POINT BRIDGE: 3D Representations for Cross Domain Policy Learning

Anonymous Author(s)

***Abstract*— Robot foundation models are beginning to deliver on the promise of generalist robotic agents, yet progress remains constrained by the scarcity of large-scale real-world manipulation datasets. Simulation and synthetic data generation offer a scalable alternative, but their usefulness is limited by the visual domain gap between simulation and reality. In this work, we present POINT BRIDGE, a framework that leverages unified, domain-agnostic point-based representations to unlock synthetic datasets for zero-shot sim-to-real policy transfer—without explicit visual or object-level alignment. POINT BRIDGE combines automated point-based representation extraction via Vision-Language Models (VLMs), transformer-based policy learning, and efficient inference-time pipelines to train capable real-world manipulation agents using only synthetic data. With additional co-training on small sets of real demonstrations, POINT BRIDGE further improves performance, substantially outperforming prior vision-based sim-and-real co-training methods. It achieves up to 44% gains in zero-shot sim-to-real transfer and up to 66% with limited real data across both single-task and multitask settings. Videos of the robot are best viewed at: https://pointbridge-anon.github.io/.**

I. INTRODUCTION

Deep learning has recently undergone a paradigm shift, moving from narrow task-specific models to generalist systems capable of complex reasoning [1], [2], [3], generating photorealistic images [4] and videos [5], and even writing code [6]. This progress has been fueled by internet-scale training data paired with scalable architectures. Lately, robot foundation models are starting to realize some of the promise of large-scale data and the training paradigm from these domains. However, unlike vision and language, which can directly exploit internet-scale datasets, robotics is inherently interactive: models must learn from datasets that contain physical interactions with the real world. This makes collecting large-scale robotic data time-consuming, prohibitively expensive, and fundamentally difficult to scale, creating a central bottleneck for building generalist robotic intelligence.

The prevailing paradigm for robot policy learning relies on large-scale teleoperated datasets, followed by training neural policies on them. While effective, this approach often requires months or years of data collection and still produces datasets far smaller than those in vision and language [7]. Simulation is a promising alternative to address this need for data, especially due to recent progress. Simulation environments are becoming easier to design, with the availability of high-fidelity physics simulators [8], [9] and the emergence of generative AI tools that automate asset and scene generation [10], [11]. Recently developed synthetic data generation tools can generate large-scale, high-quality robot manipulation demonstration datasets in such simulation environments with minimal human effort [12], [13], [14], [15]. Furthermore, recent work has shown that such synthetic simulation datasets can easily train high-performing real-world manipulation agents by co-training on these datasets and small numbers of real-world demonstrations [16], [17], [18], suggesting that synthetic simulation data could potentially reduce the dependence on large real-world datasets. However, these methods can still require careful sim and real alignment, and still rely on the presence of real-world data, owing to the mismatched representation of data between the domains. Human videos offer another scalable and complementary source of supervision, but again face challenges from the embodiment gap between human and robot morphologies as well as the representation mismatch between the domains.

A recent line of work proposes task-relevant keypoint representations [19], [20], [21] as a potential solution to this domain representation gap. By abstracting both the robot and scene into sets of keypoints, these methods enable policies that are agnostic to raw visual appearance and generalize across objects and environment conditions. However, existing approaches often rely on human annotations [19], [21], focus on bridging embodiment but not visual differences [22], [23], and are often restricted to single-task settings. We argue that such representations only scratch the surface of what is possible.

In this work, we introduce POINT BRIDGE, a framework that uses unified domain-agnostic point-based representations to unlock the potential of synthetic simulation datasets and enable zero-shot sim-to-real policy transfer. POINT BRIDGE trains real-world manipulation agents starting with just a handful of teleoperated demonstrations in simulation by using synthetic data generation tools. It then leverages advances in vision-language models (VLMs) to build unified scene representations that facilitate cross-domain policy transfer. Our core insight is that unifying representations across simulation and real-robot teleoperation unlocks scalable sim-to-real transfer without requiring explicit visual or object-level alignment. Such a representation further supports scaling to multi-task policies through transformer-based architectures, providing a framework that scales with data availability. POINT BRIDGE operates in three stages. First, scenes are filtered into point cloud–based representations aligned to a common reference frame. In simulation, this is obtained directly from object meshes, while in real experiments, we use our automated VLM-guided pipeline for keypoint extraction on task relevant objects. Second, a transformer-based policy architecture is trained on these unified point clouds for policy learning. Finally, during deployment, we employ a lightweight pipeline for scene extraction designed to minimize the sim-to-real gap, leveraging VLM filtering and supporting multiple

3D sensing strategies to balance performance and throughput.

We demonstrate the effectiveness of POINT BRIDGE on six real-world tasks, using data collected through simulation and real robot teleoperation. Our main findings are as follows:

1) We develop POINT BRIDGE, a framework that uses unified domain-agnostic point-based representations to harness synthetic simulation data and enable zero-shot sim-to-real policy transfer.
2) POINT BRIDGE contains novel components including (1) a VLM-based point extraction pipeline that bridges the visual sim-to-real gap with minimal human effort, and (2) multiple inference-time pipelines to adapt to different user needs with respect to performance and throughout.
3) POINT BRIDGE improves by 39% and 44% on single-task and multitask zero-shot sim-to-real transfer. When co-trained with a small amount of real data, POINT BRIDGE improves over prior works by 61% and 66% in single-task and multitask settings, respectively. (Section III-B, III-C).

All of our datasets, training, and evaluation code will be made publicly available. Videos of our trained policies are best viewed at: https://pointbridge-anon.github.io/.

II. POINT BRIDGE

POINT BRIDGE introduces a unified scene representation that enables sim-to-real policy transfer with minimal alignment, incorporates co-training with real-world data, and facilitates multitask learning. An overview of the framework is provided below, with details discussed in the following sections.

A. Overview

POINT BRIDGE begins with a small dataset of human demonstrations $\mathcal{D}_{src}$, which is expanded into a larger dataset $\mathcal{D}_{sim}$ using synthetic data generation [13]. We also consider an optional setting where a small set of real-world demonstrations $\mathcal{D}_{real}$ is available for co-training. All observations are converted into a compact point-based representation $\mathcal{P}$, serving as input to policies mapping observations to actions. In simulation, these representations are obtained directly from the simulator, while in the real world, they are extracted via a VLM-guided scene filtering pipeline. During deployment, the same VLM pipeline provides task-relevant points in real time for policy inference. The resulting policies enable zero-shot sim-to-real transfer, joint training with real data, and multitask learning. Details about each component are provided in the subsequent sections.

B. Data Collection and Synthetic Data Generation

For our simulated tasks, we use the MimicLabs suite [24] to construct atomic tasks, each involving different pairs of object instances. For each task, we collect a small set of human demonstrations $\mathcal{D}_{\text{src}}$, which are then expanded into a much larger dataset $\mathcal{D}_{\text{sim}}$ using MimicGen [13], a synthetic data generation technique. MimicGen adapts each demonstration segment to novel scenes by applying a constant SE(3) transformation $T_W^{o_i'}(T_W^{o_i})^{-1}$, where $T_W^{o_i}$ is the pose of the source object o_i in the world frame, and $T_W^{o_i'}$ is the pose of the same object in the target scene. The inverse transformation $(T_W^{o_i})^{-1}$ maps from the world frame to the source object's local frame, and the full product maps poses from the source object's frame to the target object's frame in the new scene. This transformation preserves the relative geometry between the end effector and the object from the source demonstration when adapting to new object poses. As a result, MimicGen enables a small set of demonstrations to be multiplied many times over with novel object configurations and types, supporting generalizable policy learning on large-scale datasets.

C. Point Extraction

Each observation in the dataset is now distilled into a compact set of task-relevant 3D keypoints. These keypoints serve as the unified representation used for downstream policy learning. The pipeline comprises two stages: (1) identifying task-relevant objects in the scene, and (2) extracting 3D keypoints for those objects.

a) VLM-Guided Scene Filtering: Given an initial scene image $\mathcal{I}_0$ and a natural language task description $\mathcal{L}$, we first use `Gemini-2.5-flash` to identify the set of task-relevant objects in the scene, denoted as $\{l^1, \ldots, l^k\}$. For example, for the command *"put the bowl on the plate"*, the model returns the object set bowl, plate. After determining the object categories, we employ Molmo-7B [25] to localize these objects as pixels $\{o^{p_1}, \ldots, o^{p_k}\}$ in the image.[1] These pixel coordinates serve as initialization for SAM2 [26], which extracts 2D segmentation masks $\{m_0^1, \ldots, m_0^k\}$ for each identified object. For subsequent frames in the trajectory, we leverage SAM2's built-in memory to propagate masks consistently and track objects robustly over time, enabling reliable handling of occlusions during both data collection and deployment.

b) 3D Projection of Task Objects: For each timestep t, N 2D object points $\mathcal{P}_t^{2D}$ are sampled uniformly from each object segmentation mask $m_t^i, \forall i \in \{1, \ldots, k\}$. To improve robustness near mask boundaries, each segmentation mask is first shrunk inward by 20% before sampling, which reduces the chance that points close to the mask edge are projected onto background geometry due to noisy depth estimates or imperfect segmentation. A stereo image pair is then used to compute depth $\mathcal{I}_t^d$ with Foundation Stereo [27]. This depth map, along with camera intrinsics and extrinsics, lifts $\mathcal{P}_t^{2D}$ to 3D. FoundationStereo generally produces less noisy depth than commodity RGB-D sensors, especially for shiny or transparent objects. To reduce redundancy while maintaining coverage, we apply farthest point sampling to downsample each object to M ($\ll N$) representative points. Finally, all object points are transformed into the robot base frame using camera extrinsics. We denote the final set as $\mathcal{P}_t^{3D}$.

[1] In our experiments, `Gemini-2.5-flash` was effective for text-based object identification but less reliable for spatial localization, motivating the use of a specialized VLM for the pointing task. As multi-modal VLMs advance, a unified model could eventually replace this modular approach.

c) Considerations for simulation data: In simulation, we bypass VLM-based detection and directly sample 3D points from task-relevant object meshes. However, mesh-based sampling covers all object surfaces, while real cameras only capture visible surfaces from specific viewpoints. To bridge this gap, we replicate real camera setups by applying the corresponding extrinsic (R, t) and intrinsic (K) parameters. Each mesh point X_{mesh} is projected to the image plane as $\tilde{x} = K[R|t]X_{\text{mesh}}$, and the pixel coordinate is $x = (\tilde{x}_1/\tilde{x}_3,\ \tilde{x}_2/\tilde{x}_3)$. We then use the ground-truth depth map $D(x)$ to lift the point back to 3D: $X_{\text{cam}} = D(x)K^{-1}\begin{bmatrix} x & 1 \end{bmatrix}$. These points are transformed into the robot's base frame for consistency. Finally, to account for sensor noise absent in simulation, we inject Gaussian noise with a 1 cm standard deviation into the point clouds to improve robustness to real-world observations.

d) Robot Representation: Similar to [19], we represent the robot end effector as a set of keypoints on the gripper. Given the robot pose T_r^t at timestep t, we define N rigid transformations T about this pose and compute the pose at each robot keypoint T_r^t such that

$$(T_r^t)^i = T_r^t \cdot T^i, \quad \forall i \in \{1, ..., N\} \tag{1}$$

The positions of the robot key points $(\mathcal{P}_r^t)^i\ \ \forall i \in \{1, ..., N\}$ are then extracted from these poses.

D. Policy Learning

We take inspiration from BAKU [28] and use a decoder-only multi-task transformer architecture for policy learning. Robot points $\mathcal{P}_r$ and object points $\mathcal{P}_o$ are combined into a point cloud $\mathcal{P}$, encoded with a PointNet [29] encoder. For multitask learning, we also input a language embedding $\mathcal{L}$, encoded using the 6-layer MiniLM [30] from Sentence Transformers [31]. The encoded representations serve as input to a BAKU transformer policy with a deterministic action head that outputs the robot end-effector pose and gripper state. Mathematically,

$$\begin{aligned} \mathcal{O}^{t-H:t} &= \{\mathcal{P}_r^{t-H:t},\ \mathcal{P}_o^{t-H:t},\ \mathcal{L}\} \\ \hat{\mathcal{A}}^{t+1} &= \pi(\cdot|\ \mathcal{O}^{t-H:t}) \end{aligned} \tag{2}$$

where H is the history length, π the learned policy, and $\mathcal{A}$ the predicted action. Following prior work in policy learning [32], [33], we use action chunking with exponential temporal averaging to ensure smoothness of the predicted tracks. The policy is optimized with mean squared error (MSE) over ground-truth and predicted actions.

E. Policy Inference

During real-world deployment, the initial scene image $\mathcal{I}_0$ and task instruction $\mathcal{L}$ are used to obtain 2D object keypoints $\mathcal{P}_0^{2D}$, which are projected to 3D using scene depth and camera parameters. Section II-C describes our primary approach with stereo images and Foundation Stereo, but we also support depth from commodity RGB-D sensors and point triangulation from two RGB cameras [19]. For RGB-D sensors, depth comes directly from the sensor depth map. For triangulation, 2D keypoints from one camera view are transferred to the other via MAST3R [34], and Co-Tracker [35] tracks them throughout the trajectory. 3D keypoints $\mathcal{P}_0^{3D}$ are then computed by triangulating tracked corresponding 2D points from multiple views and transforming them into the robot base frame. In subsequent timesteps, Co-Tracker separately tracks 2D keypoints $\mathcal{P}_t^{2D}$ in both views, followed by multi-view triangulation to extract $\mathcal{P}_t^{3D}$ in the base frame. This flexible pipeline with multiple depth sensing strategies enables the same trained policy to be deployed across diverse real setups.

III. Experiments

We provide details on our experimental setup (Sec. III-A) and subsequently show how Point Bridge effectively enables zero-shot sim-to-real policy transfer from synthetic simulation data (Sec. III-B) and how Point Bridge performance can be improved even further with a small amount of real-world data (Sec. III-C). Finally, we conduct a systematic analysis of the components in Point Bridge (Sec. **??**).

A. Experimental Setup

We evaluate manipulation tasks with significant variability in object type and placement, under minimal visual and object alignment between simulation and reality. We use Deoxys [36] at 20 Hz as the robot controller. Real-world experiments are conducted on a Franka Research 3 arm with a Franka Hand gripper. Demonstrations are collected at 20 Hz using RoboTurk [37] in simulation and Open Teach [38] in the real world, and subsampled to 10 Hz for training. For sensing, we use an Intel RealSense RGB-D and a ZED 2i stereo camera. Policies trained with Point Bridge and FoundationStereo for depth estimation run at 5 Hz, while image-based baselines reach 15 Hz. **In total, we perform 1410 real-world evaluations across varied task settings to benchmark performance.**

a) Environment Design and Data Generation: For our simulated experiments, we use the MimicLabs task suite [24] to design 3 atomic tasks - `bowl on plate`, `mug on plate`, and `stack bowls`. Each task includes 4 different object instance pairs. For every pair, a human demonstrator provides 5 demonstrations, which are scaled up to 300 using MimicGen [13], resulting in a total of 1200 demonstrations per task in simulation. For co-training, we supplement this with 45 teleoperated demonstrations in the real world across three additional object pairs, illustrating cross-domain variability. For real tasks such as `fold towel`, `close drawer`, and `put bowl in oven`, we only collect real-world data (20 demonstrations on a real robot).

B. Zero-Shot Sim-to-Real Transfer with Minimal Alignment

We evaluate Point Bridge for zero-shot sim-to-real transfer on 3 simulated tasks. Table I and Table II present the single-task and multitask results, respectively. Each configuration consists of 10 rollouts across 3 object-instance pairs, totaling 30 evaluations. For Point Bridge, each object is represented with 128 points, extracted using the VLM filtering pipeline. Our key findings are summarized below.

TABLE I: POINT BRIDGE enables zero-shot sim-to-real transfer in **single task** settings and shows further performance improvements when trained with small amounts of real-world data.

Observation Modality	Data Configuration	Bowl on plate	Mug on plate	Stack bowls
Image	Real	9/30	10/30	11/30
	Co-Train Sim	2/30	17/30	14/30
POINT BRIDGE	Real	25/30	25/30	24/30
	Zero-Shot Sim	23/30	21/30	24/30
	Co-Train Sim	**29/30**	**30/30**	**29/30**

TABLE II: POINT BRIDGE supports both zero-shot sim-to-real transfer and sim-real co-training in multi-task settings. Notably, multi-task learning shows improvements in performance over single-task training.

Observation Modality	Data Configuration	Bowl on plate	Mug on plate	Stack bowls
Image	Real	10/30	11/30	11/30
	Co-Train Sim	6/30	10/30	15/30
POINT BRIDGE	Real	22/30	26/30	24/30
	Zero-shot Sim	25/30	23/30	24/30
	Co-Train Sim	**30/30**	**30/30**	**30/30**

a) POINT BRIDGE *enables zero-shot sim-to-real transfer with minimal visual alignment.:* Our simulation and real-world setups differ significantly in table appearance, backgrounds, and lighting. Despite these differences, POINT BRIDGE's scene-filtering strategy produces domain-invariant representations, outperforming the strongest baseline by 39% in single-task transfer and 44% in multitask transfer. This stands in contrast to prior approaches, which often require carefully aligned scenes and reality [16] or photorealistic simulators [9] to achieve policy transfer. Image-based sim-to-real policies fail entirely in the zero-shot setting, and thus are excluded from the reported results for clarity.

b) POINT BRIDGE *enables zero-shot sim-to-real policy transfer across diverse object instances.:* The objects used in simulation look visually very different than those during deployment. Even under large discrepancies in visual appearance, POINT BRIDGE requires only minimal object alignment to transfer policies effectively. Additionally, by leveraging FoundationStereo for depth estimation, POINT BRIDGE is able to handle visually challenging objects such as transparent or reflective items, unlike depth sensing from RGB-D cameras, which typically struggles with such items.

c) POINT BRIDGE *enables multitask zero-shot sim-to-real transfer.:* We evaluate both single-task and multitask variants of POINT BRIDGE, where the multitask policy is conditioned on natural language instructions. Since POINT BRIDGE operates on filtered point cloud representations and is language-conditioned, it generalizes naturally to the multitask setting. Empirically, the multitask policy achieves comparable or better performance than single-task policies, demonstrating scalability across diverse tasks. Crucially, the underlying 3D point extraction and representation pipeline is task-agnostic, so the same perception stack can be reused for new tasks without additional task-specific engineering.

C. Compatibility of POINT BRIDGE with Real Data

In this section, we study the effect of jointly training policies with simulated and real-world data. This paradigm, often called *co-training*, has been widely explored in sim-to-real [16] and human-to-robot transfer [19]. Our key findings are summarized below.

a) Co-training with real robot data further improves real-world performance.: We collect 45 teleoperated demonstrations on a real robot for three tasks and jointly train POINT BRIDGE with 1200 simulated demonstrations per task, using an 80–20 simulation-to-real ratio. Results across single-task (Table I) and multitask (Table II) show that adding real data consistently boosts performance by up to 30%. By comparison, image-based co-training methods yield a mixed outcome – likely because our simulation and real setups are not as visually aligned as in prior works that assume access to digital-cousin environments in simulation [16]. Overall, POINT BRIDGE outperforms image-based co-training by 61% in single-task and 66% in multitask settings, highlighting its ability to leverage small amounts of real data alongside large-scale simulation.

b) POINT BRIDGE *supports tasks involving soft and articulated objects.:* We evaluate single-task POINT BRIDGE policies on three real-world tasks involving soft and articulated objects: folding a towel (17/20 successes), closing a drawer (18/20), and placing a bowl in an oven (16/20). For each task, we collect 20 demonstrations via real robot teleoperation, without using any simulation data. Overall, POINT BRIDGE achieves an 85% success rate across these tasks, highlighting its effectiveness beyond rigid-object manipulation and serving as a proof-of-concept that the proposed representation also extends to deformable and articulated object manipulation in the real world.

IV. LIMITATIONS & CONCLUSION

In this work, we introduced POINT BRIDGE, a framework that employs domain-agnostic point-based representations to exploit synthetic simulation datasets, enabling zero-shot sim-to-real transfer with minimal visual alignment, supporting co-training with real data, and facilitating multitask policy learning. We recognize a few limitations of this work.

a) Limitations: (1) POINT BRIDGE depends on VLMs and other vision models, making it vulnerable to their failures; as these models advance, we expect corresponding improvements in robustness. (2) POINT BRIDGE requires camera pose alignment between simulation and reality to avoid distribution mismatch. A remedy is to train with diverse simulated viewpoints, which can be scaled via synthetic generation tools such as MimicGen [13]. (3) Point-based abstractions aid generalization but discard critical scene context, limiting performance in cluttered environments. Hybrid representations that preserve sparse contextual cues could address this gap. (4) Policies trained with POINT BRIDGE currently operate at a lower control frequency than image-based baselines due to the additional depth and point processing (Section III-A), which may hinder performance in highly dynamic settings where faster feedback is beneficial.

REFERENCES

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[4] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts *et al.*, "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023.

[5] Y. Liu, K. Zhang, Y. Li, Z. Yan, C. Gao, R. Chen, Z. Yuan, Y. Huang, H. Sun, J. Gao *et al.*, "Sora: A review on background, technology, limitations, and opportunities of large vision models," *arXiv preprint arXiv:2402.17177*, 2024.

[6] Y. Li, D. Choi, J. Chung, N. Kushman, J. Schrittwieser, R. Leblond, T. Eccles, J. Keeling, F. Gimeno, A. Dal Lago *et al.*, "Competition-level code generation with alphacode," *Science*, vol. 378, no. 6624, pp. 1092–1097, 2022.

[7] K. Goldberg, "Good old-fashioned engineering can close the 100,000-year "data gap" in robotics," p. eaea7390, 2025.

[8] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ Int'l Conf on Intelligent Robots and Systems*, 2012.

[9] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[10] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan, "Robogen: Towards unleashing infinite data for automated robot learning via generative simulation," *arXiv preprint arXiv:2311.01455*, 2023.

[11] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," *arXiv preprint arXiv:2406.02523*, 2024.

[12] M. Dalal, A. Mandlekar, C. Garrett, A. Handa, R. Salakhutdinov, and D. Fox, "Imitating task and motion planning with visuomotor transformers," *arXiv preprint arXiv:2305.16309*, 2023.

[13] 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.

[14] 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.

[15] 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.

[16] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev *et al.*, "Sim-and-real co-training: A simple recipe for vision-based robotic manipulation," *arXiv preprint arXiv:2503.24361*, 2025.

[17] A. Wei, A. Agarwal, B. Chen, R. Bosworth, N. Pfaff, and R. Tedrake, "Empirical analysis of sim-and-real cotraining of diffusion policies for planar pushing from pixels," *arXiv preprint arXiv:2503.22634*, 2025.

[18] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[19] S. Haldar and L. Pinto, "Point policy: Unifying observations and actions with key points for robot manipulation," *arXiv preprint arXiv:2502.20391*, 2025.

[20] Y. Zhu, A. Lim, P. Stone, and Y. Zhu, "Vision-based manipulation from single human video with open-world object graphs," *arXiv preprint arXiv:2405.20321*, 2024.

[21] V. Liu, A. Adeniji, H. Zhan, S. Haldar, R. Bhirangi, P. Abbeel, and L. Pinto, "Egozero: Robot learning from smart glasses," *arXiv preprint arXiv:2505.20290*, 2025.

[22] M. Lepert, J. Fang, and J. Bohg, "Phantom: Training robots without robots using only human videos," *arXiv preprint arXiv:2503.00779*, 2025.

[23] ——, "Masquerade: Learning from in-the-wild human videos using data-editing," *arXiv preprint arXiv:2508.09976*, 2025.

[24] V. Saxena, M. Bronars, N. R. Arachchige, K. Wang, W. C. Shin, S. Nasiriany, A. Mandlekar, and D. Xu, "What matters in learning from large-scale datasets for robot manipulation," *arXiv preprint arXiv:2506.13536*, 2025.

[25] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini *et al.*, "Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models," *arXiv e-prints*, pp. arXiv–2409, 2024.

[26] 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.

[27] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield, "Foundationstereo: Zero-shot stereo matching," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5249–5260.

[28] S. Haldar, Z. Peng, and L. Pinto, "Baku: An efficient transformer for multi-task policy learning," *arXiv preprint arXiv:2406.07539*, 2024.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[30] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," 2020.

[31] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[32] 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.

[33] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[34] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," in *European Conference on Computer Vision*. Springer, 2024, pp. 71–91.

[35] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht, "Cotracker: It is better to track together," 2023.

[36] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," *arXiv preprint arXiv:2210.11339*, 2022.

[37] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conference on Robot Learning*. PMLR, 2018, pp. 879–893.

[38] A. Iyer, Z. Peng, Y. Dai, I. Guzey, S. Haldar, S. Chintala, and L. Pinto, "Open teach: A versatile teleoperation system for robotic manipulation," *arXiv preprint arXiv:2403.07870*, 2024.

[39] Lightwheel, "Lightwheel sim-ready assets: High-quality usd assets for nvidia isaac sim," GitHub repository, 2025, 259 robotics simulation assets under CC BY-NC 4.0. [Online]. Available: https://github.com/LightwheelAI/Lightwheel-simready-asset