Act to See, See to Act: Diffusion-Driven Perception-Action Interplay for Adaptive Policies

Jing Wang[1], Weiting Peng[2], Jing Tang[2], Zeyu Gong[2], Xihua Wang[1], Bo Tao[2], Li Cheng[1]

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

[1]University of Alberta, [2]Huazhong University of Science and Technology

Policy Learning

Learn a mapping from observations to actions that replicates or optimizes behaviors.

Category	Core Idea	Methods
Behavioral Cloning (BC)	Supervised mapping from demonstrations.	LSTM-GMM, IBC, VLA Pipelines
Reinforcement Learning (RL)	Learn via reward feedback.	PPO, SAC
Generative Policies	Model action distributions for smoother control	Diffusion Policy, Flow Matching Methods

- **Policy learning**
  - enables continuous and data-driven control from human demonstrations or self-play.
  - is foundational to robotic imitation, autonomous driving, and embodied intelligence.
- **BC** learns a direct mapping from observations to expert actions through supervised imitation, but lacks exploration or correction when faced with unseen states.
- **RL** optimizes action selection via reward-driven exploration, achieving high adaptability but at the cost of massive data and unstable convergence in high-dimensional control.
- **Generative Policies** model the distribution of actions to ensure temporal smoothness and multimodality, but still treat perception as deterministic.

Codevilla et al., "End-to-end driving via conditional imitation learning." ICRA, 2018.
Haarnoja et al., "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." ICML, 2018.
Florence et al., "Implicit behavioral cloning." CoRL, 2022.
Chi et al., "Diffusion policy: Visuomotor policy learning via action diffusion." IJRR, 2023.
Huang et al., "BiLLM: Pushing the Limit of Post-Training Quantization for LLMs." ICML, 2024.

Limitations of Existing Policy Learning

Decoupled Perception & Action

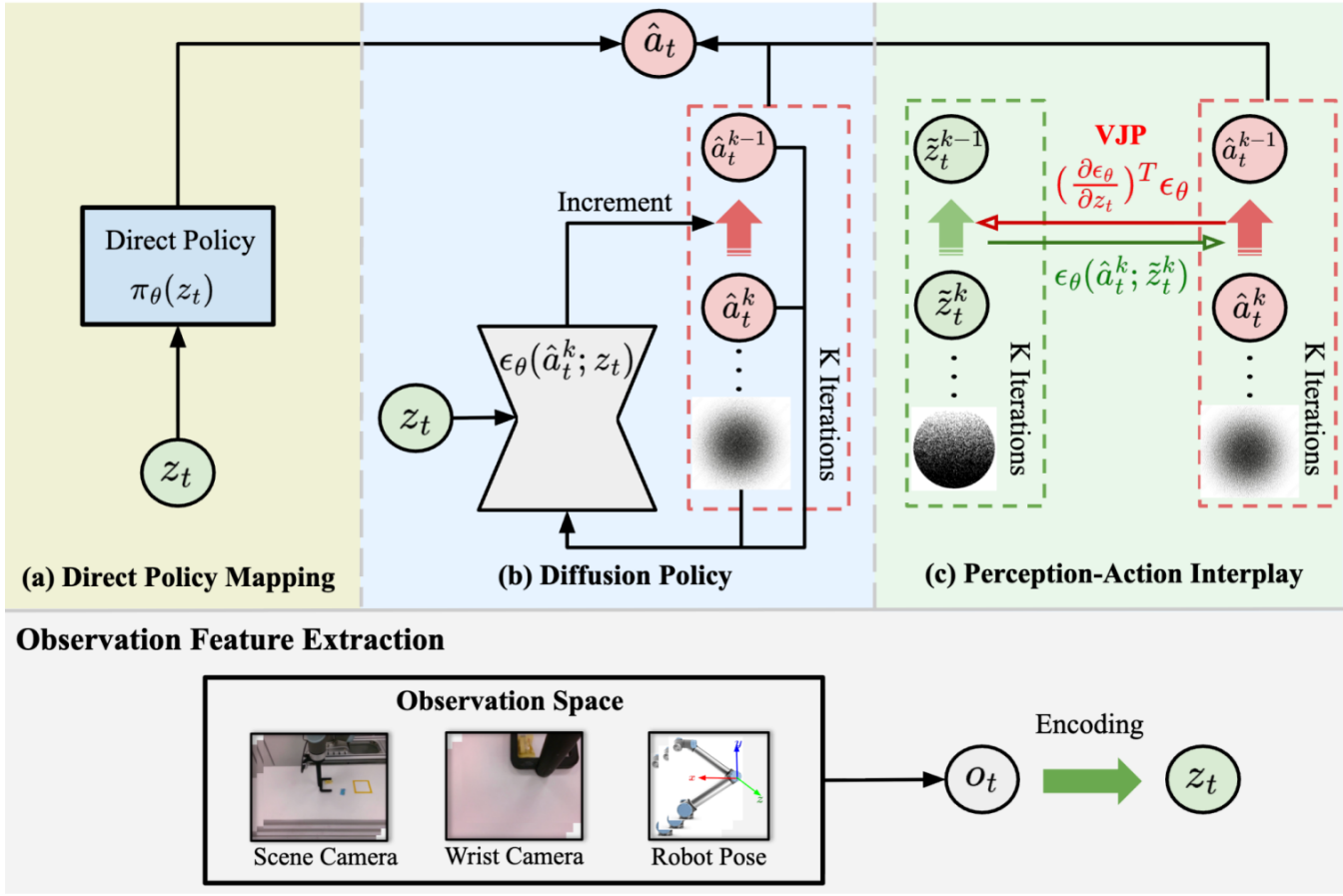
- Observation features are extracted **once** and **held fixed** during an entire action sequence.
- No feedback from the generated actions to refine perception.
- Results in discontinuous, or context-insensitive behavior.

Lack of Mutual Adaptation

- Generative policies capture temporal smoothness in actions, but perception remains static, ignoring causal reciprocity between seeing and acting.

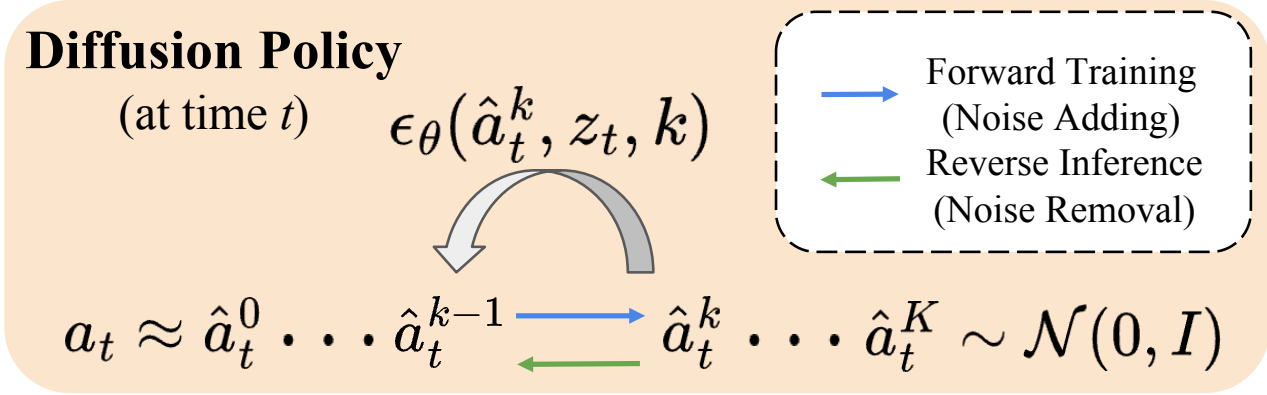
Brittle under Dynamic or Partial Observability

- When the environment changes (lighting, occlusion, moving objects), static perception cannot adjust mid-trajectory.
- Leads to failures in dynamic manipulation and poor generalization in real-world robotics.



➡ We propose a novel perception-action interplay via Vector-Jacobian Product, **Action-Guided Diffusion Policy (DP-AG)**

Context on Diffusion Policy



Conventional Imitation Learning

- Learns a direct mapping $a_t = \pi_\theta(f_\psi(o_t))$
- Actions are predicted deterministically from static observation features.
- Results in discontinuous or jerky actions.

Diffusion Policy: A Generative View of Actions

- Model the distribution $p(a_t|o_t)$ via a **Denoising Diffusion Process:**
  - Start from random noise $a_t^K \sim \mathcal{N}(0, I)$
  - Iteratively denoise over K steps using a noise predictor $\epsilon_\theta(a_t^k, z_t, k)$
  - Each step refines actions, which ensures **temporal smoothness** and **continuity**.
  - The objective is to train the noise predictor to denoise actions at each diffusion step, so that the model learns to generate smooth action trajectories from pure noise conditioned on observations.

$$\mathcal{L}_{\text{DP}}(\theta, \psi) = \mathbb{E}_{(o_t, a_t)\sim\mathcal{D}, k\sim\mathcal{U}(1,K)} \left[\|\epsilon_\theta(a_t^k, f_\psi(o_t), k) - \epsilon\|_2^2 \right]$$

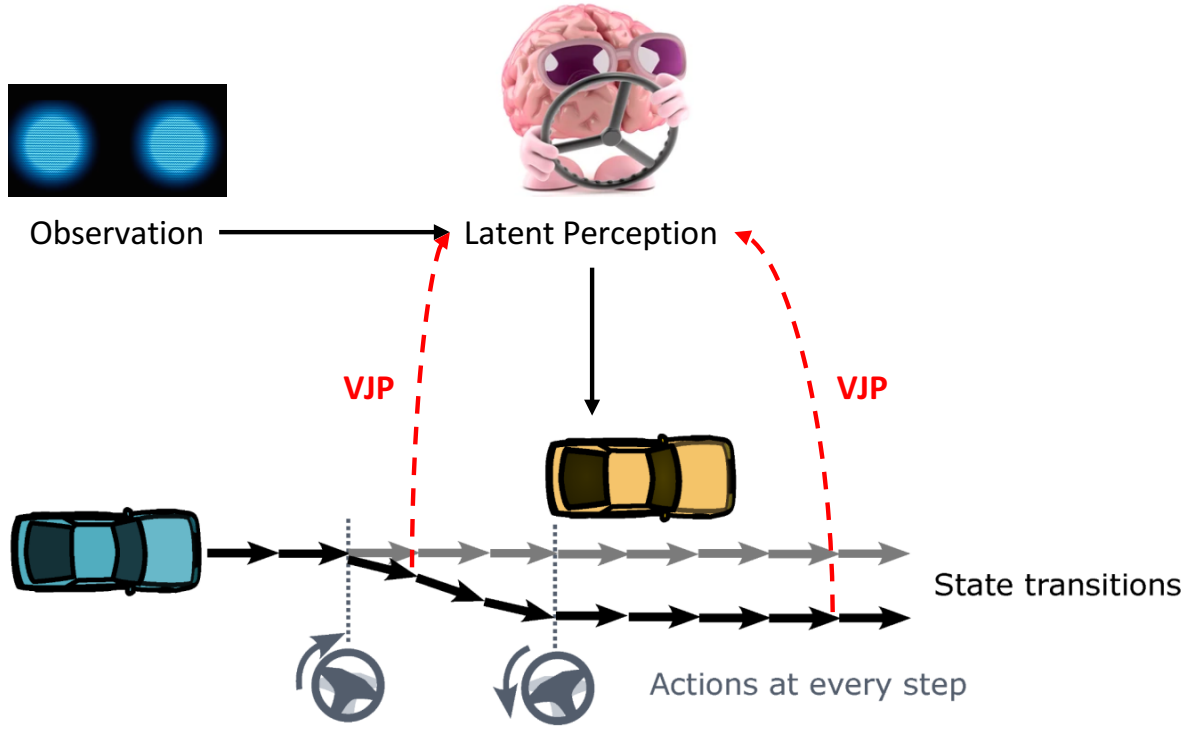
Converts discrete observations into continuous action trajectories.

- Produces smooth and more natural motions.
- Handles multimodal action distributions.
- Outperforms direct mapping methods in complex manipulation.

➡ **Remaining Gap:** Environmental perception does not adapt as actions evolve.

Motivations of Our DP-AG

- Existing methods decouple perception and action by keeping observation features fixed throughout action sequence generation, **breaking the natural feedback loop**.
- When driving on a straight road, even if the scenery ahead does not change, our perception does. As we start turning the wheel slightly, our focus shifts toward the lane edge, mirrors, and road curvature to guide the motion.
- In the same way, DP-AG updates its **internal perception** based on **the feedback from its own action sequence**, continuously **reinterpreting** fixed observations to maintain smooth and adaptive control.



DP-AG: Adaptability from a Variational View

- Real-world observations are noisy and ambiguous. To capture this uncertainty, we represent each observation with a Gaussian posterior:

$$q_\phi(z_t|o_t) = \mathcal{N}(\mu_\phi(z_t), \sigma^2_\phi(z_t))$$

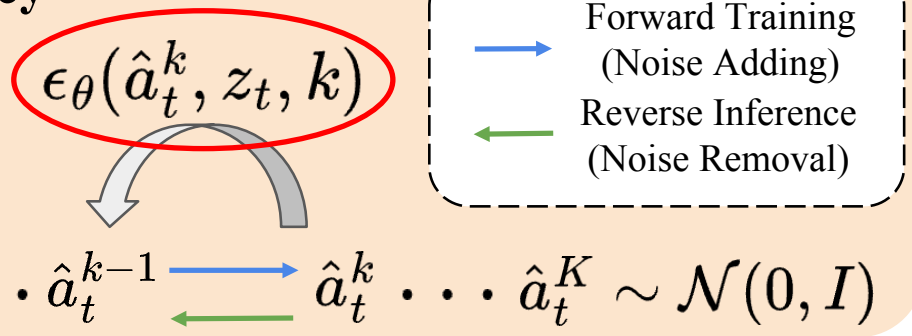
- We enable differentiable gradient flow from action to perception via the reparameterization trick, which injects structured uncertainty while keeping sampling differentiable:

$$z_t = \mu_\phi(z_t) + \sigma_\phi(z_t) \odot \epsilon; \quad \epsilon \sim \mathcal{N}(0, I)$$

- The observation is no longer deterministic but represented as a distribution that captures the policy's confidence in the current perception. This allows the agent to reason about uncertainty; however, the observation features remain fixed during action generation.

Diffusion Policy

(at time t)



In *Diffusion Policy*, actions are progressively refined through iterative denoising while observation features remain fixed.
Can we **replace the random noise with action-guided noise** so that perception adapts dynamically alongside action generation?

DP-AG: From Static Latent to Action-Guided Evolution

- The Vector–Jacobian Product (VJP) measures how small changes in perception would affect the predicted action noise:

$$\mathrm{VJP}(\hat{a}_t^k, z_t) = \left(\frac{\partial \epsilon_\theta(\hat{a}_t^k, z_t, k)}{\partial z_t} \right)^\top \epsilon_\theta(\hat{a}_t^k, z_t, k)$$

- Instead of adding random noise, DP-AG uses this VJP as a **structured feedback signal** that tells perception which direction to move to reduce action uncertainty:

$$d\tilde{z}_t^k = \mathrm{VJP}(\hat{a}_t^k, z_t)\, dt + \sigma_\phi(z_t)\, dW_t$$

- Thus, rather than using a static variational posterior, we reparameterize it with an **action-guided noise term from DP**, allowing perception to evolve with ongoing action refinements:

$$\tilde{z}_t^k = \mu_\phi(z_t) + \gamma \sigma_\phi(z_t) \odot \mathrm{VJP}(\hat{a}_t^k, z_t)$$

As diffusion unfolds, this feedback keeps perception phase-aligned with action denoising, enabling the agent to continuously reinterpret what it sees through its own actions.

Overall Framework and Cycle-Consistency Contrastive Learning

$$\varepsilon_k = \epsilon_\theta(\hat{a}_t^k, z_t, k)$$

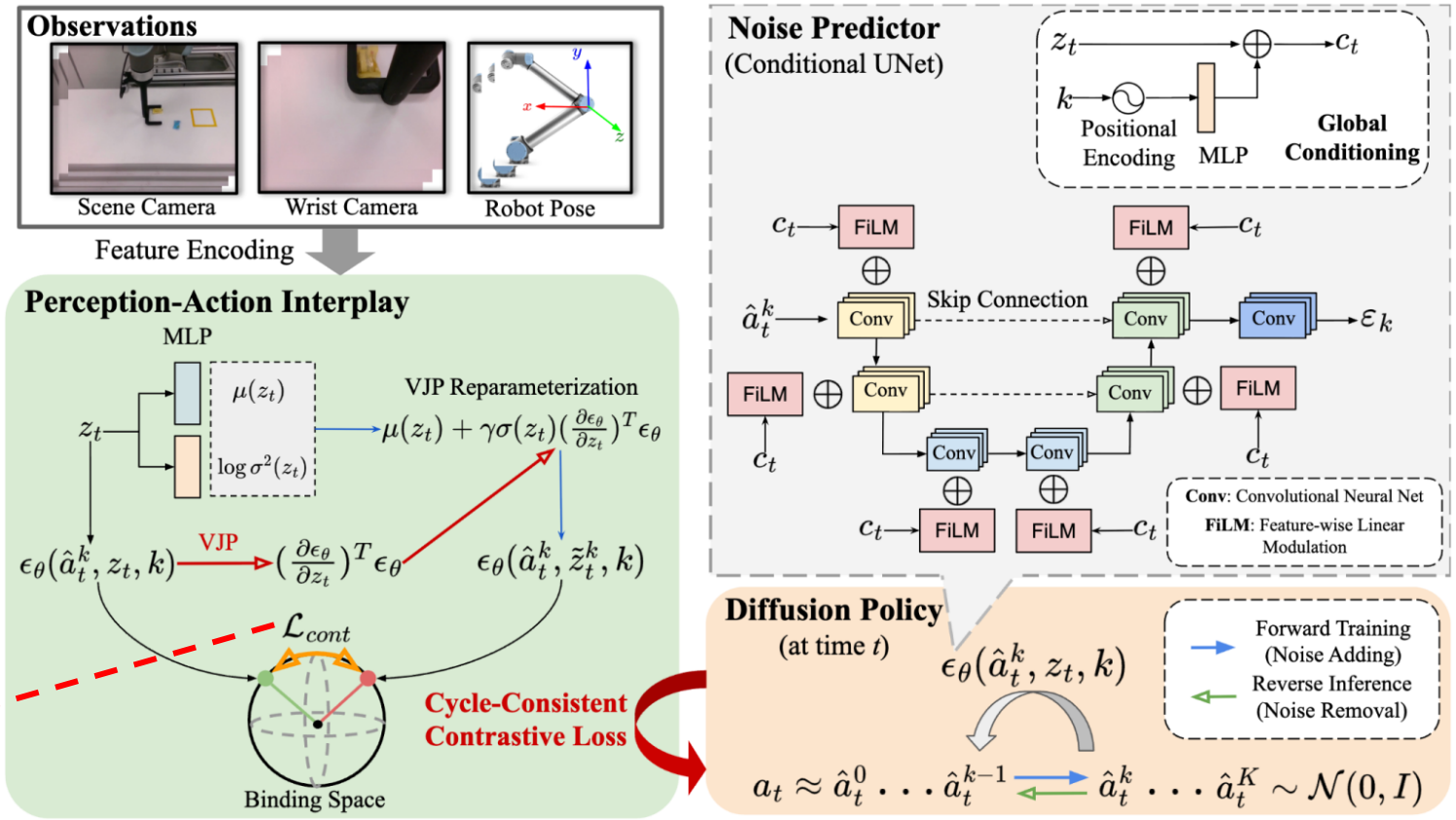
the noise predicted from the static latent.

$$\tilde{\varepsilon}_k = \epsilon_\theta(\hat{a}_t^k, \tilde{z}_t^k, k)$$

the noise predicted from the VJP-guided latent.

pulls matched pairs $(\epsilon_i^k, \tilde{\epsilon}_i^k)$ closer while pushing apart mismatched pairs, promoting consistency between static and dynamic perceptions.

$$\mathcal{L}_{\text{cont}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp\left(\text{sim}\left(\varepsilon_k^i, \tilde{\varepsilon}_k^i\right)/\tau\right)}{\sum_{j\neq i}\exp\left(\text{sim}\left(\varepsilon_k^i, \tilde{\varepsilon}_k^j\right)/\tau\right)}$$



Intuitions of Cycle-Consistency Contrastive Learning

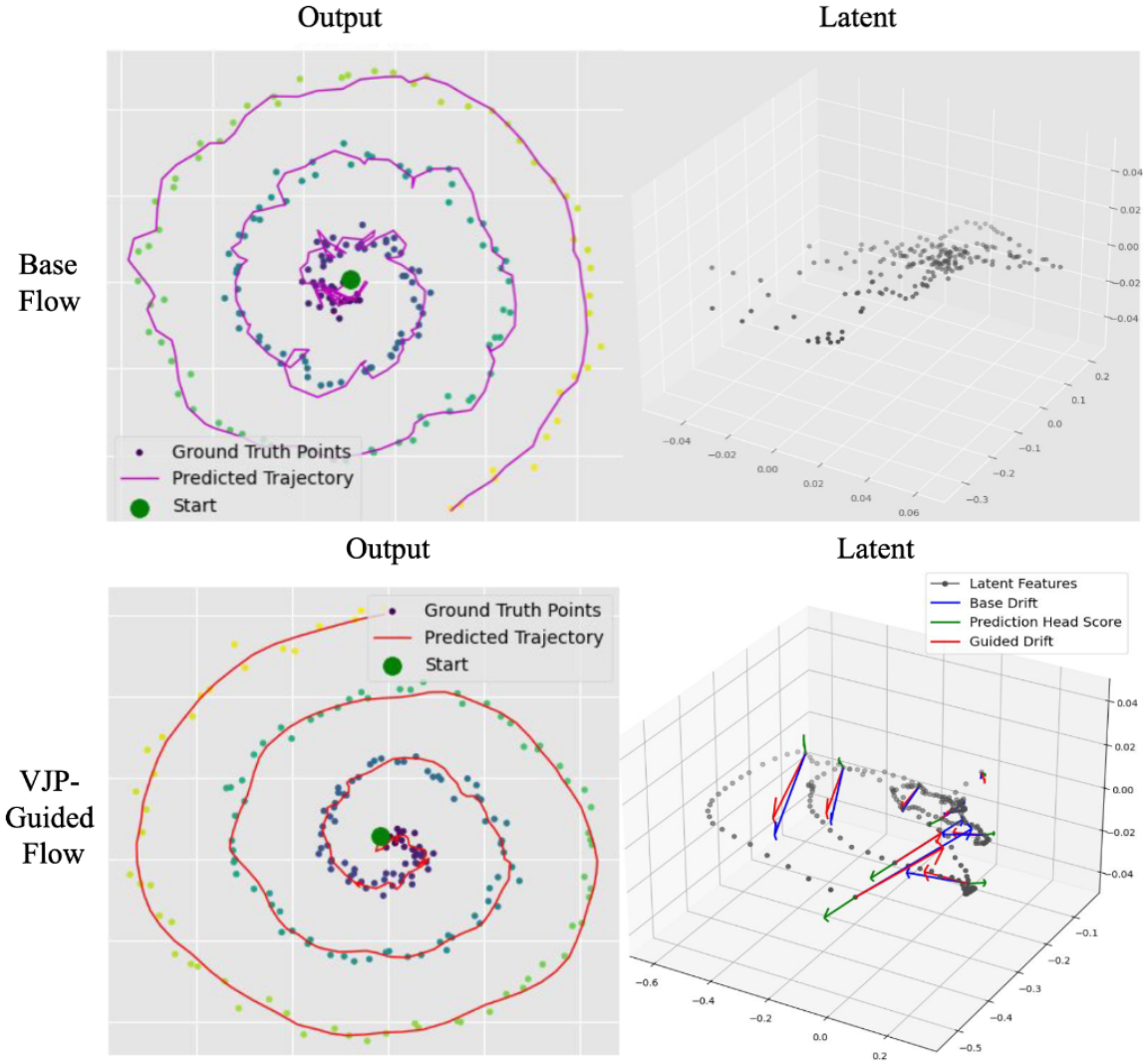
- Contrastive loss anchors evolving latents to their original semantics, preventing excessive drift during VJP-guided updates.
- It enforces cycle consistency between perception and action, ensuring both evolve coherently throughout diffusion.

Synthetic Experiment: Demonstrating Mutual Continuities

Dataset: Irregular spirals (from Neural ODE).
Task: Predict future positions under irregular time sampling.
Base Flow: LSTM + MLP.
VJP-Guided Flow: Latents evolve under a SDE guided by VJP.

Model	MSE (↓)	Latent Dynamics	Predictions
Base Flow	0.0095	Scattered latent states	Irregular, discontinuous paths
VJP-Guided Flow	**0.0052**	Structured latent manifold	Smooth, coherent trajectories

- VJP feedback shapes latent manifold to align with regression predictions, verifying our **mutual smoothness theorem**.
- Latents and outputs evolve **in synchrony**, maintaining local continuity in both spaces.



Regression results on irregular spirals

Chen et al., ”Neural Ordinary Differential Equations." NeurIPS, 2018.

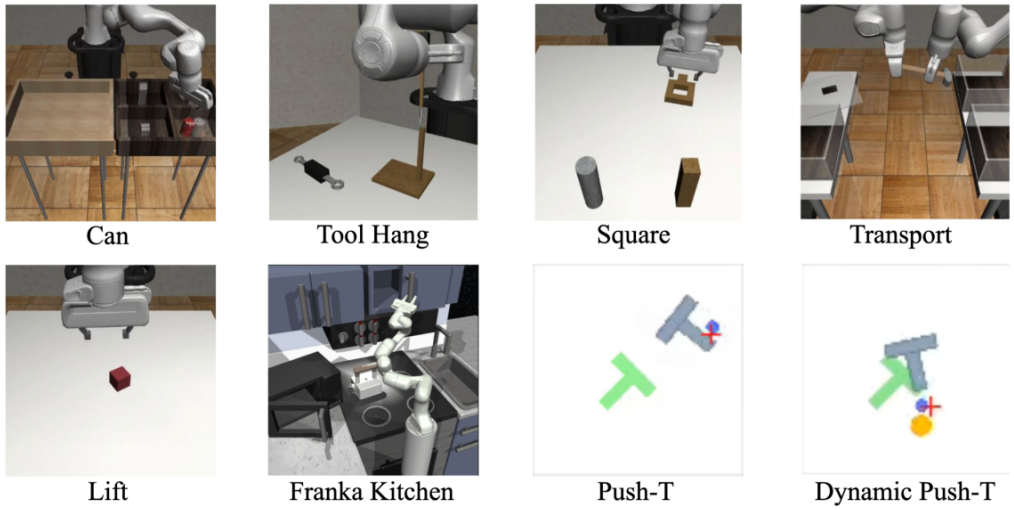
Experiments on Simulation Benchmarks

Type	Method	Push-T		Dynamic Push-T
		img	kp	img
Mapping	LSTM-GMM [Mandlekar, 2022]	0.69±0.02	0.67±0.03	0.34±1.24
	IBC [Florence, 2022]	0.75±0.02	0.90±0.02	0.52±0.98
	BET [Shafiullah, 2022]	0.80±0.02	0.79 ±0.02	0.58±1.35
Flow	FlowPolicy [Zhang, 2025]	0.85±0.01	0.88±0.01	0.53±0.88
	AdaFlow [Hu, 2024]	0.87±0.02	0.91±0.01	0.67±0.79
Diffusion	DP [Chi, 2023]	0.87±0.04	0.95±0.03	0.65±0.85
	DP-AG (ours)	**0.93**±0.02	**0.99**±0.01	**0.80**±0.53

Target coverage score on Push-T and Dynamic Push-T tasks

Type	Method	Lift		Can		Square		Transport		ToolHang	Franka Kitchen			
		ph	mh	ph	mh	ph	mh	ph	mh	ph	t1	t2	t3	t4
Mapping	LSTM-GMM [Mandlekar, 2022]	**1.00**	**1.00**	**1.00**	0.98	0.82	0.64	0.88	0.44	0.68	1.00	0.90	0.74	0.34
	IBC [Florence, 2022]	0.94	0.39	0.08	0.00	0.03	0.00	0.00	0.00	0.00	0.99	0.87	0.61	0.24
	BET [Shafiullah, 2022]	**1.00**	**1.00**	**1.00**	**1.00**	0.76	0.68	0.38	0.21	0.58	0.99	0.93	0.71	0.44
Flow	FlowPolicy [Zhang, 2025]	0.98	0.95	0.98	0.98	0.86	0.90	0.88	0.82	0.85	0.96	0.86	0.95	0.87
	AdaFlow [Hu, 2024]	**1.00**	**1.00**	**1.00**	0.96	0.98	0.96	0.92	0.80	0.88	0.99	0.89	0.92	0.83
Diffusion	DP [Chi, 2023]	**1.00**	**1.00**	**1.00**	**1.00**	0.98	0.98	**1.00**	0.89	0.95	**1.00**	**1.00**	**1.00**	0.99
	DP-AG (ours)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.94**	**0.98**	**1.00**	**1.00**	**1.00**	**1.00**

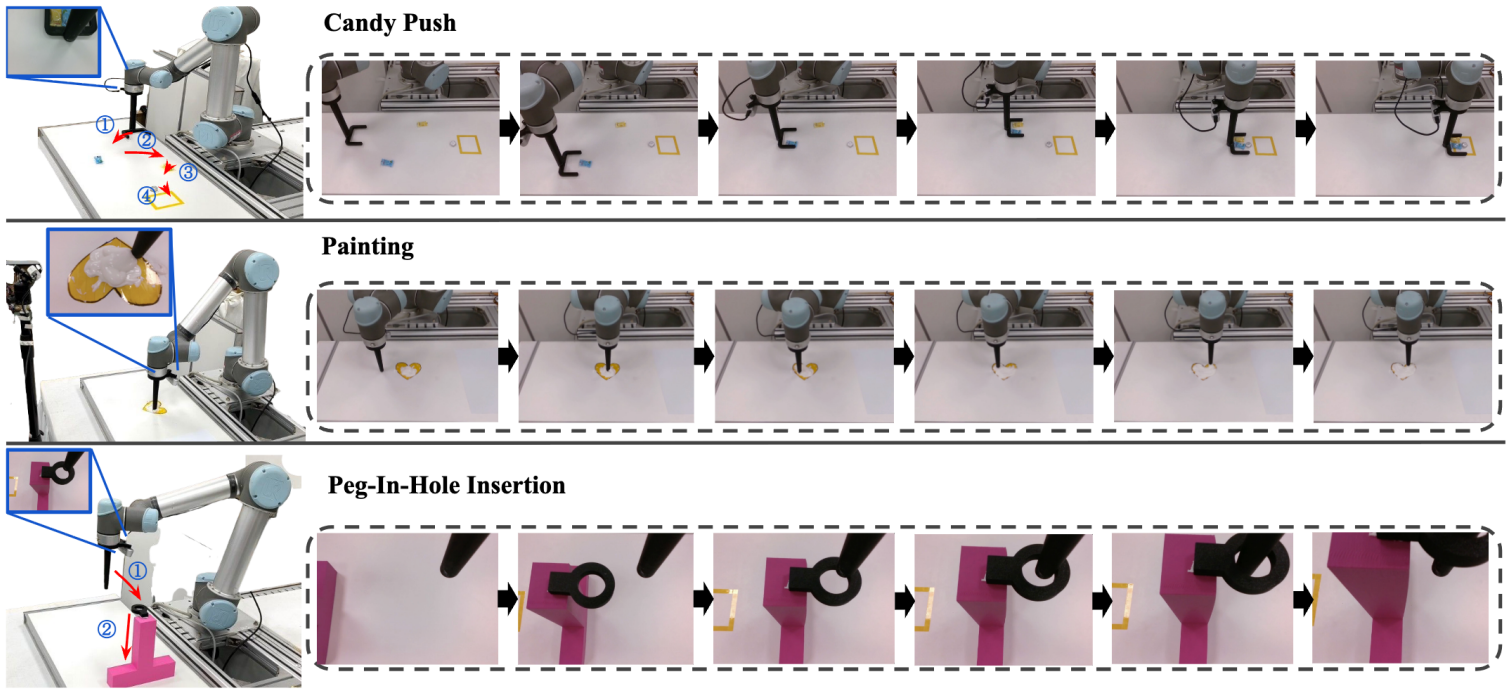
Success rates across Robomimic and Franka Kitchen tasks



Benchmark simulation environments and tasks

- **Compared to Mapping-based methods:** DP-AG captures temporal continuity and avoids abrupt action switching.
- **Compared to Flow-based methods:** DP-AG retains stochastic expressiveness for better uncertainty handling.
- **Compared to the baseline DP:** DP-AG transforms static perception into an adaptive loop, achieving higher success and smoother control.

Experiments on Real-World Tasks (UR5 Deployment)



Real-world evaluation on a UR5 robot arm across three manipulation tasks

Task	Method	Success Rate (%)	Smoothness (Avg. Jerk)	IoU (%)	Time to Complete (s)
Painting	DP	–	0.083 ± 0.014	68.9 ± 5.2	49.5 ± 4.1
	DP-AG	–	$\mathbf{0.032 \pm 0.009}$	$\mathbf{92.1 \pm 3.4}$	$\mathbf{18.0 \pm 3.2}$
Candy Push	DP	65.0 ± 8.4	0.107 ± 0.016	–	24.0 ± 3.9
	DP-AG	$\mathbf{90.0 \pm 5.5}$	$\mathbf{0.039 \pm 0.011}$	–	$\mathbf{9.5 \pm 2.6}$
Peg-in-Hole	DP	0.0 ± 0.0	0.096 ± 0.017	–	–
	DP-AG	$\mathbf{85.0 \pm 6.0}$	$\mathbf{0.036 \pm 0.008}$	–	$\mathbf{13.0 \pm 2.1}$

Performance on Real-World UR5 Tasks

- **Performance:** DP-AG outperforms the baseline DP in accuracy, smoothness, and adaptability across all real-world tasks.
- **Smooth and Stable Control:** Reduced jerk and consistent trajectories confirm latent–action smoothness in physical execution.
- **Emergent 3D Reasoning from 2D Inputs:** In the peg-in-hole task, VJP feedback allows the model to infer geometric alignment cues without depth information.
- **Robustness Beyond Simulation:** DP-AG maintains generalization under sensor noise and real-world dynamics.

Summary

- **We rethink imitation learning through perception-action interplay.** Instead of treating perception as fixed and passive, we make it dynamic, which enables the agent to *see through its own motion* as actions unfold.

- **We introduce DP-AG: Action-Guided Diffusion Policy.** Built on variational inference and diffusion policy, DP-AG evolves latent perception via VJP-guided feedback, forming a closed perception-action loop.

- **We ensure coherent co-evolution with cycle-consistent contrastive learning.** This alignment keeps perception anchored while actions refine, enforcing smooth and consistent dynamics in both latent and action spaces.

- **We validate adaptability from simulation to reality.** DP-AG achieves state-of-the-art performance across simulation benchmarks and real UR5 tasks, producing smoother, faster, and more context-aware robot behaviors.