u_0, w_1, u_1, \cdots, w_N, u_N)]$,
    - s.t. $u_{\tau+1} = G(u_\tau, w_{t+1}, \xi_\tau) \leftarrow$ *Closed-loop* requirement



- **Applications.** Underwater Robot control, fusion control, rocket control, et al.
- **Challenges.** High-dimensional and highly nonlinear system dynamics.

Introduction

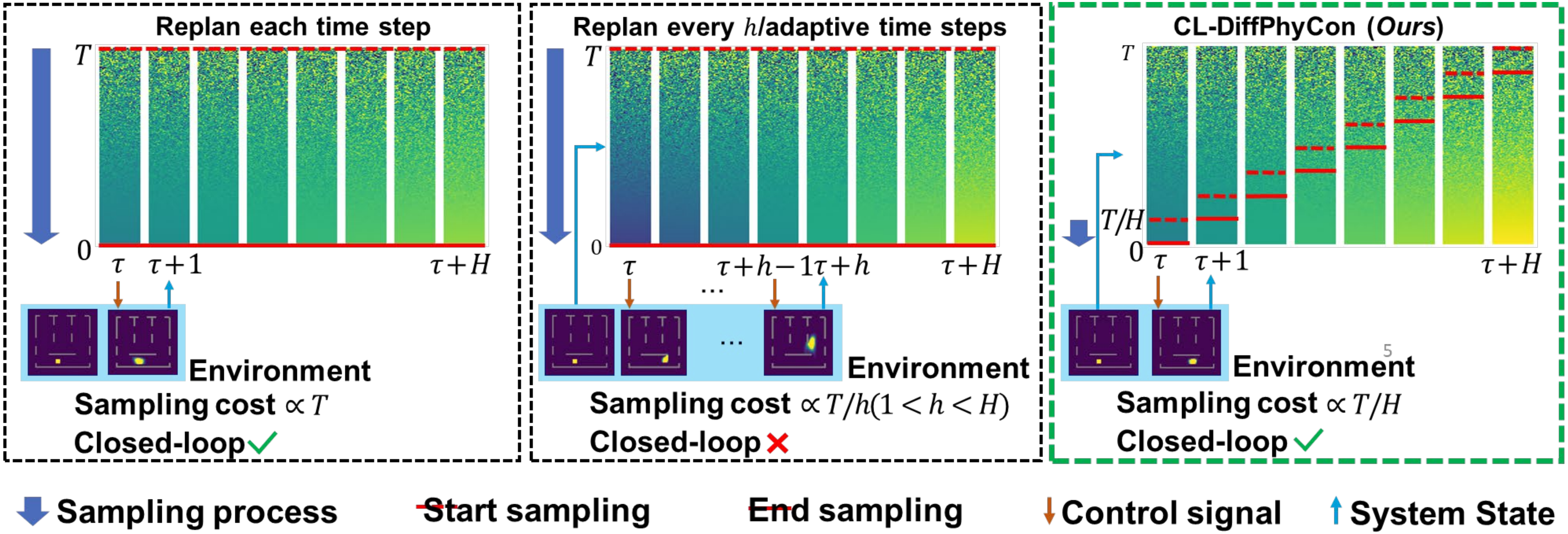
Previous diffusion control approaches

- Methods: Use diffusion models to sample control actions [1,2].
- Open-loop control: Plans all actions inside the model horizon in advance, and applies them sequentially to the environment, but with the following **limitations**:
    - Small replan interval: high computational cost; inconsistent trajectory.
    - Large replan interval: violates the closed-loop requirement.
    - Adaptive replan interval [3]: extra computation; extra hyperparameters; also violates the closed-loop requirement.

[1] Cheng Chi et al. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. RSS2023
[2] Long Wei et al. DiffPhyCon: A Generative Approach to Control Complex Physical Systems. NeurIPS 2024.
[3] Siyuan Zhou et al. Adaptive online replanning with diffusion models. NeurIPS 2023.

Contributions

We propose **CL-DiffPhyCon**, an efficient ***closed-loop*** diffusion control method, which accelerates the sampling process of diffusion model-based control significantly.



Approach

- **<u>Notations</u>.** joint variable $z_\tau(t) \triangleq [u_\tau(t), w_\tau(t)]$ with noise level t at physical time step τ
- synchronous noise variable: $z_{\tau:\tau+H-1}(t) \triangleq [z_\tau(t), z_{t+1}(t), \cdots, z_{\tau+H-1}(t)]$
- asynchronous noise variable: $\tilde{z}_{\tau:\tau+H-1}(t) \triangleq [z_\tau(t), z_{t+1}\left(t+\frac{T}{H}\right), \cdots, z_{\tau+H-1}(t+\frac{H-1}{H}T)]$

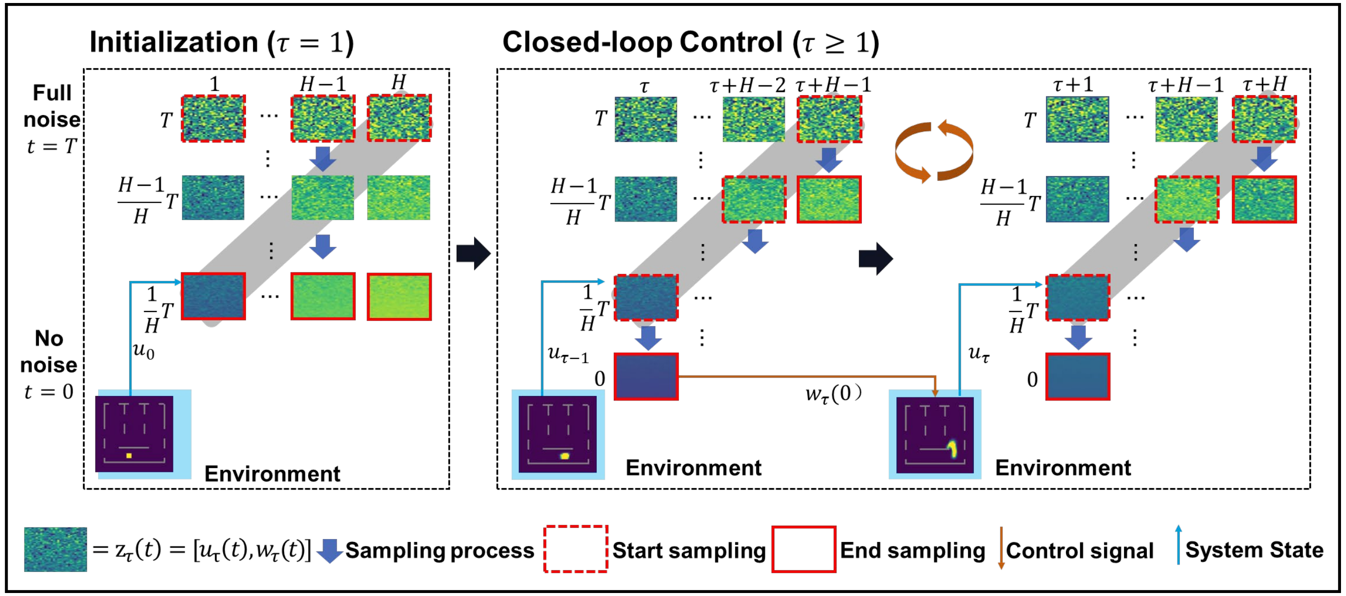
Theorem: (*informal*) To sample from the joint distribution $p(z_1, \cdots, z_N | u_0)$, we only need to learn from the following two distributions:

initializing distribution: $p(\tilde{z}_{1:H}(T/H)|u_0)$,

and *transition distribution*: $p(\tilde{z}_{\tau:\tau+H-1}(0)|a_{\tau-1}(0), \tilde{z}_{\tau:\tau+H-1}(T/H))$

- **<u>Learning</u>**. We propose to learn the *initializing distribution* by a synchronous diffusion model ϵ_ϕ, and the *transition distribution* by an asynchronous diffusion model ϵ_θ:
- $\mathcal{L}_{\text{synch}} = \mathbb{E}_{t\sim U(0,T),\epsilon\sim\mathcal{N}(0,\mathrm{I}),(u_0,z_{1:H})\sim\text{Data}}\left\|\epsilon - \epsilon_\phi(z_{1:H}(t), s_0, t)\right\|_2^2$
- $\mathcal{L}_{\text{asynch}} = \mathbb{E}_{\tau\sim U(1,N-H+1),t\sim U(0,T/H),\epsilon\sim\mathcal{N}(0,\mathrm{I}),(s_{\tau-1},z_{\tau:\tau+H-1})\sim\text{Data}}\|\epsilon - \epsilon_\theta(\tilde{z}_{\tau:\tau+H-1}(t), s_{\tau-1}, t)\|_2^2$

Approach



- **Inference (closed-loop control) algorithm.**

Initialization: Sample $z_{1:H}(T/H)$ by using ϵ_ϕ, then take its diagonal $\tilde{z}_{1:H}(T/H)$

For $\tau = 1, \cdots, N$ **do**

For $t = T/H, T/H - 1, \cdots, 1$ **do**

$$\tilde{z}_{\tau:\tau+H-1}(t-1) \leftarrow \tilde{z}_{\tau:\tau+H-1}(t) - \eta\left(\epsilon_\theta(\tilde{z}_{\tau:\tau+H-1}(t), s_{\tau-1}, t) + \lambda\nabla_z \mathcal{J}(\hat{\mathbf{z}}_k)\right) + \zeta, \zeta \sim \mathcal{N}(0, \mathrm{I})$$

End for

Input $w_\tau(0)$ contained in $z_\tau(0)$ to the environment; transfer the state to s_τ

Sample $z_{\tau+H}(H) \sim \mathcal{N}(0, \sigma_T^2 \mathrm{I})$ *//append the end of the horizon with noise*

Update $\tilde{z}_{\tau+1:\tau+H}(T/H) \leftarrow [z_{\tau+1}(T/H), \cdots, z_{\tau+H}(H)]$ *// move to the next horizon*

End for

Efficiency:.

- Our CL-DiffPhyCon is H/h times faster than DiffPhyCon-h.
- Could be enhanced further by using fast sampling methods, such as DDIM.

Results

- **1D Burgers' Equation Control.**

$$\begin{cases} \frac{\partial u}{\partial \tau} = -u \cdot \frac{\partial u}{\partial x} + \nu \frac{\partial^2 u}{\partial x^2} + w(\tau, x) & \text{in } [0, \mathcal{T}] \times \Omega, \\ u(\tau, x) = 0 & \text{in } [0, \mathcal{T}] \times \partial\Omega, \\ u(0, x) = u_0(x) & \text{in } \{\tau = 0\} \times \Omega, \end{cases}$$

Control objective:

$$\min \mathcal{J} := \int_{\mathcal{T}} \int_{\Omega} |u(\tau, x) - u_d(\tau, x)|^2 \mathrm{d}x \mathrm{d}\tau$$

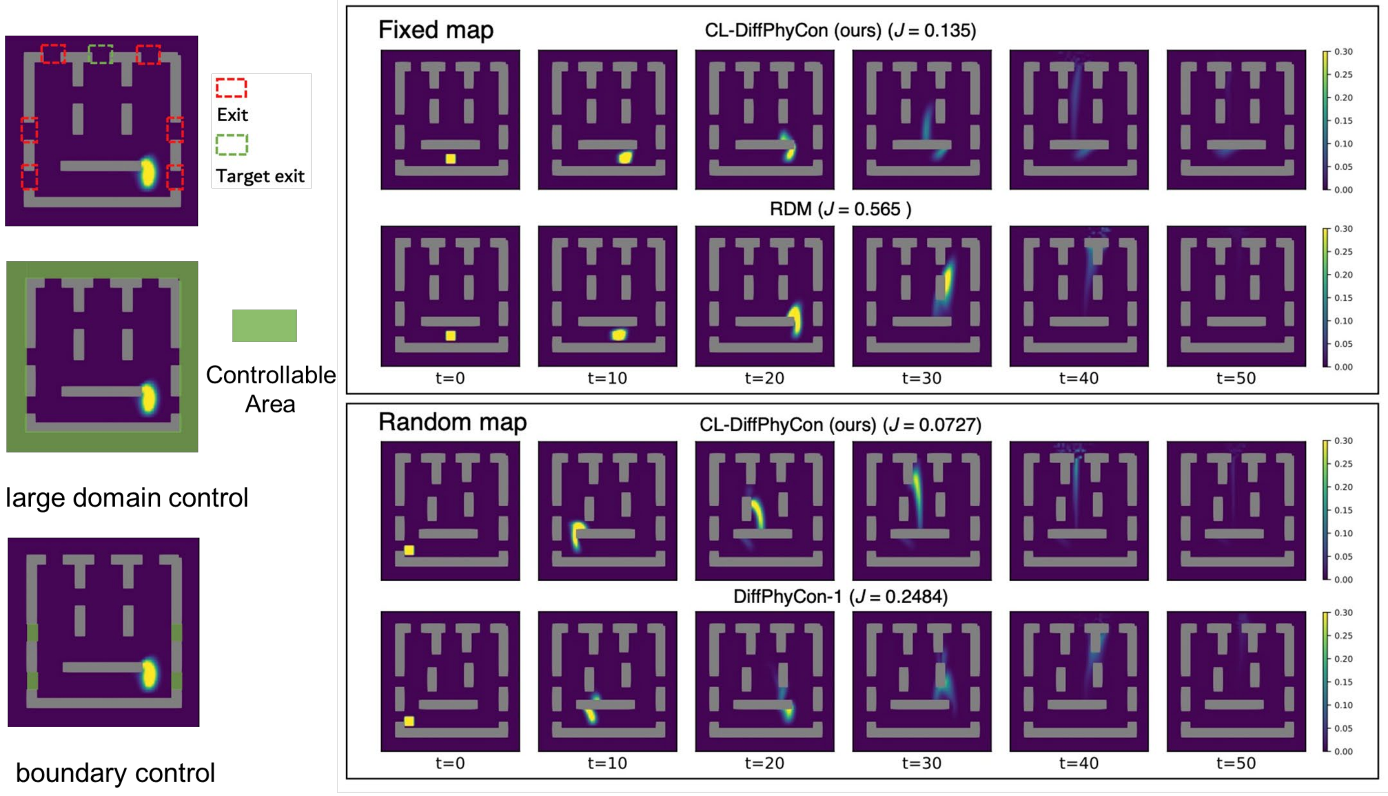
	noise-free ↓	physical constraint ↓	system noise ↓	measure noise ↓	FOPC ↓	POPC ↓	average time (s) ↓
BC	0.4708	0.4704	0.4138	0.3981	0.1093	0.0987	0.8356
BPPO	0.4686	0.4507	0.4088	0.3979	0.1079	0.0984	0.8231
PID	0.3250	0.3585	0.2323	0.2911	-	0.0827	**0.7717**
DiffPhyCon-1	0.0210	0.0214	0.0222	0.0232	0.0330	0.0332	49.2347
DiffPhyCon-5	0.0252	0.0257	0.0271	0.0272	0.0482	0.0484	10.8308
DiffPhyCon-15	0.0361	0.0365	0.0382	0.0377	0.1128	0.1132	4.9833
RDM [3]	0.0296	0.0296	0.0310	0.0336	0.1124	0.1121	6.9130
CL-DiffPhyCon (ours)	**0.0096**	**0.0110**	**0.0095**	**0.0127**	**0.0291**	**0.0295**	4.5474
CL-DiffPhyCon (DDIM, ours)	<u>0.0112</u>	<u>0.0123</u>	<u>0.0114</u>	<u>0.0146</u>	<u>0.0311</u>	<u>0.0313</u>	<u>0.8257</u>

(*DiffPhyCon-*h*: replan actions from scratch every *h* time steps by using DiffPhyCon).

Results

- **<u>2D Smoke Indirect Control (Navier-Stokes Equation)</u>**

Control objective: minimize the smoke failing to pass through the target exit.



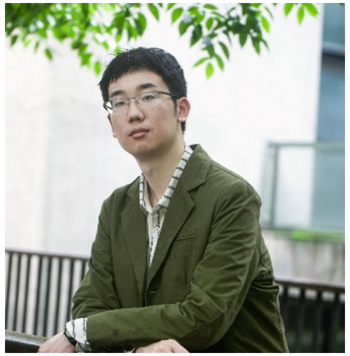
Results

- **<u>2D Smoke Indirect Control (Navier-Stokes Equation)</u>**

	Large domain control		Boundary control		Average
	Fixed map↓	Random map↓	Fixed map↓	Random map↓	time (s) ↓
BC	0.6722	0.7046	0.8861	0.8871	4.67
BPPO	0.6343	0.6524	0.8830	0.8844	4.65
DiffPhyCon-1	0.5454	<u>0.3754</u>	0.7517	0.7955	1666.50
DiffPhyCon-5	0.5051	0.5458	0.6703	0.7451	357.66
DiffPhyCon-14	0.5823	0.5621	<u>0.6498</u>	0.7221	141.88
RDM [3]	<u>0.4074</u>	0.4356	0.6553	<u>0.7087</u>	238.43
CL-DiffPhyCon (ours)	**0.3371**	**0.3485**	**0.6169**	**0.7003**	144.04
CL-DiffPhyCon (DDIM, ours)	0.4100	0.4254	0.6671	0.7109	26.01

Our group: AI for Scientific Simulation & Discovery Lab @ Westlake University


Long Wei


Haodong Feng


Yuchen Yang


Ruiqi Feng


Peiyan Hu


Xiang Zheng


Tao Zhang


Dixia Fan


Tailin Wu

Thank you!

Paper:

Group Website:

If you have any questions, please feel free to contact us at:

weilong@westlake.edu.cn
fenghaodong@westlake.edu.cn
wutailin@westlake.edu.cn