One Demo is Worth a Thousand Trajectories: Action-View Augmentation for Visuomotor Policies

Chuer Pan[1], **Litian Liang**[1], **Dominik Bauer**[2]

Eric Cousineau[3], **Benjamin Burchfiel**[3], **Siyuan Feng**[3], **Shuran Song**[1]
[1]Stanford University, [2] Columbia University, [3]Toyota Research Insititute
https://chuerpan.com/1001-demos.github.io/

Abstract: Visuomotor policies for manipulation have demonstrated remarkable potential in modeling complex robotic behaviors, yet minor alterations in the robot's initial configuration and unseen obstacles easily lead to out-of-distribution observations. Without extensive data collection effort, these result in catastrophic execution failures. In this work, we introduce an effective data augmentation framework that generates visually realistic fisheye image sequences and corresponding physically feasible action trajectories from real-world eye-in-hand demonstrations, captured with a portable parallel gripper with a single fisheye camera. We introduce a novel Gaussian Splatting formulation, adapted to wide FoV fisheye cameras, to reconstruct and edit the 3D scene with unseen objects. We utilize trajectory optimization to generate smooth, collision-free, view-rendering-friendly action trajectories and render visual observations from corresponding novel views. Comprehensive experiments in simulation and the real world show that our augmentation framework improves the success rate for various manipulation tasks in both the same scene and the augmented scene with obstacles requiring collision avoidance.

1 Introduction

Visuomotor policies trained through imitation learning [1, 2, 3] enable complex robot behaviors but often remain brittle: minor changes in the robot's initial configuration or the objects in the scene may yield out-of-distribution (OOD) observations, cascading into OOD states, and resulting in compounded execution errors that cause task failures, hindering robot performance [4, 5, 6]. To improve policies' **spatial robustness**, human demonstrators have to repeatedly demonstrate the same skill on identical objects numerous times under different spatial configurations [7]. While effective, this manual process is tedious and costly. We address this by introducing an effective data augmentation framework that improves the spatial robustness of visuomotor policies by automatically generating

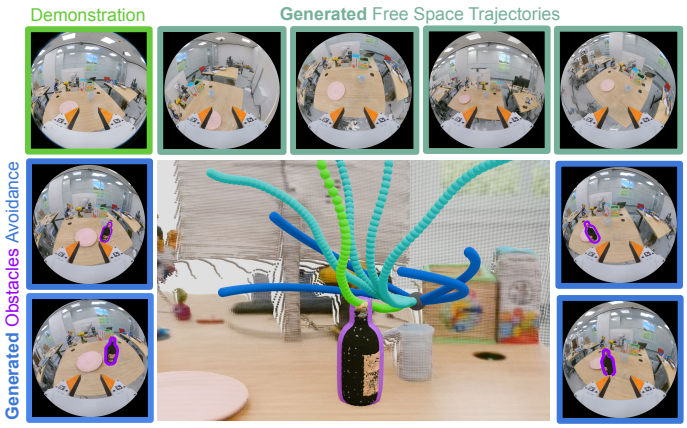


Fig. 1: **1001 DEMOS.** From a single human demonstration (e.g., picking up the blue mug), our approach generates valid training trajectories with large spatial variance and augmented obstacles, while respecting action-view consistency, 3D collision and contact dynamics constraints.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

additional real-world robot trajectory data from existing human demonstrations, thereby expanding data spatial coverage, without exhaustive manual collection.

While data augmentation is a standard procedure in other domains, such as computer vision [8], augmenting real-world robot manipulation data presents a set of unique challenges:

- **Maintaining Action-View Consistency.** Robot policy learning requires paired observation and action as data. Hence, the data argumentation algorithm need to increase both visual and action diversity and critically maintain the consistency between these two.
- **Respecting Physical Constraints of Actions.** The augmented action data needs to obey physical constraints, including both 3D collision constraints and object contact dynamics.
- **Maximizing Visual Coverage from Limited Demonstrations.** Real-world robot demonstration data is limited in terms of view coverage. Therefore, to make effective data argumentation possible, we need to make every collected trajectory demo count by maximizing their visual coverage from the same number of views during data collection.

To address these challenges, we propose **1001 DEMOS**; a data augmentation technique featuring the following key designs:

- To generate spatially consistent observation-action pairs, we propose a *trajectory-level action–view augmentation* algorithm that first reconstructs the scene, then generates physically feasible, collision-free action trajectories via trajectory optimization, to finally render the spatially matching observations via (editable) 3D Gaussian Splatting (3DGS); yielding spatially consistent, visually realistic, and physically feasible observation–action demonstration trajectory episodes.
- To obey the 3D collision constraints in 3DGS scenes with edited obstacles, we propose a *collision-aware* action generation module that uses trajectory optimization to create smooth, collision-free, and diverse action trajectories beyond the demonstrated action distribution, allowing the resulting visuomotor policies to learn collision-avoidance behaviors. To obey object contact dynamics, we propose a *contact-aware* augmentation for automatic contact event detection, which only perform augmentation before contact events, preserving the contact dynamics in the original demo.
- To maximize visual coverage during data collection, our system uses an ultra-wide fish-eye camera. However, while this drastically increases the field-of-view of the observations, this non-standard camera configuration requires us to extend the 3D Gaussian Splatting formulation by introducing a *fisheye ray sampler* in the rendering step.

Our experiments in simulation and the real world validate the effectiveness of our action-view data augmentation approach. We show that the proposed free-space data augmentation improves the manipulation policies' performance in simulation on the RoboMimic [9] benchmark. Moreover, the proposed collision-aware data augmentation improves real-world manipulation policies robustness against unseen obstacles in pick-and-place and non-prehensile tasks, leading to an improved success rate when compared to policies that are trained without **1001 DEMOS**.

2 Related Work

Image Augmentation for Visual Invariance. Robustness to visual variation – appearance, illumination, viewpoint – has been extensively studied [8]. Common augmentations include color jittering [1], image filtering [10], and cropping [11, 12]. Image editing using generative models further enables object-level modifications [13, 14, 15], embodiment swapping [16], and viewpoint interpolation with embodiment transfer [17]. Yet, such approaches are not able to augment the desired robot trajectory accordingly, and are thus limited to global appearance, background object, or minor viewpoint changes. By contrast, our method synthesizes visually realistic, multi-view observations via 3DGS and produces physically feasible action trajectories through trajectory optimization.

State-based Data Augmentation. State-based augmentation disentangles raw visual inputs from policy observations by varying scene configurations and adjusting trajectories. For example, Flo-

rence et al. [18] inject noise into keypoint-based state representations to mitigate cascading errors. Learned-dynamics methods with continuity constraints guide policies back to expert states [19, 20]. Simulation environments further ensure the validity of such augmentations [21]. However, these methods defer visual invariance to state estimation and rely on a dynamics model – simulated or learned – to ensure physical consistency. Instead, our approach jointly augments visual inputs and action trajectories in a realistic manner, producing diverse, obstacle-avoiding demonstrations that yield policies robust to out-of-distribution viewpoints and capable of obstacle avoidance.

Visual-Action Augmentation. Augmenting visual observations *and* actions enables joint variation of scene configurations and robot behaviors. Prior work generates third-person pinhole augmentations, either in simulation [22, 23], via novel-view synthesis in the real world [24, 25], or adapt egocentric pinhole observations via NeRF/3DGS and MPC [26, 27]. Zhou et al. [28] (using NeRF) and Zhang et al. [29] (using diffusion) focus on single-step pinhole image-action pairs but can neither handle wide FoV nor produce trajectory-level, obstacle-avoiding data. MimicGen [22] and follow-ups [30, 23, 31] augment third-person demonstrations but rely on costly on-robot rollouts to obtain in-domain visuals for real-world deployment. Concurrent work [32, 33] enable trajectory-level visual-action augmentation, but both are restricted to demonstrations from *static, third-person, pinhole* cameras. Yang et al. use 3DGS to edit robot, object, and background appearances but does not allow obstacle avoidance augmentation; Xue et al. focus on visuomotor policies with point cloud inputs, using point cloud editing to produce obstacle-aware augmentations. In contrast, we integrate a fisheye ray sampler into 3DGS to elegantly handle *fisheye* images – preserving 3DGS' speed and editability while enabling *trajectory-level*, *obstacle-avoiding* for *eye-in-hand* observation-action demonstrations; creating robust visuomotor policies across diverse camera viewpoints and obstacle avoidance behaviours.

3 The 1001 Demos Framework

We introduce **1001 DEMOS**, an offline data augmentation framework for visuomotor policies. From a single real-worldtask demonstration, using a portable manipulation-data collection device equipped with a fisheye camera, our augmentation technique generates **1001 DEMOS** of *visually realistic* image sequences for *physically feasible* action trajectories.

Fig. 2: **1001 DEMOS Overview.** From an initial mapping run, we reconstruct the 3D scene point cloud for easy trajectory planning and a fisheye 3DGS scene for fast novel view rendering (§3.1). Given a single demonstration video (green), we optimize additional physically feasible action trajectories (§3.2) and render the corresponding visually consistent fisheye-image observations (§3.3), thereby generating thousands of diverse action-view demonstrations from a single real-world demonstration.

As illustrated in Fig. 2, given a fisheye video pair – from scene scanning and task demonstration, **1001 DEMOS** is able to generate (1) demonstrations with vastly different initial configurations and (2) collision-avoiding demonstrations with obstacles added through scene editing. **1001 DEMOS** achieves this via three modules. First, we use fisheye image sequences to reconstruct a 3D scene point cloud for motion planning and a *fisheye* 3D Gaussians representation of the scene for

novel view rendering (§3.1). Second, given the extracted scene point cloud and the original demonstration trajectory, we generate smooth and collision-free trajectories in the same scene, starting from different initial camera poses (§3.2) and render the corresponding novel observations (§3.3). Finally, with the generated free-space and obstacle-avoiding demonstrations, we train visuomotor policies on the original expert-collected demonstration and the generated demonstrations. (§3.4), enabling downstream robot policies that gracefully handle unseen initial configurations and avoid novel obstacles in the scene.

3.1 3D Scene Reconstruction from Eye-in-Hand Fisheye Video

We use UMI [7] and its demonstration dataset format for hand-held data collection, and Diffusion Policy [1] for policy learning. Specifically, we assume that each dataset $\mathcal{D} = \{d\}_N$ consists of N data episodes. Each episode $d = (o, a)$ is composed of a sequence of visual observations $o \triangleq \{o_{fish}\}$ as eye-in-hand fisheye RGB images, o_{fish}, and a sequence of action, $a \triangleq \{a_{ee}, a_{gp}\}$. Where each action a is composed of a 6D end-effector pose, $a_{ee} \in \mathbf{SE}(3)$, and a gripper-opening width, $a_{gp} \in \mathbb{R}$. Given a video pair, collected during scene scanning and task demonstration, we use these fisheye image sequences for 3D reconstruction to produce a 3D scene point cloud for motion planning and a Fisheye 3D Gaussian representation of the scene for novel view rendering.

Scene Point Cloud Reconstruction & Contact Detection. We leverage COLMAP [34] to reconstruct high-fidelity 3D point clouds from the fisheye image sequences captured during scene scanning. We then split each demonstration into pre-contact and post-contact phases by finding the first frame where the gripper exceeds a collision threshold with the reconstructed point cloud.

Fisheye 3D Gaussians. A critical design choice enabling fast rasterization, 3DGS [35] tiles pinhole image into 16×16 pixel patches and uses 256-thread cuda blocks per tile, one thread per pinhole ray. To elegantly handle fisheye images, we replace the original ray sampler with a KB8-based [36] fisheye ray sampler. As shown in Fig.3, for each fisheye pixel (u, v), compute its ray direction $r_d = \mathrm{KB8}(u, v)$ [36], then project r_d through the camera intrinsics K to obtain the pinhole coordinates $(u_p, v_p) = K\, r_d$, thus associating each fisheye ray with its 3D-Gaussian splat location on the 2D image plane. We redistribute tile assignments from pinhole to fisheye rays, partition fisheye rays into the original 256-ray-per-tile layout to preserve the CUDA block–thread structure for fast rasterization while accurately modeling fisheye distortion. We optimize the fisheye 3DGS with pixel-wise losses from [35] between rendered and ground-truth fisheye images.

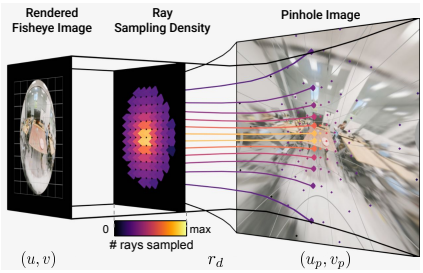


Fig. 3: **Fisheye 3DGS.** We propose Fisheye-3DGS, using a ray sampler that accounts for fisheye distortion. Sampling density adapts to pixel location, allocating more rays to the image center than the periphery for better rasterization quality.

3.2 Action Generation via Trajectory Optimization

Utilizing the extracted 3D scene point cloud from Sec 3.1 and the input demonstration trajectory, we employ trajectory optimization to generate two types of novel trajectories: (1) *free space* demonstration trajectories in the same scene with different starting poses, sampled randomly in the free space of the scene; (2) *obstacle-avoiding* demonstration trajectories that are planned around the added obstacle point cloud in the scene.

Trajectory Optimization Formulation. Given a sequence of 6D camera poses, $O_{\mathrm{ee}} \triangleq \{o_{\mathrm{ee}}^m\}_{m=1}^H \subset \mathbf{SE}(3)$ – extracted from the pair of scene scanning and task demonstration –, the 3D point cloud of the task scene, $P_{\mathrm{scene}} \in \mathbb{R}^{N_{\mathrm{scene}}\times3}$, and the start and end poses specified as $x_{\mathrm{init}} \in \mathbf{SE}(3)$ and $x_{\mathrm{goal}} \in \mathbf{SE}(3)$ (chosen as the pre-contact pose here), respectively. Provided with a trajectory initialization, $X \triangleq \{x^k\}_{k=1}^T \subset \mathbf{SE}(3)$, we consider the following trajectory optimization problem,

$$\underset{\{x_k\}_{k=1}^{T}}{\text{argmin}} \quad \mathcal{L}_{\text{funnel}}(X, O_{\text{ee}})) + \mathcal{L}_{\text{collision}}(X, \text{tsdf}(P_{\text{scene}})) + \mathcal{L}_{\text{render}}(X, O_{\text{ee}}) + \mathcal{L}_{\text{smooth}}(X),$$
$$\text{subject to} \quad x^1 = x_{\text{init}}, x^T = x_{\text{goal}}, X \cap \text{convhull}(P_{\text{obstacle}}) = \emptyset,$$

where $P_{\text{obs}} \in \mathbb{R}^{N_{\text{obs}} \times 3}$ is the point cloud of the augmented obstacle added for obstacle-avoiding trajectory generation (Sec 3.2). $\text{convhull}(\cdot)$ retrieves the convex hull from a point cloud, $\text{tsdf}(\cdot) : \mathbb{R}^3 \to \mathbb{R}$ is a truncated signed distance function (TSDF) which maps a 3D coordinate to a scalar distance. x_{init} is sampled within a θ quaternion cone around each original viewpoint, with r as the radius of the quaternion sphere.

Free-Space Trajectory Generation.

We preserve the original contact dynamics using a delta funnel loss $\mathcal{L}_{\text{funnel}}$ to produce trajectories that converge consistently to the same pre-contact pose of the original demonstration. Let R and t represent the rotation and translation components of $x \in SE(3)$, we have $\mathcal{L}_{\text{funnel}} = \sum_k w_k \left\| t^k - t^k_{\text{ee}} \right\|_2^2$, where w_k is a temporally dependent weight defined as, $w_k = w_{\text{min}} + (w_{\text{max}} - w_{\text{min}}) \cdot \left(\frac{k}{T}\right)^3, \sum_k w_k = 1, \; 0 \leq w_{\{\text{max,min}\}} \leq 1$. Novel view rendering from 3DGS suffers from floater artifacts and blurry scene reconstruction when the rendering viewpoint differs too much from the training viewpoint distribution. To free the generated data from these rendering artificats, which may have negative impact on downstream policy training, we introduce $\mathcal{L}_{\text{render}}$ to optimize each generated pose x^k to be close to the 6D pose distribution of the original demonstration and scanning views within a ball neighborhood $\mathcal{N}_k$, $\mathcal{L}_{\text{render}} = \sum_k \sum_{j \in \mathcal{N}_k} \left\| \log\left((R^k)^\top R^j_{\text{ee}}\right) \right\|^2 + \left\| t^k - t^j_{\text{ee}} \right\|_2^2$. To ensure the generated trajectories are free of collisions with the environment, we incorporate a collision-loss $\mathcal{L}_{\text{collision}} = -\sum_k \text{tsdf}(x^k)$.

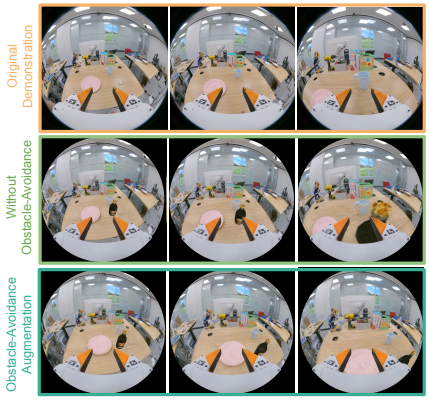


Fig. 4: **Augmentation with Obstacle Avoidance.** Top: original demo; Middle: augmented trajectories without obstacle avoidance; Bottom: augmented trajectories with obstacle avoidance.

In addition, to ensure smoothness of the generated trajectory, we introduce $\mathcal{L}_{\text{smooth}}$ to penalize velocity jerkiness. For free-space augmentation, the trajectory initialization is produced by linear interpolation between the newly sampled x_{init} and the pre-contact demonstration pose x_{goal}.

Obstacle Augmentation and Collision-Free Trajectory Generation. As shown in Fig. 4, given an obstacle – a point cloud sampled from an Objaverse [37] object, we compute convex hull of the object to update the scene TSDF. For obstacle-avoiding augmentation, an initial-guess trajectory is produced using RRT* [38], sampling a trajectory that connects the new x_{init} and the pre-contact demo pose x_{goal}, already avoiding the added obstacle in the scene. Then, we use the same trajectory optimization formulation above, with the added collision constraint, $X \cap \text{convhull}(P_{\text{obstacle}}) = \emptyset$.

3.3 View Generation via Fisheye 3D Gaussian Splatting

We generate the visual observations that correspond to the augmented (action) trajectories using *Fisheye Gaussian Splatting*, both in free space and in scenes with added 3D obstacles. We optimize the 6D trajectories from (Sec. 3.2) to match the input video's viewpoint distribution, yielding collision-free paths that maintain high-fidelity Gaussian Splat renderings by keeping views within the distributions of the training viewpoints for 3DGS.

Free-Space View Generation. We use the generated free-space trajectories that start from different initial scene observation directions to render from the trained Fisheye 3DGS of the scene to generate corresponding visual fisheye image observations.

Obstacle-Scene View Generation. We augment the original Fisheye 3DGS of the scene with a trained Fisheye 3DGS of obstacles obtained from Objverse to generate unseen scene configurations. Then, we use the generated obstacle-avoiding trajectories starting from different initial scene observation directions to render from the trained Fisheye 3DGS of the scene to generate corresponding obstacle-avoiding visual fisheye image observations.

3.4 Action-View Augmentation for Visuomotor Policies

We train a Diffusion Policy [1] on the union of the original and augmented datasets, $\mathcal{D} \cup \tilde{\mathcal{D}}$, using a CLIP-pretrained ViT-B/16 encoder [39, 40] with a relative action representation.

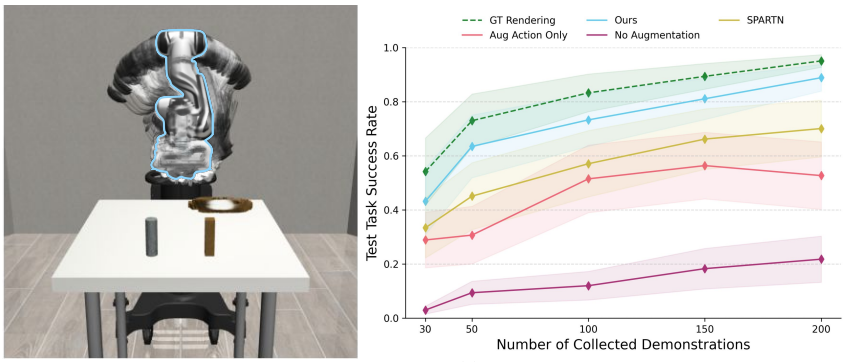
Action-View Data Compilation. We collect these original demonstrations $\mathcal{D}$ on a modified UMI [7] with an iPhone. ARKit VIO provides metric 6D end-effector poses a_{ee}, removing the need for an extra AprilTag SLAM mapping round. The gripper-opening width a_{gp} is measured via fiducial markers. For augmented data $\tilde{\mathcal{D}}$, each fisheye observation o_{fish} is segmented by SAM2 [41] and we overlay the gripper onto rendered images, yielding $\tilde{o}_{\text{fish}}$. We assign trajectory-optimized 6D poses (Sec. 3.2) as $\tilde{a}_{ee}$ and retain the original gripper-opening width as $\tilde{a}_{gp}$.

4 Experiments

Our experiments in real-world and simulated environments aim to answer the following key questions: 1) Does action-view augmentation help imbue policies with improved robustness against unseen initial configurations and obstacles(§4.1)? 2) How should augmentation be performed (§4.2)? and 3) How much augmentation is beneficial (§4.3).

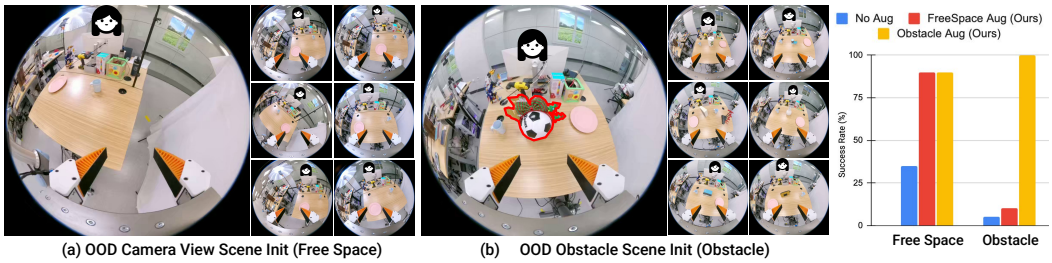
Simulation Evaluation. For evaluation in simulation, we use the *RoboMimic* [9] "square" task to evaluate the performance gain afforded by **1001 DEMOS** in free-space augmentation, compared to no augmentation, augmentation with ground truth novel-view rendering (obtainable in MuJoCo [42]), and representative baselines – `Aug Action Only` [18], and `SPARTN` [28](§4.2). We used the 200 expert-collected task demonstrations provided in RoboMimic as the base dataset. The RoboMimic "square" task is a peg-in-hole task that requires a Franka Emika Panda robot to pick a square nut and insert it onto a rod. To follow the same data format as in §3.1, we convert the pinhole image observations from the wrist camera into Fisheye images using intrinsic & distortion parameters of a GoPro Fisheye lens with a 155° FoV. To study the effects of different methods of augmentation, we keep the augmentation scheme constant – for each base demonstration episode, we generate 20 free-space augmentation episodes. The initial camera pose is sampled within a θ quaternion cone around the initial camera pose of each base demonstration, with 0.15m as the scaled radius of the quaternion sphere, and the initial camera viewing direction defined as the zero quaternion. For quantitative comparison, shown in Fig. 5(b), we found $\theta = 50°$ to strike a balance between novel-view rendering quality and policy performance, as detailed in §4.3. All the compared methods are used to train a Diffusion Policy [1] on subsets of $\{30, 50, 100, 150, 200\}$ of the expert dataset, respectively, and tested on a fixed set of 1000 initial robot configurations. These are sampled with the end-effector poses within a quaternion cone of $50°$ and 0.15m radius with respect to the base RoboMimic dataset, while keeping the test object configurations unaltered. The distributions of training and testing initial states are shown in Fig.5(a), with the resulting policy success rates reported in Fig.5(b).

Real-world Evaluation. In our real-world evaluation, we aim to determine the effectiveness of **1001 DEMOS** for enabling visuomotor policies to handle out-of-distribution (OOD) scenarios with respect to our UMI-collected training data distribution on the following two axes: (a) OOD robot and object initial states; (b) unseen obstacles in the scene. We report evaluation results on a cup serving task. This task requires the robot to pick a cup with its handle to the left and place it on the pink serving plate, as shown in Fig. 6. We define the task as successfully completed when the cup is placed upright on the serving plate with its handle within $\pm 10°$ towards the left of the table. The base dataset includes 89 demonstration episodes, collected by a single demonstrator using UMI [7].



(a) Init State Train vs Test Distribution (b) Simulation Quantitative Comparison

Fig. 5: **Simulation Evaluation.** (a) Initial state distribution for training data highlighted in blue overlay over custom test data. (b) Task success rate with action-view augmentation, compared to no augmentation, oracle action-view augmentation and other augmentation baselines.



(a) OOD Camera View Scene Init (Free Space) (b) OOD Obstacle Scene Init (Obstacle)

Fig. 6: **Real-world Evaluation.** We report task performance for two versions of our augmented policies – trained with free-space augmentation (*FreeSpace Aug*), and free-space & obstacle-distractor augmentation (*Obstacle Aug*) – against a vanilla policy trained with no augmentation (*No Aug*). Initial states for a subset of all evaluation episodes for (a) OOD camera view test case, (b) OOD obstacle distractor test case shown. (c) Success rates, averaged over 20 evaluation episodes.

All demonstrations were collected in obstacle-free scenes, with all initial camera views in an upright overhead orientation (Fig. 1).

We manually select 50 episodes that produce the best novel-view rendering results for both free-space and obstacle augmentation. From these, we generate 7245 free-space augmented episodes by sampling initial camera orientations within a 45° quaternion cone around each original viewpoint, with the Euclidean distance between the initial demonstration start pose and the pre-contact pose as the radius of the quaternion sphere. For obstacle augmentation, we select 50 objects from Objaverse [37], render 256 orbit views for each to train a fisheye-3DGS per obstacle. Then, by integrating each obstacle into the original demonstrations' 3DGS scenes, we generate 5000 obstacle-aware episodes in total. We train Diffusion Policy [1], following UMI's policy interface protocols, to obtain a `No Aug` policy with the original human-collected data, a `FreeSpace Aug` policy with the original data and additionally generated free-space augmentation data, and an `Obstacle Aug` policy with the original data and additionally generated obstacle augmentation data. Policy success rates are averaged over 20 evaluation episodes for two test scenarios: (a) OOD camera-view and (b) OOD obstacle initialization, are shown in Fig. 6, with a subset of the init distributions shown. We further evaluated the three policies in a harder obstacle setup with more challenging placements and larger, more complex shapes, as shown in A.2.

4.1 Does Action-View Augmentation help?

Action-view augmentation helps in improving sample efficiency. To investigate how much improvement our augmentation scheme provides compared to no augmentation, we evaluate visuomotor policies trained solely on the converted fisheye RoboMimic expert dataset, totaling 200 episodes, without any augmentation. Conversely, to establish an upper bound for our action-view augmentation scheme if it would provide perfect novel-view synthesis, we train a policy using the union of fisheye RoboMimic expert demonstrations and free-space augmentation demonstrations gener-

ated using **1001 DEMOS** with ground-truth rendering (which is only possible in simulation). This produces an oracle policy to provide an upper bound for our action-view augmentation scheme. Figure 5(b) shows that, with the same amount of original expert demonstrations, our action-view augmentation closely tracks the perfect GT-rendering upper bound, with a performance gap of 8% in the low-data regime and 11% in the high-data regime. Compared to policies trained without augmentation, our free space augmentation provides an average of 56% task success rate improvement. In the real-world experiment shown in Fig 6, we find that, with the same amount of human demonstrations, our free space augmentation provides a performance boost of 55% over policies trained without augmentation.

Action-view augmentation helps in obstacle avoidance. In the real-world experiment, shown in Fig. 6, we find that, by augmenting obstacle-free demos with our generated obstacle-avoiding demos, our action-view augmentation equips the visuomotor policy to robustly complete the task while avoiding obstacles – behaviors not shown in the original human demonstrations. We find that our full `Obstalce Aug` is able to complete the task with a 100% success rate, significantly outperforming `FreeSpace Aug` with 10% and `No Aug` with 5% success rates, respectively.

4.2 How to Augment?

Comparison to action-only augmentation. One simple and effective augmentation for local feedback stabilization is `Aug Action Only` [18], which slightly perturbs proprioceptive and gripper action data while leaving visual data unchanged. While effective for small, local robot state variations under third-person views, it breaks down with eye-in-hand observations, where minor end-effector pose changes produce drastic visual shifts. We replicate this baseline by applying free-space augmentation in the end-effector and gripper actions, but using the original visual observations. As Fig. 5 shows, `Aug Action Only` boosts the average success rate by 29% over `No Aug` – peaking at 56%. In turn, `Ours` outperforms `Aug Action Only` by up to 35%, especially in high-data regimes.

Comparison to single-step augmentation. In this experiment, we compare our proposed method, which augments both visual and action data for the whole trajectory, with prior work SPARTN [28], which augments visual and action data for a single step and uses NeRF to reconstruct the visual scene. Single-step action-view augmentation using `SPARTN` improves policies' performance for OOD camera views, as shown in Fig. 5 with a performance gain of 41% over `No Aug`. However, SPARTN's single-step augmentation cannot produce smooth, trajectory-level collision-avoidance behaviors, more easily causing policies to enter unseen configurations for which no recovery behaviors exist in the training data. By comparison, **1001 DEMOS** performs trajectory-level action-view augmentation. We observe that `Ours` thereby is able to outperform `SPARTN`, boosting the average success rate by 15% – peaking at an improvement of 18%. more easily end up in unseen configurations for which no recovery behavior is present in the training data.

4.3 How Much to Augment?

While larger rotation bounds increase diversity, they can harm rendering quality under limited viewpoint coverage. To quantify this trade-off, we trained visuomotor policies on 50 RoboMimic "square" export demonstrations (fisheye, eye-in-hand), augmenting each with rotation bounds of $\{20°, 30°, 40°, 50°, 60°\}$, generating 20 augmented episodessamples per demo. We then evaluated on a held-outout test set shown in Fig 5. As shown in Fig A1, success rates plateaued at $50°$, which we therefore adopt for all other experiments.

5 Conclusion

We present **1001 DEMOS**, an offline data-augmentation framework for visuomotor policies that effectively endows robots with skills not demonstrated in original demo – obstacle avoidance and robustness to novel initial robot configurations, by generating visually realistic and physically feasible *trajectory-level*, *obstacle-avoiding*, *eye-in-hand*, *fisheye* action-view demonstrations.

Acknowledgments

This work was supported in part by the Toyota Research Institute, NSF Award #2143601, #2037101, and #2132519. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

6 Limitations & Future Work

With only a single eye-in-hand moving camera, the view coverage of the demonstration stage is inadequate for dynamic scene reconstruction or for generating novel views far from the original viewpoints. As a result, we currently restrict the **1001 DEMOS** pipeline to static scenes before or after contact. Future work could explore more advanced sensing setup, such as multi-camera rigs or ToF sensors, or adopt advanced dynamic reconstruction methods that demand fewer training viewpoints [43, 44].

Our novel-view generation module inherits 3DGS's multi-view inconsistent nature, yielding floating artifacts for generated viewpoints outside of the training viewpoint distribution. This could be alleviated by adopting inherently view-consistent representations like 2DGS [45].

Similar to UMI [7], since the kinematic limits of the downstream deployment robots are unknown at the time of data collection, the generated demonstration trajectories do not account for kinematic limits of the downstream deployment robots. We carefully bound both, the sampling range for initial poses and the placement of obstacles, to ensure that the generated trajectories lie within the task space of the deployment robot. Our work could be extended to incorporate the downstream robot kinematics constraint in the trajectory optimization module, ensuring embodiment-aware trajectory generation with respect to the specific deployment robot and enable kinematically feasible and smooth action trajectory, not just in the task space as addressed by this work, but also in the configuration space. Furthermore, this could allow retrofitting the original embodiment-agnostic demonstrations to be kinematically feasible for the downstream robot hardware deployment, thereby allowing an embodiment-aware policy learning framework that can transfer skills from semantically and physically valid but hardware-infeasible actions to different robot embodiments.

References

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[3] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[4] 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*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[5] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[6] S. Mirchandani, S. Belkhale, J. Hejna, E. Choi, M. S. Islam, and D. Sadigh. So you think you can scale up autonomous robot data collection? *arXiv preprint arXiv:2411.01813*, 2024.

[7] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[8] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[9] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[10] N. Hansen and X. Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13611–13617. IEEE, 2021.

[11] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

[12] D. Yarats, I. Kostrikov, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International conference on learning representations*, 2021.

[13] Z. Chen, Z. Mandi, H. Bharadhwaj, M. Sharma, S. Song, A. Gupta, and V. Kumar. Semantically controllable augmentations for generalizable robot learning. *The International Journal of Robotics Research*, page 02783649241273686, 2024.

[14] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

[15] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[16] L. Y. Chen, C. Xu, K. Dharmarajan, M. Z. Irshad, R. Cheng, K. Keutzer, M. Tomizuka, Q. Vuong, and K. Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. *arXiv preprint arXiv:2409.03403*, 2024.

[17] H. Chen, C. Zhu, Y. Li, and K. Driggs-Campbell. Tool-as-interface: Learning robot policies from human tool usage through imitation learning. *arXiv preprint arXiv:2504.04612*, 2025.

[18] P. Florence, L. Manuelli, and R. Tedrake. Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 5(2):492–499, 2019.

[19] 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.

[20] A. Deshpande, L. Ke, Q. Pfeifer, A. Gupta, and S. S. Srinivasa. Data efficient behavior cloning for fine manipulation via continuity-based corrective labels. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8531–8538. IEEE, 2024.

[21] P. Mitrano and D. Berenson. Data augmentation for manipulation. *arXiv preprint arXiv:2205.02886*, 2022.

[22] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

[23] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv preprint arXiv:2410.24185*, 2024.

[24] S. Yang, W. Yu, J. Zeng, J. Lv, K. Ren, C. Lu, D. Lin, and J. Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. *arXiv preprint arXiv:2504.13175*, 2025.

[25] S. Tian, B. Wulfe, K. Sargent, K. Liu, S. Zakharov, V. Guizilini, and J. Wu. View-invariant policy learning via zero-shot novel view synthesis. *arXiv preprint arXiv:2409.03685*, 2024.

[26] A. Tagliabue and J. P. How. Tube-nerf: Efficient imitation learning of visuomotor policies from mpc via tube-guided data augmentation and nerfs. *IEEE Robotics and Automation Letters*, 2024.

[27] J. Low, M. Adang, J. Yu, K. Nagami, and M. Schwager. Sous vide: Cooking visual drone navigation policies in a gaussian splatting vacuum. *IEEE Robotics and Automation Letters*, 2025.

[28] A. Zhou, M. J. Kim, L. Wang, P. Florence, and C. Finn. Nerf in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17907–17917, 2023.

[29] X. Zhang, M. Chang, P. Kumar, and S. Gupta. Diffusion meets dagger: Supercharging eye-in-hand imitation learning. *arXiv preprint arXiv:2402.17768*, 2024.

[30] R. Hoque, A. Mandlekar, C. Garrett, K. Goldberg, and D. Fox. Intervengen: Interventional data generation for robust and data-efficient robot imitation learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2840–2846. IEEE, 2024.

[31] C. Garrett, A. Mandlekar, B. Wen, and D. Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. *arXiv preprint arXiv:2410.18907*, 2024.

[32] S. Yang, W. Yu, J. Zeng, J. Lv, K. Ren, C. Lu, D. Lin, and J. Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. *RSS*, 2025.

[33] Z. Xue, S. Deng, Z. Chen, Y. Wang, Z. Yuan, and H. Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

[34] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[36] J. Kannala and S. S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

[37] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[38] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

[39] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[41] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[42] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[43] Z. Zhu, Z. Fan, Y. Jiang, and Z. Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European conference on computer vision*, pages 145–163. Springer, 2024.

[44] J. Chung, J. Oh, and K. M. Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 811–820, 2024.

[45] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.

A.1 How Much to Augment?

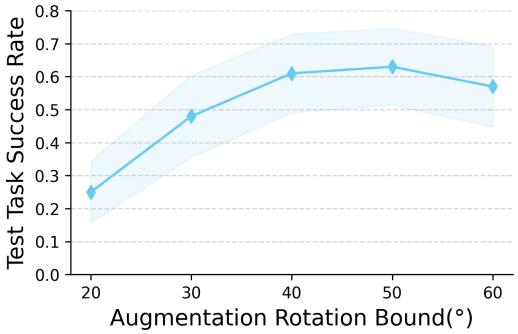


Fig. A1: **How much to augment?** While larger augmentation range could increase the data diversity, it also reduces the image rendering quality due to limited demonstration viewpoint coverage. We found 50° as the optimal trade-off.

A.2 Real World Evaluation on Challenging Obstacles.

To further examine the capability of our policies enabling obstacle avoidance enhancement, we additionally evaluated policy performance for the cup serving task on a set of more challenging scenarios with more challenging obstacle placement, and obstacles of larger size and more diverse geometric shape; we term this set of experiments *Challenging Obstacles*. As shown in Tab A1, we conducted 10 trials on 10 different obstacle sets as shown in Fig. A2, on the same three policies `No Aug`, `FreeSpace Aug`, `Obstacle Aug`, as tested in real world experiments for *Free Space* and *Obstacle* as reported in manuscript, and found that ours `Obstalce Aug` was able to complete 10/10 trials, while `No Aug` and `FreeSpace Aug` both fail complete any trials.

Method	Task Success Rate
`No Aug`	0%
`FreeSpace Aug` (Ours)	0%
`Obstacle Aug` (Ours)	**100%**

Table A1: **Real World Evaluation Results for *Challenging Obstalces***. Task success rate reported over 10 trials.

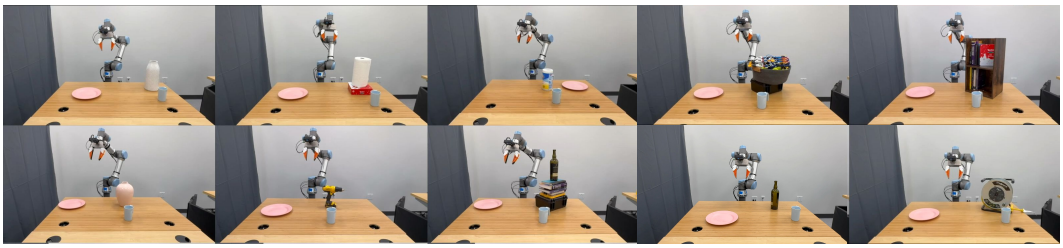


Fig. A2: **Challenging Obstacle Evaluation.** Initial states for all 10 evaluation episodes for *Challenging Obstacle* experiment. Please see the accompanying video for more comparisons.