Application level presentation

Imitation learning methods

Phuong Nguyen

Figure 01: A robot arm doing pick and place task [1]

AGENDA

- Motivation
- Imitation learning data collecting methods
- Diffusion policy and Streaming-flow policy
- Demo from the authors
- Training strategy
- Challenges – Future work
- Q&A

Motivation

Why imitation-learning?

- When do robots need to learn from a human?
- Why don't we just copy the expert movements 100%?
- And how can we **collect expert data**?

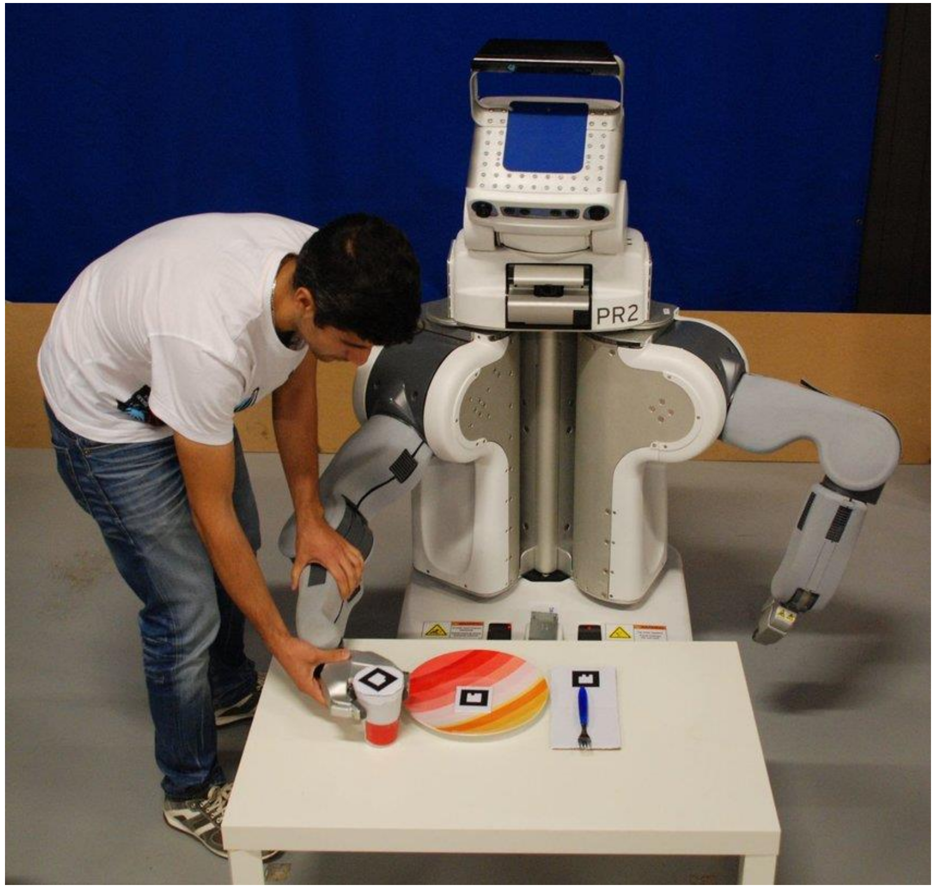


Figure 02: kinesthetic teaching [2]

MOTIVATION

Approaches (SOTA) in **imitation-learning** models:

- Diffusion model in robot learning.
- Streaming-flow: Simplify Diffusion policy, less computational resources required.

Terminology

- **Trajectories**: A sequence of something (actions, states,…)
- **Policy**: How the robot will act given an observation
- **Expert/demonstration**: Someone that can do the task correctly
- **Action space**: all the possible actions (discrete or continuous)
- **Velocity**: A “rule” that tells the robot “From this action, move a little bit in this direction next.”

Training Strategy: Data collecting methods

https://www.youtube.com/watch?v=L3rLT84qqLk

Training Strategy: Data collecting methods

Non- teleoperation: Dex Cap (2025) [3]

More portable, cost-effective

Good for occlusions

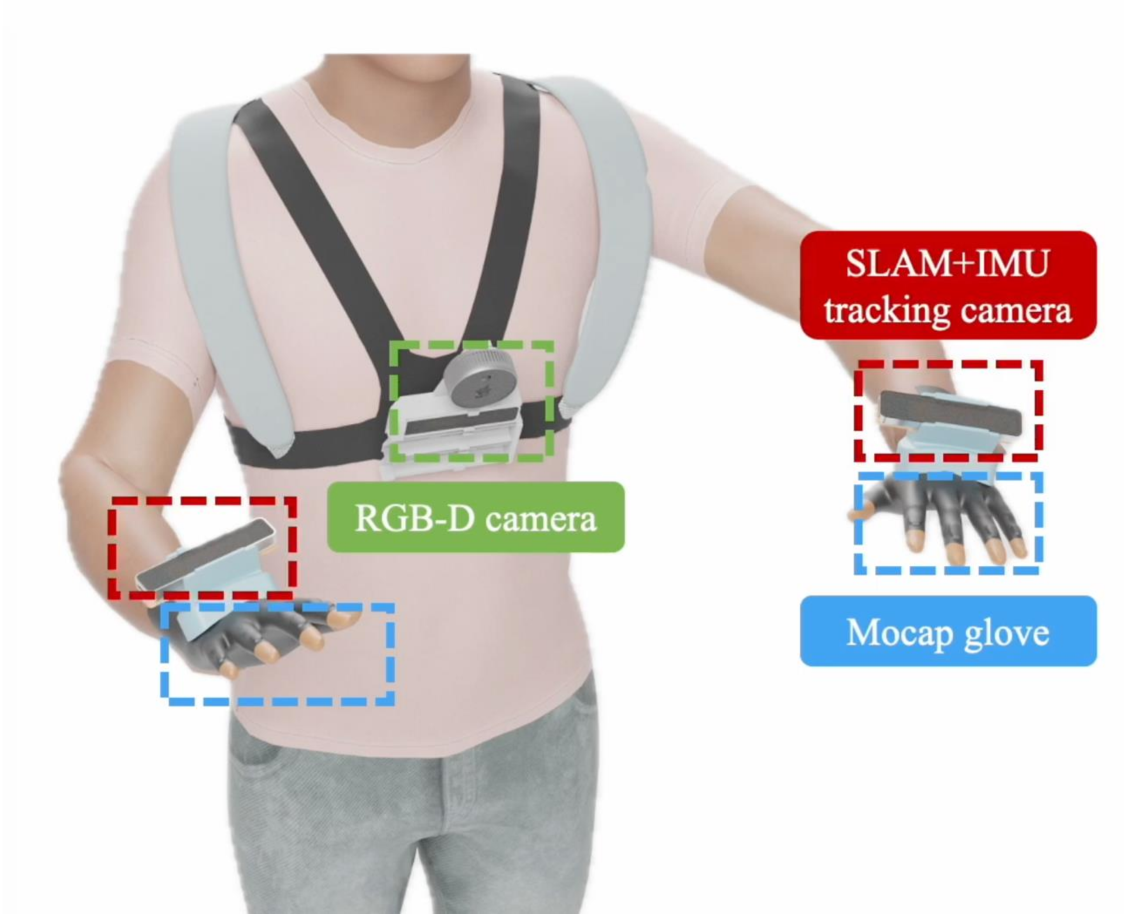


Figure 03: Dex Cap set [3]

Training Strategy: Data collecting methods

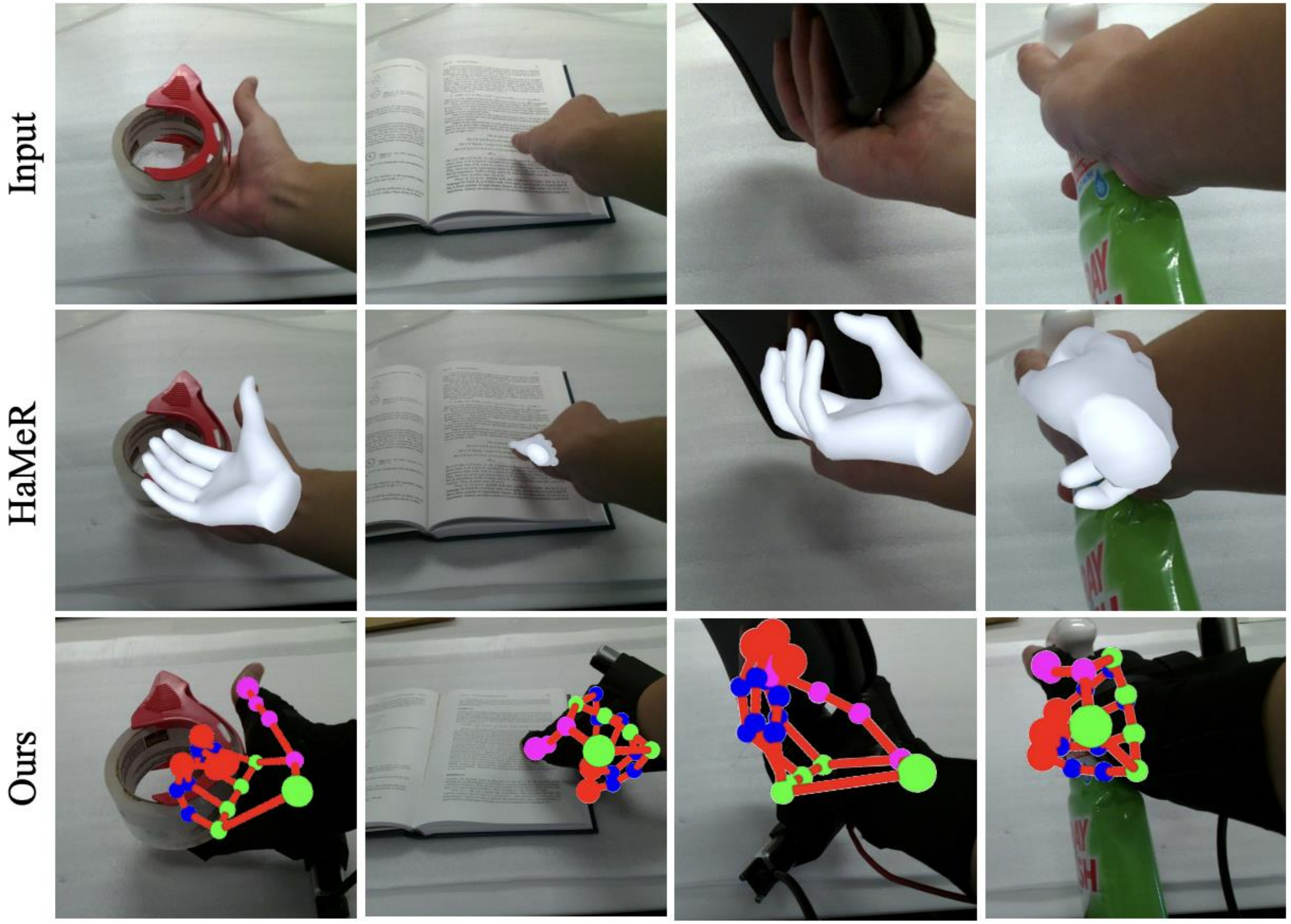


Figure 04: DexCap vs Hamer [3]

Training Strategy: Data collecting methods

Simulation method: DexMimicGen [3]

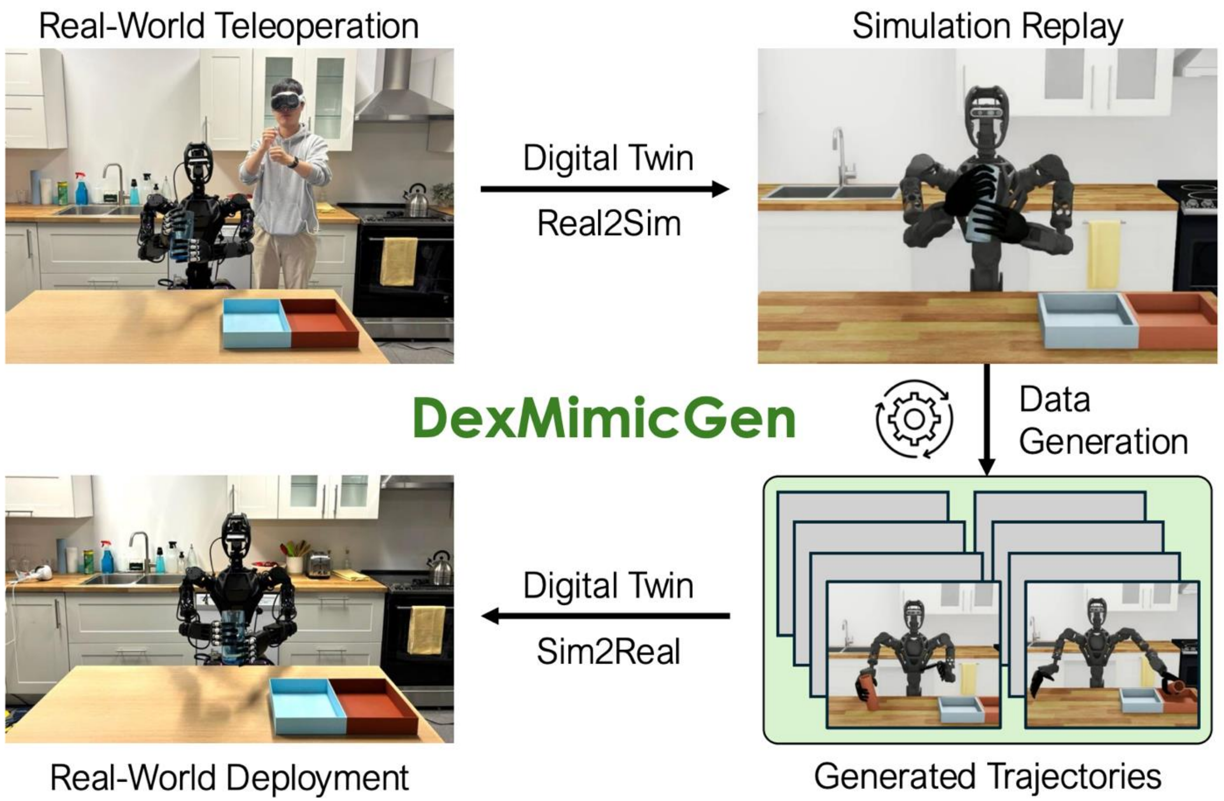
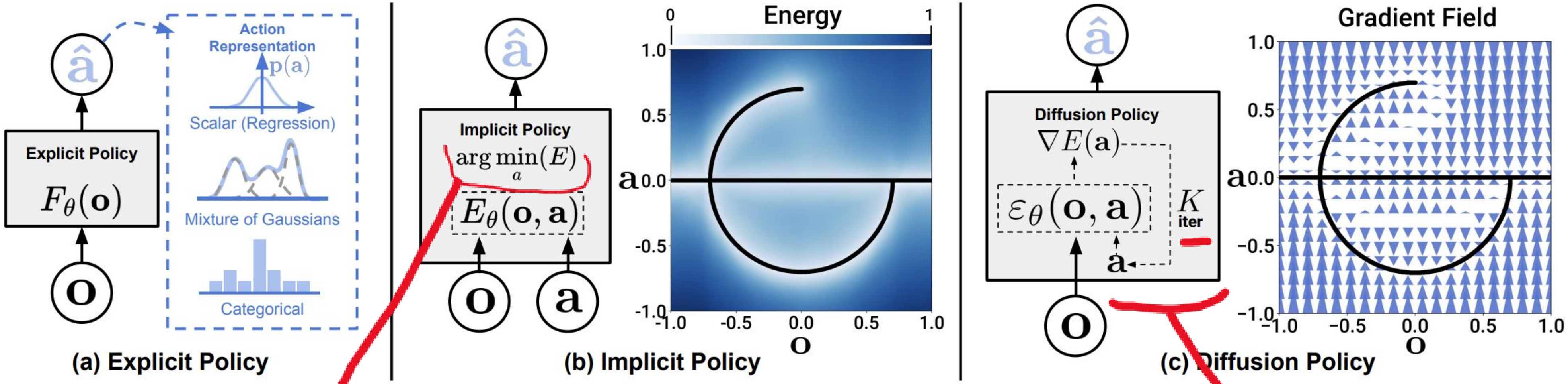


Figure 05: DexMimicGen [3]

Training Strategy: Data collecting methods

	Kinesthetic teaching	Teleoperation	DexCap	Simulation (DexMimicGen)
Pros	• Perfect embodiment alignment • Immediate feasibility check • Low in $$$	• Safety • Immediate feedback • High data-quality	• Can collect data in the wild • Fine-grained dexterity • Low latency • Occlusion-resistant	• Fast data scale • Easy to add variation
Cons	• Physical interference • Slow • Poor with high DoF robots	• Steep operator learning curve • Relatively slower • Higher cost	• Embodiment gap • No immediate feedback	• Sim-to-real domain gap • Data quality • Limited to the sim capabilities

Training Strategy: Diffusion Policy



Action that has the **smallest** energy field?
Q-function without rewards

Denoising loops

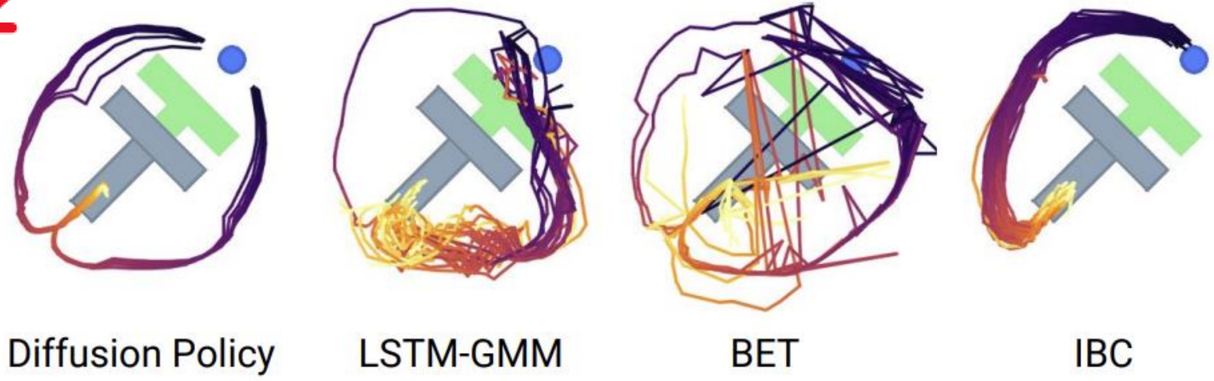
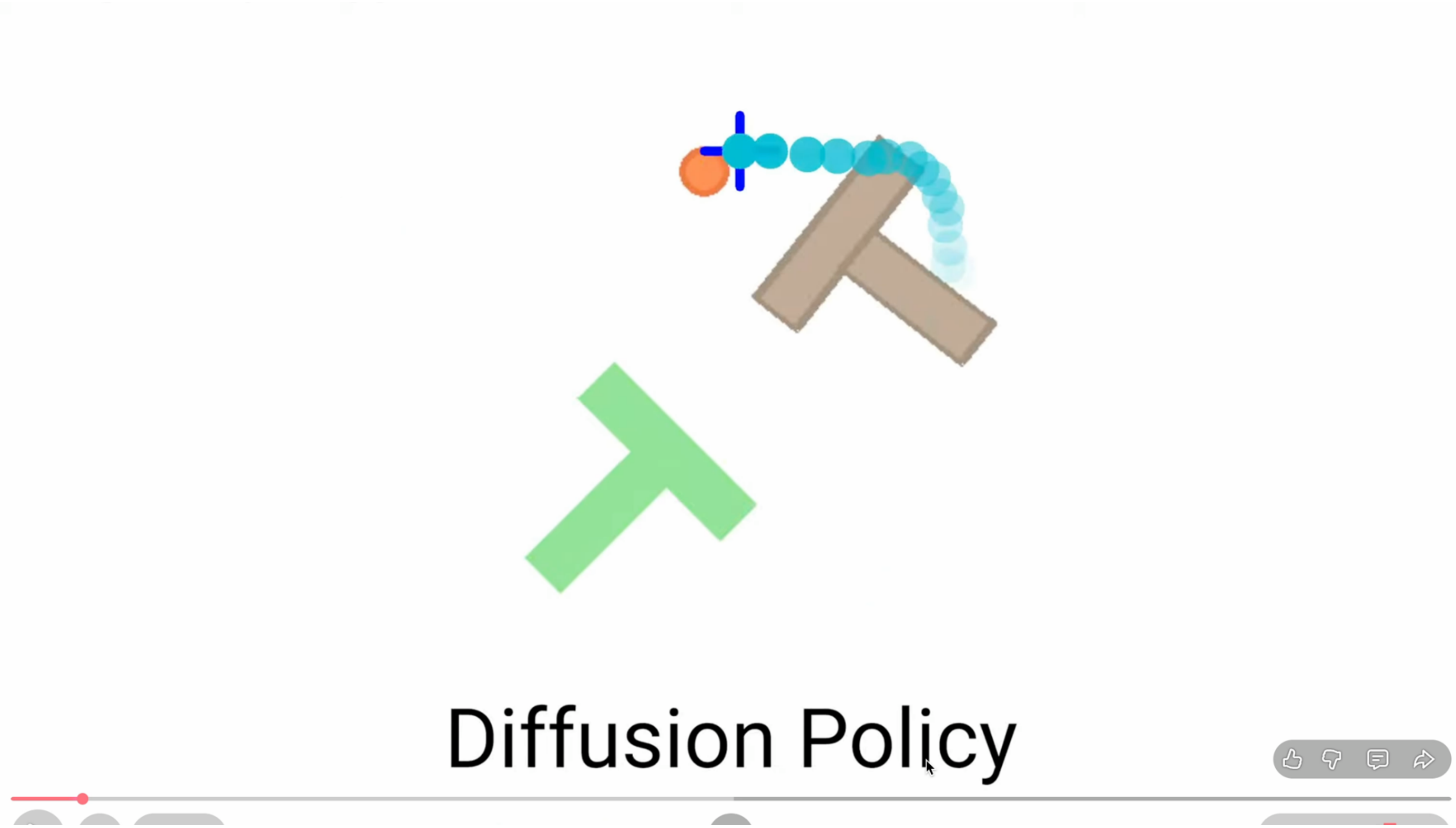


Figure 06: Diffusion Policy and others [5]

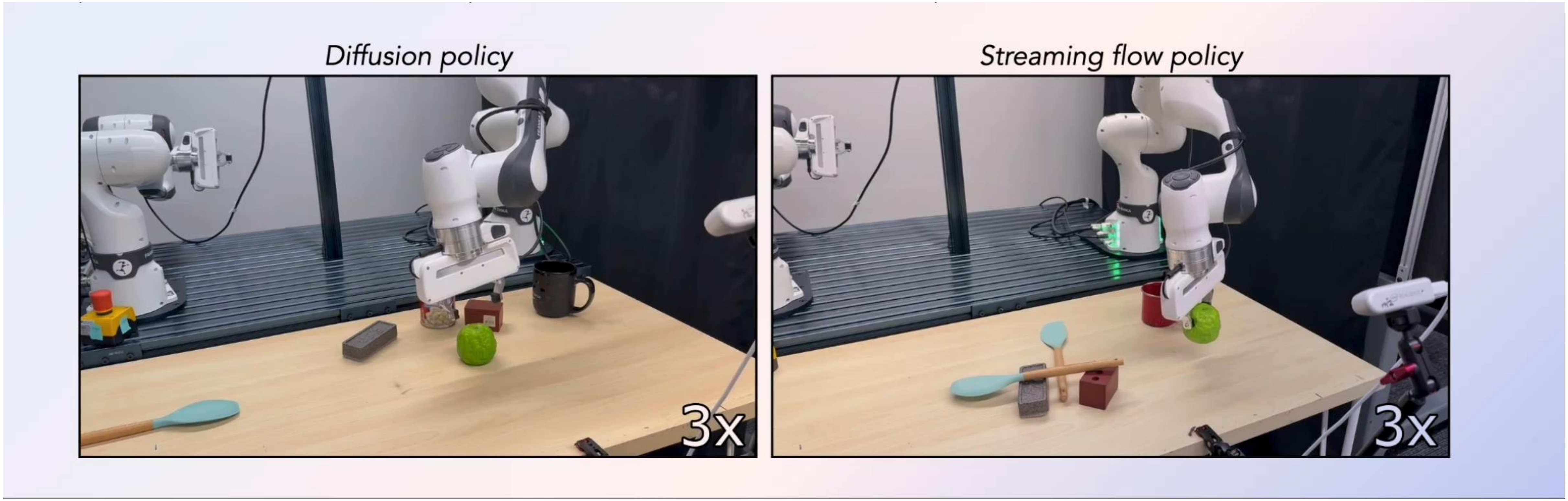
Diffusion Policy in action

https://www.youtube.com/watch?v=gMceHMO9IlU



DEMO - Diffusion VS Streaming-flow

Hardware: A 7-DOF Franka FR3 robot arm



https://siddancha.github.io/streaming-flow-policy/

Diffusion or Streaming-flow?

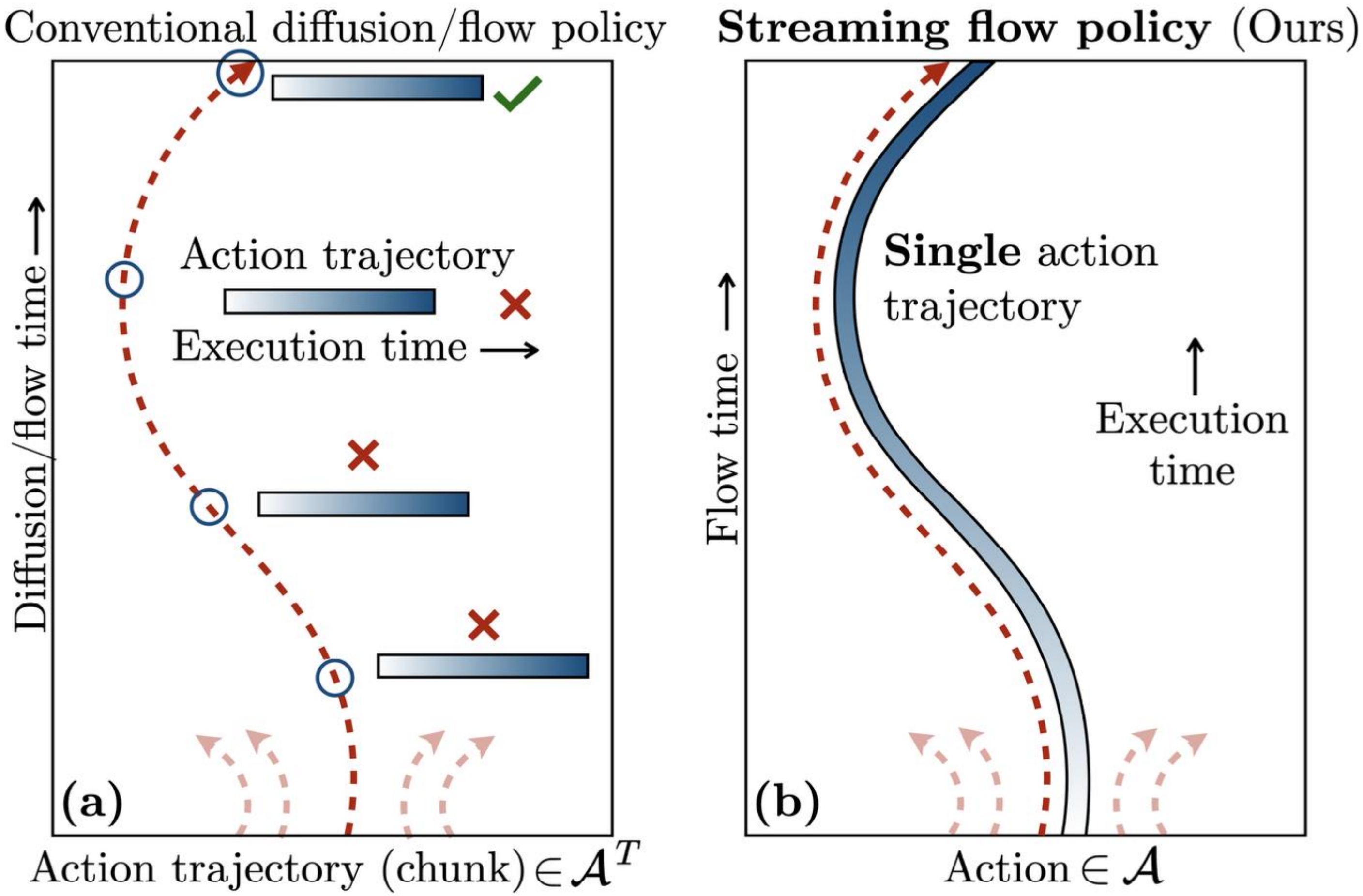


Figure 07: Diffusion vs Streaming-flow [1]

Diffusion or Streaming-flow?

Diffusion Policy	Streaming flow policy
• Generates full action chunks • Higher latency • Needs expert demonstrations • Multi-modality • Strategy: **Diffusion model**	• Generate 1 action on-the-go • Very low latency • Needs expert demonstrations • Multi-modality • **Flow-matching model** (neural ODE)

Training Strategy: Streaming flow

$$\frac{da}{dt} = v_\theta\left(a, t \mid h\right)$$

Time-derivative of action trajectory

action $\in \mathcal{A}$

flow time $\in [0, 1]$

observation history

Initial sampling: $a(0) \sim \mathcal{N}\left(a_{\text{prev}}, \sigma_0^2\right)$

ODE integration: $a(t) = a(0) + \int_0^t v_\theta\left(a(t'), t' \mid h\right) \mathrm{d}t'$

LIMITATION

SFP has been winning (diffusion) in:

- Latency and smoothness
- Computational resources

SFP is still struggling with:

- Accuracy:
  - Each individual action at each time looks expert-like BUT the order and relationships between actions may be wrong.

LIMITATIONS

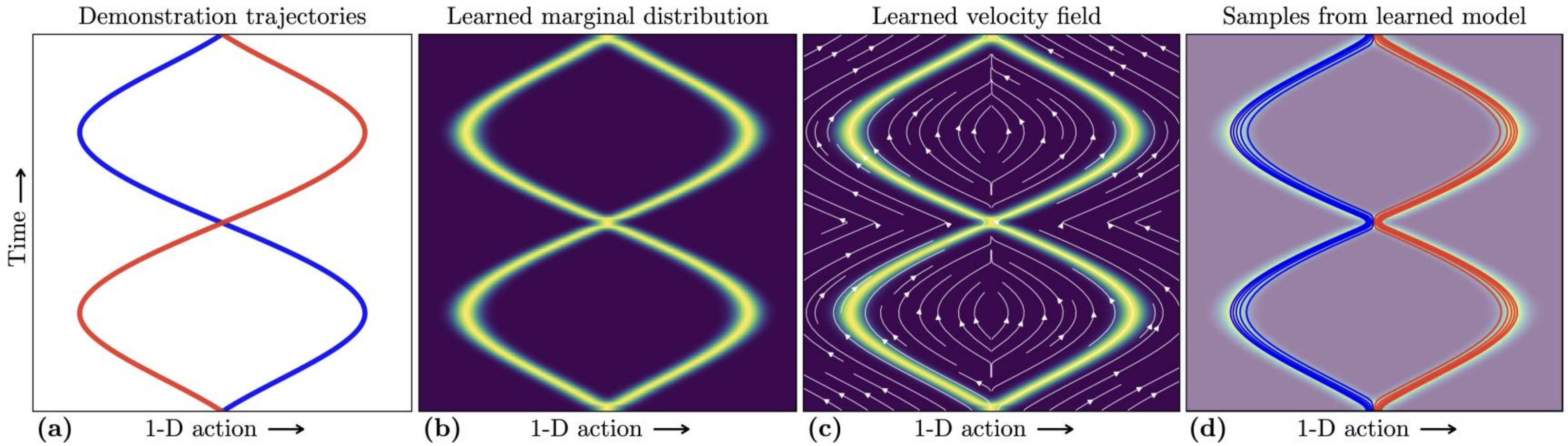


Figure 08: The loss of fidelity to the joint distribution [1]

Diffusion preserves the entire structure of the expert trajectories

SFP only guarantees per-timestep correctness, not full trajectory structure

FUTURE WORK

Improvement for the Diffusion policy?

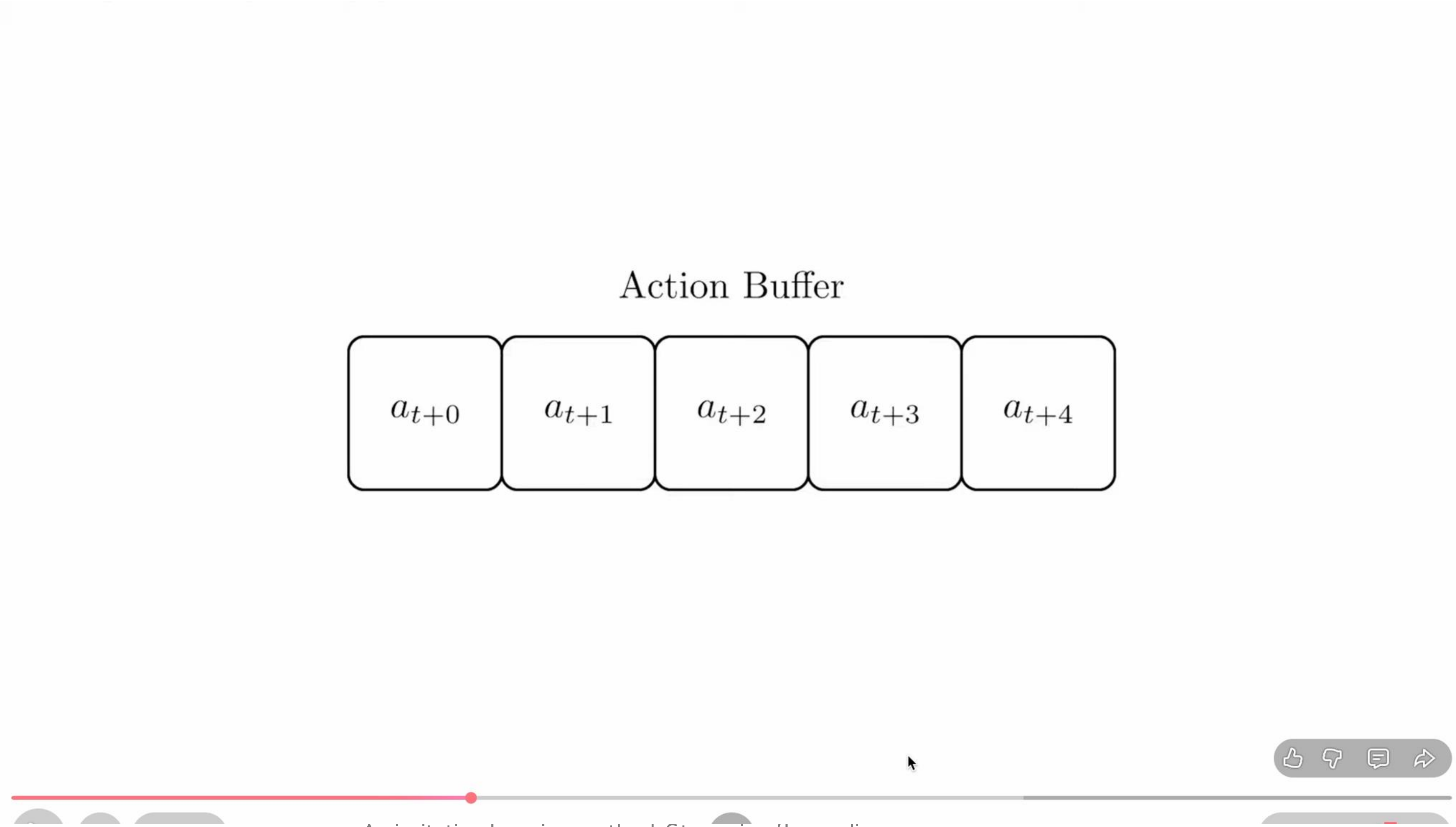
- Streaming diffusion policy – Having buffering actions

Improvements of the Streaming-flow policy?

- On-going work, returning a chunk of actions but smaller

Streaming diffusion policy

https://www.youtube.com/watch?v=gMceHMO9IlU



To be continued in the method-level discussion

- Diffusion policy architecture: CNN-based vs Transformer
- Streaming flow policy architecture: How do we collect expert velocity
- Models testing / demo.

Thank you for your listening!!

Have a good day

REFERENCES

- [1] Sunshine Jiang, Xiaolin Fang, Nicholas Roy, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Siddharth Ancha. Streaming Flow Policy: Simplifying diffusion/flow-matching policies by treating action trajectories as flow trajectories. arXiv preprint arXiv:2505.21851, 2025.
- [2] Abdo, Nichola & Spinello, Luciano & Burgard, Wolfram & Stachniss, Cyrill. (2014). Inferring what to imitate in manipulation actions by using a recommender system. Proceedings - IEEE International Conference on Robotics and Automation. 1203-1208. 10.1109/ICRA.2014.6907006.
- [3] Wang, Chen & Shi, Haochen & Wang, Weizhuo & Zhang, Ruohan & Fei-Fei, Li & Liu, Karen. (2024). DexCap: Scalable and Portable Mocap Data Collection System for Dexterous Manipulation. 10.15607/RSS.2024.XX.043.
- [4] S. H. Høeg, Y. Du and O. Egeland, "Fast Policy Synthesis with Variable Noise Diffusion Models," 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 2025, pp. 4821-4828, doi: 10.1109/ICRA55743.2025.11127858.