Viewpoint-Invariant Manipulation via 3D Geometric Priors, Synthetic Data, and VLA Co-Distillation

Rishabh Jain
Department of Computer Science
Columbia University, NY
rj2790@columbia.edu

Rohit Ramesh
Department of Computer Science
Columbia University, NY
rr3713@columbia.edu

Tejal Bedmutha
Department of Computer Science
Columbia University, NY
tb3206@columbia.edu

***Abstract*—Imitation Learning from human demonstrations excels at encoding human dexterity into visuomotor policies, yet real-world deployment hinges on robustness to perturbations like camera repositioning or partial observability. While robot foundation models mitigate this via internet-scale pre-training, they remain computationally prohibitive for large-scale deployment. Our work investigates how low-cost imitation learning policies can be strengthened for a pick-and-place manipulation task through (i) 3D geometric priors (ii) algorithmic synthetic data generation and (iii) knowledge co-distillation with a foundation model. Empirical results demonstrate that while the baseline ACT policy collapses under camera configuration changes, the proposed 3DEgoACT formulation significantly mitigates performance degradation under unseen viewpoints with marginal increase in inference-time cost. Our findings highlight the central role of data quality, diversity and state representation choice in imitation learning and outline practical pathways for improving robustness without relying exclusively on large, expensive foundation models.**

I. Introduction

Learning robust visuomotor manipulation policies from limited demonstrations remains a central challenge in robotics. Imitation Learning methods such as Action Chunking Transformers (ACT) and Diffusion Policy [29, 7] have demonstrated strong performance in low-data regimes, but they often overfit to specific camera viewpoints and fail when visual conditions change[27]. Traditional 2D image-based policies, trained on fixed viewpoints, suffer from covariate shift during inference as subtle camera angle changes alter feature distributions. In real-world deployment, small perturbations in camera placement or occlusion can be detrimental to downstream performance due to the compounding error problem [29].

One promising direction to address this challenge is to incorporate stronger inductive biases into visuomotor policies. Traditional 2D image-based representations, while expressive, entangle semantic appearance with geometric structure and are therefore sensitive to viewpoint changes. In contrast, 3D geometric representations provide viewpoint-invariant spatial priors that can stabilize policy learning across diverse visual configurations. Large-scale robot foundation models, such as π_0 [3], partially alleviate these issues through internet-scale pretraining across heterogeneous data sources. However, such models remain computationally expensive and impractical for deployment on low-cost robotic platforms [16], motivating the need for alternative, data-efficient solutions.

In this work, we investigate how robustness in visuomotor imitation learning can be improved without relying on large foundation models. We focus on a pick-and-place manipulation task using a 7-DoF xArm manipulator and study three complementary mechanisms: (i) the incorporation of 3D geometric priors, (ii) algorithmic synthetic data generation via search-based planning, and (iii) knowledge distillation from VLAs.

We first established a strong ACT baseline trained on teleoperated demonstrations collected in simulation using a 7-DoF xArm manipulator with multi-view RGB inputs. Then, inspired by 3D Diffusion Policy (DP3)[27], we propose 3DEgoACT, which extends ACT by fusing point cloud representations with RGB egocentric views. MLP-encoded point clouds provide viewpoint-invariant geometric priors, while egocentric RGB view allows fine-grained manipulation without significant overhead. Evaluations over a pick-and-place task on randomized box-target configurations reveal negligible degradation under unseen camera shifts, contrasting ACT's collapse to 0%. Future augmentations via RRT-generated synthetic trajectories and VLA co-distillation (e.g., SmolVLA) target sub-10 human demonstration training.

In this work, the following goals for a pick-and-place task with a xArm-7 7-DOF manipulator are achieved:

1) Establish a strong ACT baseline trained on teleoperated demonstrations and analyze its sensitivity to camera configuration.
2) Fuse 3D geometric priors with 2D feature embed-

dings to obtain a view-point invariant policy
3) Generate synthetic data by using search-based trajectory generation based on human demonstrations for diversity
4) Implement foundation model co-distillation.

II. Related Work

A. Imitation Learning From Demonstrations

Imitation learning has long been a foundational approach for robotic manipulation due to its simplicity and effectiveness in learning policies directly from expert demonstrations. Behavioral cloning (BC) variants dominate IL, mapping pixels to actions via supervised losses [26]. The Action Chunking Transformer (ACT) [29], introduced with the ALOHA embodiment introduced 'action chunking' (predicting k-step sequences) to mitigate compounding errors, achieving 80-90% success on fine tasks like battery insertion from 50 demos. It uses a transformer backbone, taking ResNet encoded image robot proprioceptive features as input tokens for next predicting a chunk of k next actions. Notably, it follows a Conditional-VAE [22] setup while training to capture multi-modality in human demonstrations. Despite its effectiveness in low-data regimes, ACT remains highly sensitive to dataset quality, camera configuration, and visual redundancy, often failing to generalize across unseen viewpoints without careful tuning or additional supervision [7, 27].

B. Beyond 2D features: Geometric Representations

In general, standard behavior cloning methods are known to suffer from distributional shift when demonstrations inadequately cover the task state space, leading to brittle performance under viewpoint changes or novel configurations. Large-scale benchmarks such as ManiSkill3 highlight the importance of diverse demonstrations and simulation-driven data collection [18, 24].

3D-aware IL shifts toward geometric priors: PerAct [20] and RVT [12] use depth for keypoint prediction. DP3 [27] pioneers sparse point clouds in diffusion policies, outperforming 2D baselines by 24% on 72 sim tasks with 10 demos via MLP encoders for allocentric reasoning. ACT3D [11] proposes a 3D feature field transformer that lifts pretrained 2D features from multi-view RGB-D inputs into an adaptive-resolution 3D scene cloud, employing recurrent coarse-to-fine point sampling and relative cross-attention to efficiently regress 6-DoF keyposes (translation, rotation, gripper state, collision). While powerful for waypoint detection in multi-task settings, this discrete paradigm may limit temporal fluidity in dense, contact-rich control. In the proposed 3DEgoACT, we instead preserve ACT's simple transformer backbone and continuous action chunking, augmenting it with lightweight, single-view sparse point clouds to create low-cost, viewpoint-invariant policies.

Hybrid 2D+3D inputs disentangle semantic textures from geometric invariants, reducing viewpoint-induced covariate shift in imitation learning. Notable works include ROMAN [15], GraphCoT-VLA [13] and Gaussian World Models [17]. Unlike these complex hierarchies, our 3DEgoACT augments ACT with a single-view PointNet token for global geometry understanding plus EEF RGB tokens for local cues simultaneous viewpoint invariance and fine-grained manipulation.

C. Foundation Models

Vision-Language-Action (VLA) models, such as OpenVLA, π_0, represent a complementary direction toward generalization by leveraging large-scale multimodal pretraining. Recent works such as SmolVLA[21] demonstrate that compact VLAs can achieve competitive performance with reduced computational cost. Nonetheless, recent studies [30, 2] reveal that these models still face challenges in cross-task and cross-view generalization, particularly when precise manipulation is required. Geometry-aware and observation-centric VLA variants further emphasize the importance of grounding actions in spatial representations [1, 28, 6], but remain computationally expensive, leading to slow inference speeds - making them impractical for low-cost systems.

D. Synthetic Data Generation using planners

In alternative line of work, we leverage RRT-based planning explicitly as a mechanism for synthetic data generation rather than online control. While planners are widely used in classical robotics, their role as data generators for end-to-end visuomotor imitation learning remains underexplored, particularly in conjunction with transformer-based policies. By using RRT-style exploration to generate diverse, collision-free trajectories [8, 25], we augment limited teleoperated demonstrations with geometrically consistent and physically feasible motion data. This use of search-based planners as structured data generators complements human demonstrations and provides additional coverage of the state–action space for training end-to-end visuomotor policies.

III. Method

A. Simulation Environment and Task

The robotic embodiment utilized in this work is the xArm-7, a 7-DoF manipulator simulated within the MuJoCo physics engine. Environmental interaction is mediated via the Gymnasium library.

For the task, the manipulator grasps a box on the table and places it on a red mark as target (Figure 1). Both the box and target are randomly positioned.

B. Dataset Collection

Dataset quality is of crucial importance in imitation learning in a low data regime. Given data generated from human teleoperation is costly and time consuming, each teleoperated trajectory has to be carefully 'crafted' to help the policy generalize well at test time [29, 21, 7]. Data quality emerges as the paramount concern for training imitation learning policies. Despite ACT's CVAE-based modeling to capture multi-modal human behaviors, it is still sensitive to noise, inconsistencies, or suboptimal trajectories. Adequate coverage of the task state space is equally critical to improve robustness to distribution shift. It is therefore essential to prioritize quality and diversity over sheer quantity to maximize learning efficiency and generalization.

1) Teleoperation Module: We created a custom XBox teleoperation module compatible with LeRobot dataset API to streamline experimentation across multiple policies. As LeRobot supports a limited number of robots, we successfully extended their codebase to integrate our custom gym-xarm environment with automated evaluation methods to scale our experiments. To enhance dataset diversity and promote robust policy learning, we introduce randomization across both the cube's initial position and the target location in each episode.

2) Camera positioning: (Figure 1) outlines camera configurations in our simulation setup.

3) Final dataset: All recorded demonstrations are stored in the `LeRobot` dataset format, enabling seamless integration across multiple policies. The dataset used for our best performing models is available on HuggingFace with the following format:

- Observation space:
  - 4 224×224 RGB camera views
  - 512×6 point cloud obtained from FPS downsampling of a point cloud obtained from a 84×84 front RGB-D camera for 3DEgoAct
  - 7D proprioceptive state (6 joints + EEF θ)
- Action space: A 4D control vector $[\Delta X, \Delta Y, \Delta Z, W]$, where $[\Delta X, \Delta Y, \Delta Z]$ specifies incremental EEF cartesian delta for the end effector location, and W is the gripper command.
- Dataset contains 50 episodes sampled at 50Hz, and each approximately 20 seconds in length

C. Sampling-Based Exploration

We propose an RRT*-based [8, 25] trajectory generation framework designed for synthetic data augmentation under joint and workspace constraints, where teleoperated demonstrations(states) are used as ground truth data (in other words, they are weak supervisory signals rather than strict motion targets). The method explicitly addresses the challenges of noise, jitter, and over-constrained waypoint tracking that arise when naively replaying teleoperation trajectories using sampling-based planners [9, 10, 23].

After offline keyframe filtering to suppress teleoperation noise and retain only semantically meaningful states, execution proceeds using a lookahead-based planning strategy that promotes long-horizon smoothness. Intermediate keyframes are treated as soft progress constraints, triggering replanning upon being reached rather than serving as explicit planning targets. This separation of planning objectives from progress evaluation enables smooth, feasible motion without enforcing brittle, point-wise replay of demonstrated trajectories. Complete algorithmic details are provided in Appendix A.

D. Training Policies - ACT Architecture

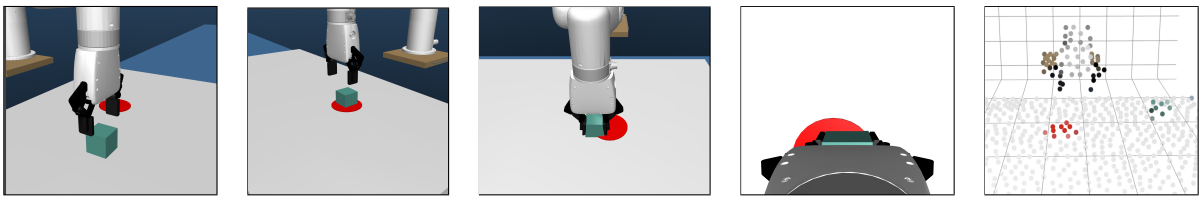
The Action Chunking Transformer [29] is an imitation learning policy that addresses the multi-modal and temporal nature of robotic manipulation by formulating the task as a Conditional Variational Autoencoder (CVAE). The observation space s_t comprises visual inputs and proprioceptive state. Visual observations are encoded via a ResNet-18 backbone, while proprioceptive states are projected via a multilayer perceptron (MLP). These features are concatenated and fed into a Transformer encoder to obtain the observation embedding. ACT predicts a sequence of future actions $a_{t:t+k}$, termed an "action chunk," rather than a single time-step action.

During training, the CVAE encoder approximates the posterior distribution $q_\phi(z|s_t, a_{t:t+k})$ of a latent variable z, which captures the variance and style of the demonstration data. The decoder, implemented as a Transformer, is conditioned on both the latent variable z and the observation embedding s_t to reconstruct the action sequence, effectively learning the policy $\pi(a_{t:t+k}|s_t, z)$. The objective function minimizes the reconstruction L1 norm loss $\mathcal{L}_{recon}$ of the action sequence combined with a Kullback-Leibler (KL) divergence term:

$$\mathcal{L} = \mathcal{L}_{recon} + \beta D_{KL}(q_\phi(z|s_t, a_{t:t+k})||p(z)) \quad (1)$$

where $p(z)$ is a standard isotropic Gaussian prior.

To mitigate temporal inconsistencies and jerky motion during inference, ACT employs *temporal ensembling*. Rather than executing a single predicted chunk open-loop, the policy is queried at every time step t, producing overlapping action predictions for the same future time index. The final executed action at time t is computed as a weighted average of all predictions covering that time step, where weight of action predicted for time $t - i$ is $w_i = \exp(-m \cdot i)$. This aggregation strategy smooths the trajectory and enhances robustness against distributional shift.



(a) Left view (b) Right view (c) Front view (d) EEF view (e) Point cloud

Fig. 1: Multi-view recording from 4 different cameras. In our final model, we found training on 2 cameras: (c) and (d) outperformed all other configurations on a fixed training budget. EEF(d) and downsampled point cloud(e) was used in 3DEgoACT.

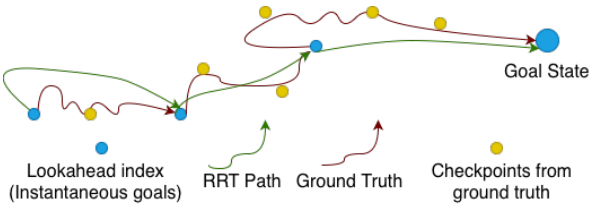


Fig. 2: RRT* path integration with teleoperated data

E. 3DEgoACT: Fusing 3D geometric priors with egocentric cues

We draw inspiration from 3D Diffusion Policy (DP3) [27], which achieves strong viewpoint generalization by integrating simple 3D representations with diffusion-based action generation. DP3 constructs a sparse point cloud from one or more RGB-D views, downsamples it, and encodes it into a compact fixed-dimensional vector. This 3D token, concatenated with robot proprioception, conditions a denoising diffusion model that iteratively refines Gaussian noise into coherent action sequences.

1) Fusing 2D egocentric view: While DP3 utilizes no egocentric views, we hypothesize that it is central for fine-grained manipulation such as pick-and-place. We therefore came up with 3DEgoAct, which fuses 3D point cloud representations with features from an egocentric camera. An RGBXYZ point cloud is embedded into a 512×1 token by the PointNet network[19], and is fused with the 2D egocentric ResNet feature embeddings via self-attention. We hypothesize that point cloud embedding would provide robust and viewpoint invariant geometric priors for environment understanding, while egocentric feature embeddings provide crucial information for fine-grained manipulation.

2) Point Cloud generation and downsampling: We follow [27] to generate and embed the point cloud. With the base of the robot as the world coordinate origin, we compute camera extrinsic and intrinsic matrices to recover a 6D RGBXYZ point cloud from a single RGB-D camera. DP3 [27] perform static thresholding of table surface. However, we do not take such an approach as it is not practical for noisy depth cameras which would cause intermittent disappearance of artifacts. Utilizing Farthest Point Sampling, we downsample the cloud down to 512 points for eliminating redundancy and enhancing run-time efficiency. We also chose to retain RGB information as it was crucial target localization.

F. Knowledge Co-Distillation

We implemented a knowledge co-distillation framework to transfer learned behaviors between fundamentally different robot policy architectures for visuomotor manipulation. Our goal is to combine the efficiency and task-specific precision of lightweight imitation learning models with the semantic richness and generalization capabilities of large-scale vision-language-action (VLA) models. In particular, we bridge ACT with SmolVLA, a VLA model with approximately 450 million parameters that integrates a vision-language backbone with a diffusion-based action prediction head. SmolVLA leverages natural language task descriptions and pretrained multimodal representations to enable few-shot generalization across tasks and environments.

In parallel, we explored a similar teacher–student distillation setup using the $\pi_{0.5}$ group of models by Physical Intelligence[4], which provides an alternative VLA-based supervisory signal. Both SmolVLA and $\pi_{0.5}$ serve as high-capacity teacher models, guiding ACT toward more robust and semantically grounded policies without incurring the full inference cost of large foundation models at deployment time.

Knowledge distillation transfers the learned behavior from a *teacher* model to a *student* model. In our bidirectional framework, either model can serve as teacher or student:

- VLA $\rightarrow$ ACT: The vision-language model acts as teacher, transferring its rich semantic understanding to the lightweight ACT model. This allows ACT to benefit from the VLA's pretrained knowledge while maintaining fast inference speed.
- ACT $\rightarrow$ VLA: The task-specific ACT model acts as teacher, fine-tuning the VLA on domain-specific

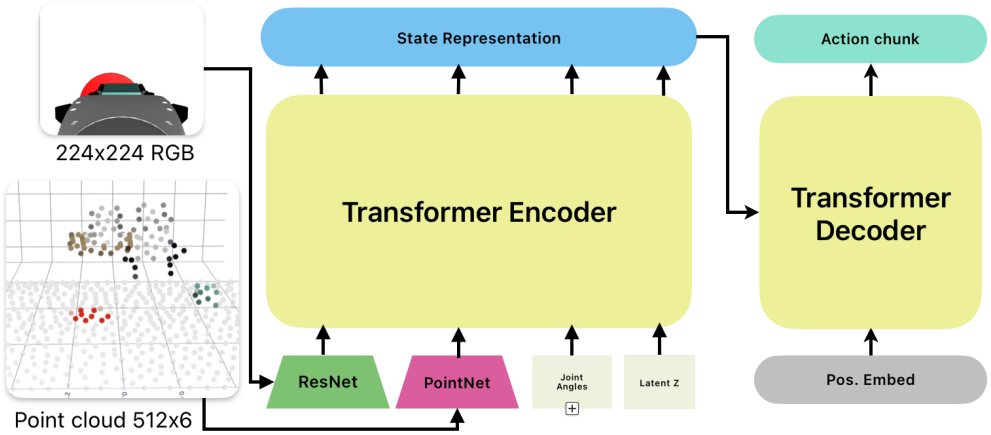


Fig. 3: 3DEgoACT: New PointNet encoder embeds 512×6 RGBXYZ point cloud into a 512×1 input token to transformer encoder providing global geometry information while EEF RGB provides local cues, enabling zero-shot generalization to shifts in allocentric viewpoint while retaining fine-grained dexterity.

behaviors. This helps adapt the general-purpose VLA model to the specific manipulation task.

IV. Experiments and Results

We first describe our evaluation strategy which is crucial to understand the structure of our experiments. Then, we describe our baseline, followed by development of 3DEgoACT. We then discuss RRT-based data generation and VLA co-distillation. We present the best performing policy achieved on a fixed training budget of 100 epochs.

A. Evaluation Strategy

To understand the sensitivity to changes in camera angles, in addition to running evaluation on the front view(training configuration), we perform zero-shot evaluation by perturbing the view to left and right camera configurations. Table I summarizes the results.

B. Baseline

1) Experiments: We train baseline ACT and explore key factors including camera count and positioning, learning rate, KL regularization β and chunk size k were studied to establish a reliable baseline. Initial attempts with the default LeRobot hyperparameters ($\beta = 10$) on small datasets (20–30 episodes). The robot either remained frozen or exhibited no signs of learning. As we improved our dataset over time, we saw noticeable improvement beyond $\beta = 20$, and set it as 30 for all our experiments. We set chunk size to 50 to match the 50Hz sampling rate as prescribed by [29]. A large learning rate of $7e^{-5}$ was set to complement a batch size of 856 on NVIDIA L4 GPU. The model had 51M parameters.

2) Results: Within a training budget of 100 epochs, the best performing policy was found to be the one trained on 2 camera(front+EEF) configuration. Table I summarizes the results per task stage. As hypothesized, all policies collapsed and failed to produce meaningful rollouts on our zero-shot viewpoint shift evaluation test. Evaluations were run on Macbook M1, and an inference speed of 24 actions/s was achieved. Representative policy rollouts can be found here ACT Evaluation

3) Beyond 2 cameras: A possible approach for achieving viewpoint invariance is to train from multiple view simultaneously. The baseline was trained on multiple allocentric camera configurations from 1, with the EEF camera present in all setups. It was observed that training the model with three cameras led to parallax where the robot could not accurately judge the position of the box, while four cameras resulted in significant computational overhead while training slow inference. We therefore chose the two camera configuration model as our baseline for further experiments.

C. 3DEgoACT Evaluation

1) Experiments:

a) Point Cloud generation: As described in Section III-E1, we generate a 512×6 RGBXYZ point cloud from a single 84×84 RGB-D camera after FPS downsampling. This cloud is embedded by PointNet (trainable parameters) to a 512×1 token. The new module increased the parameter count to 51.5M parameters.

b) Training: We trained 3DEgoACT for 100 epochs on feed from the EEF camera and the downsampled point cloud from the front RGB-D camera. Hyperparameters were set to those identified in the baseline experiments, while the learning rate for the new PointNet encoder was the same as the transformer.

Camera Position	Reach box	Pick box	Transport to target	Place precisely ($\delta \leq 0.02$)	Place within $\delta \leq 0.05$
Baseline ACT					
Front	45/50	40/50	35/50	27/50	39/50
Left	4/50	0/50	0/50	0/50	0/50
Right	3/50	0/50	0/50	0/50	0/50
3DEgoACT					
Front	48/50	44/50	40/50	30/50	37/50
Left	48/50	36/50	36/50	27/50	34/50
Right	47/50	38/50	35/50	25/50	32/50
3DEgoACT with no EEF camera					
Front	49/50	0/50	0/50	0/50	0/50
Left	47/50	0/50	0/50	0/50	0/50
Right	47/50	0/50	0/50	0/50	0/50

TABLE I: Success rates (out of 50 trials) for sequential task stages in the box placement task, broken down by camera view. Each policy was trained with the front camera as the allocentric view, and evaluated zero-shot on other views to study view-point sensitivity

2) Results: Proposed 3DEgoACT model demonstrates strong global understanding, exhibiting negligible performance degradation (refer Table I) across varying camera perspectives in zero-shot evaluation. Quantitative results reveal consistent success rates across the reach, pick, transport and place phases, while qualitative observations indicate smoother trajectories and enhanced spatial reasoning compared to the baseline. Evaluations were run on Macbook M1, and an inference speed of 14 action/s was achieved (23.4% drop from baseline). Notably, good performance was achieved with a negative temporal ensembling coefficient ($\alpha = -1 \times 10^{-3}$), suggesting greater weighting of recent actions. We attribute this to the coarse nature of the downsampled point cloud (Figure1(e)), which introduces abrupt shifts in feature representations that the policy must rapidly accommodate, necessitating weighing up of recent chunks.

3) Ablation: To assess the role of the end-effector (EEF) camera in enabling fine-grained control, we conducted an ablation by training the model using only the front-facing depth-RGB camera (no EEF view). The resulting policy achieved near-perfect performance (98% success) in object localization and approach phases but exhibited 0% success in transitioning to the grasping stage even after 250 epochs of training. This stark contrast underscores the critical importance of the EEF camera in our setup, as it provides high-resolution, egocentric cues essential for precise contact-rich interactions. Nonetheless, we hypothesize that improved point cloud representation through denser sampling, superior downsampling strategies (e.g., multi-token representations), or enhanced encoders, could mitigate this dependency, potentially allowing fully allocentric 3D policies to suffice for end-to-end manipulation. [27].

D. RRT Data Generation

The generated trajectory in Fig. 4 preserves the high-level intent of the teleoperated demonstration while avoiding brittle keyframe-by-keyframe tracking. By leveraging long-horizon planning and forced replanning through buffer pruning, the proposed method produces smoother end-effector motion with significantly reduced high-frequency spatial oscillations.

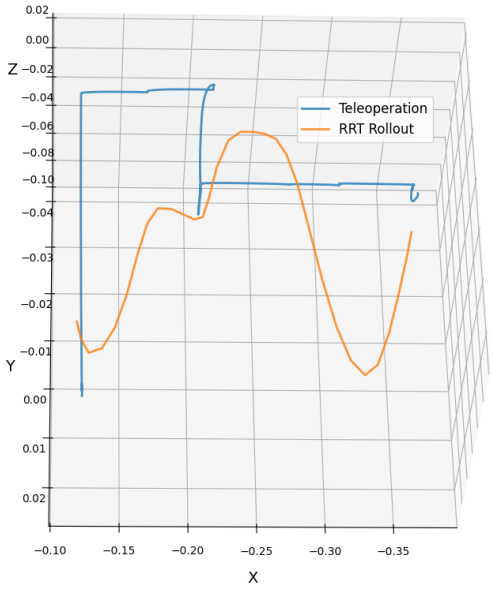


Fig. 4: Generated RRT* trajectory vs Teleoperated trajectory

This experiment serves as an ablation demonstrating the importance of long-horizon planning and trajectory pruning compared to naïvely planning toward each demonstrated keyframe. Despite being guided by a single teleoperated trajectory, the RRT-based strategy yields diverse yet feasible rollouts, as shown in the supplementary videos (Diversity of RRT* trajectories). In these examples, the robot successfully grasps the object from different lateral approaches, highlighting the inherent

variability enabled by the RRT-based data augmentation framework.

E. Co-Distillation Experiments

We conducted a series of experiments using both SmolVLA and $\pi_{0.5}$ as Vision-Language-Action (VLA) models within our manipulation pipeline. For SmolVLA, we enabled end-to-end inference on the xArm pick-and-place task using visual observations and natural language instructions, and performed fine-tuning on our demonstration dataset VLA). SmolVLA served the purpose as a teacher in the co-distillation setup with ACT, where ACT's task loss decreased from 1.18 to 0.34 (a 71% reduction) over the course of training. However, the distillation loss showed minimal improvement, decreasing from 0.58 to 0.53. In parallel, we integrated $\pi_{0.5}$ as an alternative VLA teacher model and validated action prediction through partial fine-tuning and qualitative rollouts. Due to architectural complexity and computational constraints, $\pi_{0.5}$ experiments were restricted to short training runs and inference-only evaluations, as they confirmed the feasibility of extending the distillation framework to multiple VLA teachers.

F. Limitations and Challenges Faced

1) Baseline: ACT is sensitive to hyperparameters such as KL weight and data quality, which necessitated multiple training runs and dataset versions. While we fixed our training budget to 100-200 epochs, it is possible that multi-camera setup outperforms 2 camera when trained for a long duration. However, since we focus on low-cost policies, we confine ourselves to 2 cameras.

2) 3DEgoACT: While robust to viewpoint shifts, we notice that almost all policy failures occur in the case when the robot arm obscures the object or target. We attribute this to the coarse nature of the downsampled point cloud (Figure 1(e)). Better feature aware downsampling techniques can be utilized for generating point cloud to filter artifacts such as the table to better utilize the point limit to objects of interest. The policy should also be evaluated on inference time camera movement to verify the sensitivity to temporal shifts.

3) RRT based synthetic data generation: While RRT* is effective in producing collision-free motion trajectories with variability across runs, we observe that directly applying it to object manipulation tasks lacks robustness and is highly sensitive to hyperparameters such as step size, planning horizon, and the minimum spatial threshold used to identify relevant observation states. The problem of *reaching* a target configuration, and then *interacting* with an object constitute fundamentally different challenges. A single RRT* planning layer is insufficient to reliably address both, necessitating an additional layer that explicitly incorporates goal and interaction-aware processing and task semantics. Although we demonstrate successful execution in isolated cases using carefully tuned parameters (object transport task video), the same parameter configuration fails to generalize across different initial conditions or task variations. This highlights the need for principled integration of interaction dynamics beyond standalone sampling-based planning [5, 14].

4) VLA Co-distillation: Our initial approach attempted to use $\pi_{0.5}$-droid, a state-of-the-art VLA. However, we encountered a fundamental domain gap problem: $\pi_{0.5}$-droid, trained exclusively on real-world images, failed to recognize objects or produce meaningful actions in MuJoCo simulation. Visual inputs in MuJoCo lack the visual complexity and realism that the model expected, making the input distribution drastically different from its training data. We explored fine-tuning approaches including Low-Rank Adaptation (LoRA), but encountered technical barriers with framework compatibility between JAX (used by $\pi_{0.5}$) and PyTorch (used by our training pipeline). Another key challenge in cross-architecture distillation is reconciling differences in input and output formats across models. ACT predicts action chunks in a 4D end-effector action space, whereas SmolVLA produces single-step actions in a higher-dimensional space. To enable distillation, we align the action representations by retaining the shared 4D components corresponding to end-effector position and gripper state, and compare with either the first or the mean action predicted by ACT's chunk with SmolVLA's single-step output.

V. Conclusion

We established an ACT baseline trained on a self-collected dataset, achieving a task success rate of 78% under a fixed training budget. Through systematic experimentation, we evaluated multiple dataset collection strategies, conducted ablations on ACT training and showed that it collapses under viewpoint shifts. We propose 3DEgoACT, which leverages a sparse allocentric point cloud for global cues along with a 2D RGB embeddings from EEF egocentric view for fine-grained manipulation. We find that egocentric features are crucial for downstream performance as demonstrated by our ablations. Our policy demonstrates strong global understanding, exhibiting negligible performance degradation across varying camera perspectives on zero-shot evaluation. Quantitative results reveal consistent success rates across the reach, pick, transport and place phases, while qualitative observations indicate smoother trajectories and enhanced spatial reasoning compared to the baseline.

A sampling-based data augmentation approach using RRT* that showed promising diversity was developed. Furthermore, we analyzed the effect of distilling task-

specific visuomotor policies, such as ACT and diffusion-based policies, from foundation models.

Future work will focus on comparing policies trained solely on teleoperated demonstrations against those trained with additional synthetically generated trajectories. Secondly, extending the synthetic data generation pipeline to explicitly account for interaction-centric manipulation tasks, enabling more reliable learning in contact-rich settings.

VI. Team Member Contributions

- **Rishabh Jain**: Implemented tele-operation and dataset collection pipeline, implemented and trained ACT, 3DEgoACT and ran evaluations.
- **Rohit Ramesh**: VLA-ACT Co-distillation pipeline, VLA fine-tuning.
- **Tejal Bedmutha**: RRT-based planning, dataset integration, planner evaluation.

VII. Acknowledgments

- Discussion with TAs:
  - Suggested to implement ACT as baseline policy.
  - Results: Implemented the same with two different methods of dataset collection.
- Discussion with Course Instructor:
  - Suggested to implement local search technique for dataset collection.
  - Results: Implemented the same with RRT sampling technique.

References

[1] Ali Abouzeid, Malak Mansour, Zezhou Sun, and Dezhen Song. Geoaware-vla: Implicit geometry aware vision-language-action model. *arXiv preprint arXiv:2509.14117*, 2025.

[2] Sumeet Batra and Gaurav Sukhatme. Zero-shot visual generalization in robot manipulation, 2025. URL https://arxiv.org/abs/2505.11719.

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*.

[4] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025.

[5] brian ichter, Pierre Sermanet, and Corey Lynch. Broadly-exploring, local-policy trees for long-horizon task planning. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=yhy25u-DrjR.

[6] Anzhe Chen, Yifei Yang, Zhenjie Zhu, Kechun Xu, Zhongxiang Zhou, Rong Xiong, and Yue Wang. Toward embodiment equivariant vision-language-action policy. *arXiv preprint arXiv:2509.14630*, 2025.

[7] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[8] Howie Choset. Rapidly–exploring random trees (rrts) — lecture slides for ri 16-735. Lecture slides (PDF), 2023. URL https://www.cs.cmu.edu/~motionplanning/lecture/lec20.pdf. Slides by Howie Choset with materials from James Kuffner.

[9] H. Fan, J. Huang, X. Huang, H. Zhu, and H. Su. Bi-rrt*: An improved path planning algorithm for secure and trustworthy mobile robots systems. *Heliyon*, 10(5):e26403, February 2024. doi: 10.1016/j.heliyon.2024.e26403.

[10] Yaolin Ge, Jo Eidsvik, and André Julius Hovd Olaisen. Rrt*-enhanced long-horizon path planning for auv adaptive sampling using a cost valley. *Know.-Based Syst.*, 315(C), April 2025. ISSN 0950-7051. doi: 10.1016/j.knosys.2025.113261. URL https://doi.org/10.1016/j.knosys.2025.113261.

[11] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation, 2023. URL https://arxiv.org/abs/2306.17817.

[12] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation, 2023. URL https://arxiv.org/abs/2306.14896.

[13] Helong Huang, Min Cen, Kai Tan, Xingyue Quan, Guowei Huang, and Hong Zhang. Graphcot-vla: A 3d spatial-aware reasoning vision-language-action model for robotic manipulation with ambiguous instructions, 2025. URL https://arxiv.org/abs/2508.07650.

[14] Haewon Jung, Donguk Lee, Haecheol Park, Jun-Hyeop Kim, and Beomjoon Kim. `SPIN`: distilling `Skill-RRT` for long-horizon prehensile and non-prehensile manipulation, 2025. URL https://arxiv.org/abs/2502.18015.

[15] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[16] Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 13:162467–162504,

2025. ISSN 2169-3536. doi: 10.1109/access.2025.3609980. URL http://dx.doi.org/10.1109/ACCESS.2025.3609980.

[17] Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation, 2025. URL https://arxiv.org/abs/2508.17600.

[18] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021. URL https://arxiv.org/abs/2107.14483.

[19] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017. URL https://arxiv.org/abs/1612.00593.

[20] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation, 2022. URL https://arxiv.org/abs/2209.05451.

[21] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

[22] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

[23] Y. Sun and S. Zhang. Multi-strategy improved rapid random expansion tree (rrt) algorithm for robotic arm path planning. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 16(3):416–422, 2025. URL https://thesai.org/Downloads/Volume16No3/Paper_41-Multi_Strategy_Improved_Rapid_Random_Expansion_Tree.pdf. Accessed: 2025-12-15.

[24] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2410.00425.

[25] Konstantinos Tsianos, Dan Halperin, Lydia Kavraki, and Jean-Claude Latombe. *Robot algorithms*, page 4. Chapman & Hall/CRC, 2 edition, 2010. ISBN 9781584888208.

[26] Maryam Zare, Parham M. Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges, 2023. URL https://arxiv.org/abs/2309.02473.

[27] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations, 2024. URL https://arxiv.org/abs/2403.03954.

[28] Tianyi Zhang, Haonan Duan, Haoran Hao, Yu Qiao, Jifeng Dai, and Zhi Hou. Grounding actions in camera space: Observation-centric vision-language-action policy. *arXiv preprint arXiv:2508.13103*, 2025.

[29] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[30] Jiaming Zhou, Ke Ye, Jiayi Liu, Teli Ma, Zifang Wang, Ronghe Qiu, Kun-Yu Lin, Zhilin Zhao, and Junwei Liang. Exploring the limits of vision-language-action manipulations in cross-task generalization. *arXiv preprint arXiv:2505.15660*, 2025. URL https://jiaming-zhou.github.io/AGNOSTOS/.

APPENDIX

DETAILED TELEOPERATION-GUIDED RRT

1) Problem Setting: A teleoperated demonstration is given as a sequence of observations

$$\mathcal{D} = \{o_0, o_1, \ldots, o_T\},$$

where each state is defined as

$$s_t = (x_t, y_t, z_t, g_t).$$

Here, (x_t, y_t, z_t) denotes the Cartesian position of the end effector, and g_t is a discrete variable indicating the gripper state (open or closed). Our objective is to generate a smooth, dynamically feasible trajectory that:

- respects kinematic and joint constraints,
- preserves the high-level intent of the demonstration,
- avoids jerky motion induced by high-frequency teleoperation noise.

Rather than enforcing exact waypoint tracking, we treat demonstrations as providing *structural guidance* for motion generation.

2) Offline Keyframe Filtering: Teleoperated demonstrations often contain redundant, noisy samples due to human control jitter. To remove this redundancy, we perform an offline spatial filtering step prior to planning.

Let $p_t \in R^3$ denote the absolute end-effector position corresponding to observation o_t. A new observation is retained only if:

$$\|p_t - p_{\text{last}}\|_2 > \epsilon,$$

where $\epsilon = 4$ cm is a variance threshold.

This produces a filtered keyframe sequence

$$\mathcal{D}' = \{o'_0, o'_1, \ldots, o'_K\},$$

which captures only semantically meaningful spatial progress. This filtering is performed entirely offline and remains fixed throughout execution.

3) Decoupling Planning Targets from Progress Constraints: A central design principle of our method is the decoupling of *planning targets* from *progress constraints*.

Filtered observations in $\mathcal{D}'$ are not treated as explicit planning goals. Planning sequentially to each keyframe results in short-horizon replanning and oscillatory motion. Instead, we introduce a lookahead-based planning strategy.

At filtered index i, the planner targets a future lookahead state:

$$s_{\text{goal}} = s_{i+L},$$

where L is a fixed or adaptively increased lookahead horizon. This lookahead goal is used to compute a directional bias vector:

$$d = s_{i+L} - s_{\text{current}},$$

which guides RRT expansion without imposing a hard constraint.

Importantly, the immediate next filtered observation o'_{i+1} is never used as a planning goal.

4) Buffered Local Execution: Once an RRT path is found to the lookahead target, the resulting sequence of waypoints is stored in a buffer. Execution proceeds incrementally: at each control step, only a single waypoint is executed.

This design ensures that planning remains responsive to state changes and environmental constraints, while still benefiting from long-horizon smoothness.

5) Progress Detection via Soft Waypoint Constraints: While filtered observations are not planning goals, they act as *progress checkpoints*. After each execution step, the algorithm evaluates proximity to the next filtered observation:

$$\|p_{\text{current}} - p_{i+1}\|_2 < \delta,$$

where δ corresponds to the planner step size.

If this condition is satisfied, the filtered index is advanced. This indicates that the robot has implicitly passed the demonstrated waypoint while following a longer-horizon plan.

Crucially, filtered waypoints are treated as *soft constraints*. If a waypoint cannot be reached exactly due to kinematic or environmental constraints, the algorithm does not enforce hard tracking, thereby avoiding deadlock.

6) Forced Replanning and Directional Correction: Upon advancing the filtered index, any remaining buffered trajectory is discarded. This forced pruning ensures that future motion is replanned from the updated state with a newly computed directional bias:

$$d_{\text{new}} = s_{(i+1)+L} - s_{\text{current}}.$$

This mechanism prevents stale long-horizon plans from dominating execution and enables local directional correction at each meaningful stage of the demonstration. As a result, the method achieves a balance between global smoothness and local trajectory fidelity.

7) Handling Planning Failures and Local Infeasibility: In some cases, RRT may fail to find a valid non-trivial path from the current state to the lookahead target due to joint limits, collisions, or inconsistencies in the demonstration. To ensure robustness, we introduce a bounded replanning strategy.

If no valid path is found after a maximum number of replanning attempts, the algorithm advances the filtered index regardless. This reflects the interpretation of filtered waypoints as guidance rather than strict constraints.

After advancing the index, planning resumes toward the next lookahead target, and a new directional bias is computed from the current state. This local recovery mechanism ensures forward progress and prevents the system from stalling due to infeasible intermediate configurations.

8) Local Path Interpretation: Each planned trajectory segment should be interpreted as a *local feasible path* conditioned on the current state and lookahead intent. These paths are not globally optimal or guaranteed to pass through all filtered waypoints exactly. Instead, they provide smooth, constraint-respecting motion that approximately follows the demonstrated structure.2

By repeatedly replanning local paths from updated states, the algorithm constructs a globally coherent trajectory without enforcing brittle waypoint tracking.

TABLE II: Training Configuration

Hyperparameter
Batch size: 32 samples
Learning rate: 3×10^{-5} with cosine annealing
Training steps: 10,000 iterations
Optimizer: AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with weight decay 0.01
Hardware: Single NVIDIA L4 GPU

VLA CO-DISTILLATION CONFIGURATION

The student model is trained using a combination of two loss terms:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{task}} + \beta \cdot \mathcal{L}_{\text{distill}} \tag{2}$$

where $\mathcal{L}_{\text{task}}$ is the standard behavior cloning loss and $\mathcal{L}_{\text{distill}}$ is the distillation L2 loss that encourages the student's predicted actions to match the teacher's. We empirically set $\alpha = 0.3$ and $\beta = 0.7$ to balance the two objectives. The distillation loss is computed as:

$$\mathcal{L}_{\text{distill}} = \frac{1}{N} \sum_{i=1}^{N} \| a^{(i)}_{\text{student}} - a^{(i)}_{\text{teacher}} \|^2 \tag{3}$$

where a_{student} and a_{teacher} are the predicted actions from the student and teacher models respectively.