Velocity Field Based Data Augmentation for Corrective Imitation Learning

Shiping Ma[1,2], Sayantan Auddy[1] and Marc Toussaint[1,2]

***Abstract*— While imitation learning (IL) has demonstrated strong performance in a variety of tasks, policies trained purely from demonstrations often suffer from distribution mismatch during execution. Although distribution shift in IL has been widely studied, existing approaches often provide limited coverage of recovery behaviors from off-trajectory states. In this abstract, we collect demonstrations using a motion-planning solver in simulation and then construct a velocity field around the trajectories to generate additional recovery behaviors. We investigate whether the proposed approach can enhance the robustness of imitation learning.**

I. INTRODUCTION

Imitation learning (IL) is widely used in robot manipulation due to its simplicity as a supervised learning paradigm and its high sample efficiency, and it has demonstrated reliable performance on complex tasks. However, IL still suffers from covariate shift during execution [1]. Even small deviations in the policy output can accumulate over time, causing the system to drift into out-of-distribution states and leading to task failure. This issue is fundamentally caused by insufficient state coverage in the training dataset.

Demonstration data can be collected through human teleoperation, which is easy to deploy but inefficient for acquiring large-scale datasets. In contrast, large-scale data can be collected more easily in simulation; however, such data is often overly smooth and lacks noise [2], resulting in limited state coverage. As a consequence, policies trained purely on simulation data struggle to learn stabilizing behaviors at off-trajectory states that are rarely observed during training.

To address this problem, several works inject system noise during data collection to improve robustness [2], [3]. [4]–[6] synthesize off-trajectory states together with corrective actions to stabilize the system around the demonstrated trajectory. By generating states close to the original trajectories along with corresponding corrective actions, these methods have been shown to alleviate covariate shift in practice, particularly under relatively small perturbations. Another line of research focuses on learning stable dynamics [7]–[10], which learns a velocity field that guides the system back towards the trajectory, which can promise robustness under larger perturbation. In contrast, some works directly construct velocity fields for manipulation tasks without learning [11]. Data augmentation expands the training distribution with additional corrective samples, whereas stable dynamics methods aim to encode corrective behavior through a learned dynamical structure.

Motivated by these ideas, we study whether a simple corrective velocity field can be used as a mechanism for data augmentation in simulation, generating recovery-like state–action pairs beyond nominal demonstrations.

II. METHOD

In this extended abstract, we investigate whether velocity-field-based data augmentation can improve the robustness of the policy learned from simulation data. The augmentation is performed through simulation rollouts, which allows additional sensory observations (e.g., images) to be recorded alongside state–action pairs, making the pipeline naturally compatible with training vision-based policies.

A. Data collection in simulation

We generate demonstration trajectories in simulation using a motion-planning solver [12]. The optimization problem can be generally formulated

$$\begin{aligned} \min_x \sum_{t=0}^{T} c(x(t), \dot{x}(t), \ddot{x}(t)) + \psi(x(T)) \\ \text{s.t. } g(x) \leq 0,\ h(x) = 0\ . \end{aligned} \tag{1}$$

The planner optimizes a trajectory objective that combines task-space goals (e.g., end-effector pose constraints) and collision avoidance, producing smooth trajectories [13], [14]. These trajectories serve as nominal demonstrations for imitation learning.

B. Velocity field construction

Given a dataset $\mathcal{D} = \{\tau_i | \tau_i = \{(\mathbf{s}_j, \mathbf{a}_j)\}_{j=1}^{M}\}_{i=1}^{N}$ with N demonstrations, we construct a corrective velocity field around each nominal trajectory to generate recovery data from off-trajectory states. The idea is to provide a simple local mechanism that drives perturbed states back toward a reference trajectory. Given a trajectory τ and a new state $\mathbf{x}$ which is the proprioception state of robot off trajectory, then we can define a velocity field with a distance field $d(\mathbf{x})$ representing the distance from an off-trajectory state $\mathbf{x}$ to τ

$$\dot{\mathbf{x}} = -\alpha \nabla d(\mathbf{x}) + \dot{\mathbf{f}}(t), \tag{2}$$

where $\nabla d(\mathbf{x})$ indicates the velocity direction that flows back to the trajectory, $\dot{\mathbf{f}}(t)$ is the velocity along the trajectory $\mathbf{x}(t)$ and α is a scaling factor. It can be proved like in [11] that an arbitrary state can move back to the trajectory.

*This work has been supported by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG).

[1]Learning & Intelligent Systems Lab, TU Berlin shiping.ma@tu-berlin.de

[2]Robotics Institute Germany (RIG), Germany.

Algorithm 1: Velocity-field-based data augmentation

Generate demonstrations $\mathcal{D} = \{\tau_i\}$ with a motion planner in simulation ; // Equation (1)
Initialize $\mathcal{D}_{\text{aug}} \leftarrow \mathcal{D}$
foreach *trajectory* τ_i **do**
  for $k = 1, \ldots, K$ **do**
    Sample perturbed state $\mathbf{x}$ around τ_i
    Compute the velocity $\dot{\mathbf{x}}$; // Equation (2)
    Compute action $\mathbf{a}_t = \dot{\mathbf{x}}\mathrm{d}t$
    Roll out $\mathbf{a}_t$ and record $\{(\mathbf{o}_t, \mathbf{a}_t)\}$
    $\mathcal{D}_{\text{aug}} \leftarrow \mathcal{D}_{\text{aug}} \cup \{(\mathbf{o}_t, \mathbf{a}_t)\}$
  end for
end foreach
Train imitation policy $\pi_\theta(\mathbf{o})$ on $\mathcal{D}_{\text{aug}}$

C. Data augmentation pipeline

Our overall pipeline is summarized in Algorithm 1. We first generate demonstrations in simulation using a motion-planning solver, where the action is defined as end-effector delta motion in global frame. We then sample perturbed initial states uniformly within the reachable area of the robot and around each demonstration trajectory, and generate corrective actions from the constructed velocity field. By rolling out these actions in simulation, we collect augmented observation–action pairs and add them to the training set. Figure 1 visualizes the new corrective trajectories generated by the velocity field for a cube-lifting task.

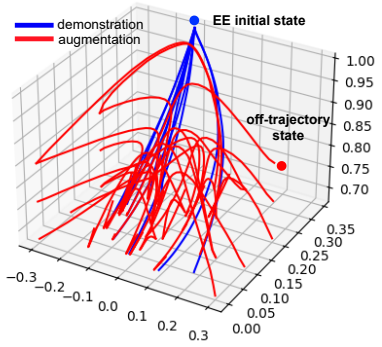

Fig. 1. End-effector trajectory demonstrations (blue) for the cube-lifting task shown in figure 2, and the velocity-field rollouts used for data augmentation (red). The red trajectories are generated by initializing from perturbed states around the demonstrations and rolling out a corrective field that pulls the state back toward the nearest point on the reference trajectory while progressing along it.

III. Preliminary Results

A. Experiment Setup

We evaluate the augmented dataset on a simple grasp task in simulation. The robot is required to approach an object, close the gripper, and lift the object above a target height. We generate 200 demonstrations with the motion-planning solver, then 500 new trajectories are collect using velocity field based augmentation. We also compared the augmented dataset with a dataset in which the initial pose of the end-effector is randomized and contains 700 trajectories. We use multi-layer-perceptron (MLP) and diffusion policy (DP) [15] to evaluate the data. During inference, we investigate if the augmented dataset can improve robustness by perturbing the object at different simulation steps t (the object is randomly set to a new pose within a fixed area).

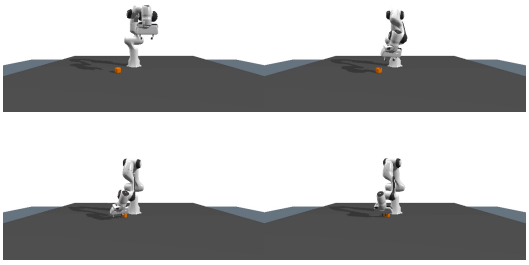

Fig. 2. Rollout of the cube-lifting task under object-pose perturbations. We reinitialize the object pose at a specified time step during execution. The sequence illustrates how late-stage resets lead to deviations from the nominal behavior and more challenging recovery behaviors.

TABLE I
SUCCESS RATE (%) UNDER DIFFERENT PERTURBATION TIMING.

step	MLP			DP [15]		
	w/o aug	w/ aug	rnd	w/o aug	w/ aug	rnd
0	1.0	1.0	1.0	1.0	1.0	1.0
30	0.15	0.6	0.6	0.8	0.95	0.95
70	0.0	0.35	0.3	0.3	0.6	0.9

B. Results

We evaluate robustness to environment-induced distribution shift by reinitializing the object pose at different simulation steps during policy rollouts. As t increases, the success rate decreases, since later resets more strongly disrupt the ongoing manipulation phase (e.g., pre-grasp alignment/contact), driving the system into states that are poorly covered by the motion-planned demonstrations. Table I reports robustness under perturbations for policies trained with and without data augmentation. Dataset augmentation mitigates this effect by providing additional off-trajectory state–action pairs, resulting in higher success rates after object-pose resets.

C. Limitations

This extended abstract focuses on a single cube-lifting task as a controlled testbed to study the effect of velocity-field-based augmentation. Augmentation can improve the robustness but a dataset with randomized data can also achieve similar performance in this simple test environment. Therefore extending to long-horizon and contact-rich manipulation tasks is an important direction, where compounding errors, mode switches, and multi-stage objectives may require different augmentation strategies. Another direction is to investigate more efficient strategies for sampling perturbed states for data augmentation.

References

[1] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[2] S. Belkhale, Y. Cui, and D. Sadigh, "Data quality in imitation learning," *Advances in neural information processing systems*, vol. 36, pp. 80 375–80 395, 2023.

[3] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg, "Dart: Noise injection for robust imitation learning," in *Conference on robot learning*. PMLR, 2017, pp. 143–156.

[4] M. Jia, D. Wang, G. Su, D. Klee, X. Zhu, R. Walters, and R. Platt, "Seil: Simulation-augmented equivariant imitation learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1845–1851.

[5] L. Ke, Y. Zhang, A. Deshpande, S. Srinivasa, and A. Gupta, "Ccil: Continuity-based data augmentation for corrective imitation learning," *arXiv preprint arXiv:2310.12972*, 2023.

[6] L. Ke, J. Wang, T. Bhattacharjee, B. Boots, and S. Srinivasa, "Grasping with chopsticks: Combating covariate shift in model-free imitation learning for fine manipulation," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6185–6191.

[7] J. Z. Kolter and G. Manek, "Learning stable deep dynamics models," *Advances in neural information processing systems*, vol. 32, 2019.

[8] J. Urain, M. Ginesi, D. Tateo, and J. Peters, "Imitationflow: Learning deep stable stochastic dynamic systems by normalizing flows," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5231–5237.

[9] A. Sochopoulos, M. Gienger, and S. Vijayakumar, "Learning deep dynamical systems using stable neural odes," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 163–11 170.

[10] H. Beik-Mohammadi, S. Hauberg, G. Arvanitidis, G. Neumann, and L. Rozo, "Extended neural contractive dynamical systems: On multiple tasks and riemannian safety regions," *The International Journal of Robotics Research*, p. 02783649251366326, 2025.

[11] Y. Li and S. Calinon, "From movement primitives to distance fields to dynamical systems," *IEEE Robotics and Automation Letters*, vol. 10, no. 9, pp. 9550–9556, 2025.

[12] M. Toussaint, "A tutorial on newton methods for constrained trajectory optimization and relations to slam, gaussian process smoothing, optimal control, and probabilistic inference," *Geometric and numerical foundations of movements*, pp. 361–392, 2017.

[13] Y. Wang, Z. Wang, M. Nakura, P. Bhowal, C.-L. Kuo, Y.-T. Chen, Z. Erickson, and D. Held, "Articubot: Learning universal articulated object manipulation policy via large scale simulation," *arXiv preprint arXiv:2503.03045*, 2025.

[14] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T.-k. Chan, *et al.*, "Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai," *arXiv preprint arXiv:2410.00425*, 2024.

[15] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.