LEARNING TO GRASP ANYTHING BY PLAYING WITH RANDOM TOYS

Anonymous authors
Paper under double-blind review

ABSTRACT

Robotic manipulation policies often struggle to generalize to novel objects, limiting their real-world utility. In contrast, cognitive science suggests that children develop generalizable dexterous manipulation skills by mastering a small set of simple toys and then applying that knowledge to more complex items. Inspired by this, we study if similar generalization capabilities can also be achieved by robots. Our results indicate robots can learn generalizable grasping using **randomly assembled objects that are composed from just four shape primitives**—spheres, cuboids, cylinders, and rings. We show that training on these "toys" **enables robust generalization to real-world objects**, yielding strong zero-shot performance. Crucially, we find the key to this generalization is an object-centric visual representation induced by our proposed detection pooling mechanism. Evaluated in both simulation and on physical robots, our model achieves a 67% real-world grasping success rate on the YCB dataset, outperforming state-of-the-art approaches that rely on substantially more in-domain data. We further study how zero-shot generalization performance scales by varying the number and diversity of training toys and the demonstrations per toy. We believe this work offers a promising path to scalable and generalizable learning in robotic manipulation.

1 INTRODUCTION

"Treat nature by means of the cylinder, the sphere, the cone, everything brought into proper perspective."

PAUL CÉZANNE

Robotic manipulation policies have recently achieved impressive progress, solving complex tasks in domains such as dexterous manipulation (Kumar et al., 2016; Chen et al., 2022; Wang et al., 2024; Chen et al., 2023; Qin et al., 2021), robust sim-to-real transfer (Chukwurah et al., 2024; Pinel et al., 2023; Ho et al., 2020), and long-horizon planning for multi-step tasks (Mishra et al., 2023; Simeonov et al., 2020; Pertsch et al., 2020). Yet, a fundamental challenge remains: they often fail to generalize their manipulation skills to novel objects, limiting their practical application. In stark contrast, humans show astonishing generalization capabilities in dexterous manipulation. For example, cognitive literature (Schneiberg et al., 2002; Oztop et al., 2004; Rochat, 1989; Thelen et al., 1993; Needham et al., 2002; Ruff, 1984; Bonaiuto & Arbib, 2015) suggests that children can learn to grasp by mastering only a small set of simple toys and then applying that skill to unseen complex objects. This raises a central question: *can robotic manipulation policies generalize similarly?*

In this work, we demonstrate that robots can learn to grasp novel real-world objects when trained only on *randomly constructed toys*. The design of these toys is inspired by a classic insight from Cézanne: that complex objects can be deconstructed into a vocabulary of simple shape primitives. Specifically, we construct our toys as random compositions of just four shape primitives: spheres, cuboids, cylinders, and rings. These "Cézanne toys" preserve the structural essence of real objects while remaining sufficiently out-of-distribution, providing a challenging yet principled testbed for generalization. Trained on these random toys, our policy learns to grasp complex, unseen real-world objects in a zero-shot manner. See Figure 1 for an overview.

The key to this generalization capability, as we empirically show, lies in the usage of *object-centric visual representations*. Specifically, we introduce detection pooling (DetPool) to obtain an object-

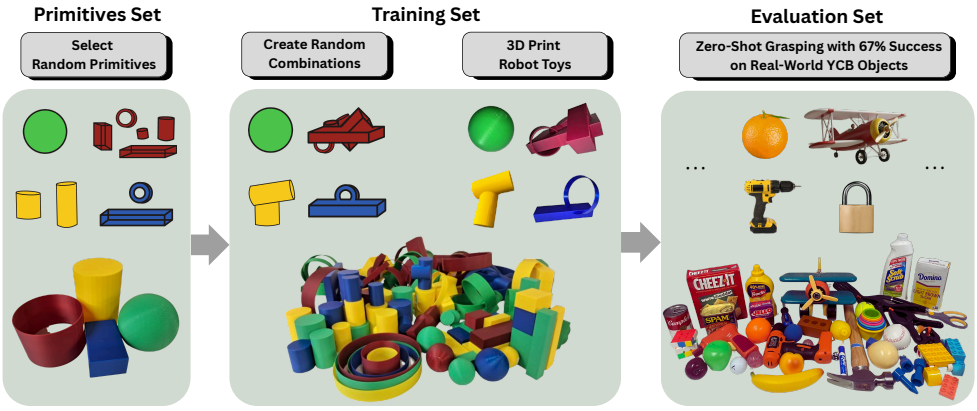

Figure 1: Our grasping policy, trained exclusively on random toy compositions (middle) built from just four basic primitives (left), zero-shot generalizes to real-world objects (right) and achieves an 67% success rate on 64 objects from the YCB dataset.

centric visual representation. This method first uses a mask of the target object to constrain the vision encoder's attention to the object region, and then applies mean pooling on the output features corresponding to the object patches. In this way, we ensure the final vision representation only contains information about the object and not the background or other distractors. We find this visual representation is the key to enable a grasping policy to generalize between the vastly different objects in training and testing. We name our framework LEGO (***LE**arning to **G**rasp from t**O**ys*).

To evaluate our model's generalization capabilities, we conduct a comprehensive experimental evaluation. First, we test its zero-shot performance: trained on a small dataset of 250 "Cézanne toys" with 1,500 demonstrations, our policy achieves a 67% success rate on 64 real-world YCB (Calli et al., 2015) objects, significantly outperforming larger, state-of-the-art models like OpenVLA-OFT (Kim et al., 2025) and π_0-FAST (Black et al., 2024; Pertsch et al., 2025) that are pretrained on much more data. Second, detailed ablations confirm that the key to this success is the object-centric representation induced by our DetPool mechanism, which significantly outperforms standard pooling baselines. Furthermore, we conduct thorough scaling experiments, finding that the zero-shot generalization performance scales with both toy diversity and the number of demonstrations, with the latter being more critical. Finally, we show this generalization capability is robust across robot diverse embodiments, including simple grippers and dexterous hands.

2 Related Work

Cognitive Approaches for Manipulation. Developmental psychology has long been studying how infants acquire manipulation skills through exploration and practice (Thelen et al., 1993; Schneiberg et al., 2002; Needham et al., 2002). Early works (Ruff, 1984; Rochat, 1989; Yoshida & Smith, 2008) show that infants gradually learn manipulation skills by focusing on increasingly diverse object features such as shape, texture, and weight. Rakison & Butterworth (1998) demonstrate that infants generalize their manipulation skills to unseen objects by applying learned actions with familiar parts to the novel objects. Motivated by this literature, we explore whether robotic manipulation can achieve a similar level of generalization to unseen objects.

Existing approaches have explored infant-inspired learning as a foundation for modeling objects (Farhadi et al., 2009), either through descriptive attributes (Cohen et al., 2019; Sun et al., 2013), explicit segmentation (Liu et al., 2024; Li et al., 2024a;b; Vahrenkamp et al., 2016; Aleotti & Caselli, 2011), or representing objects as 3D primitives (Tulsiani et al., 2016; Monnier et al., 2023; Lin et al., 2025). Our work builds on these ideas and explores whether generalized object representations can emerge from just a few primitives.

Generalization in Robotic Manipulation. Robotic manipulation models have shown capability of mastering various real-world tasks (Zhao et al., 2023; Fu et al., 2024; Barreiros et al., 2025).

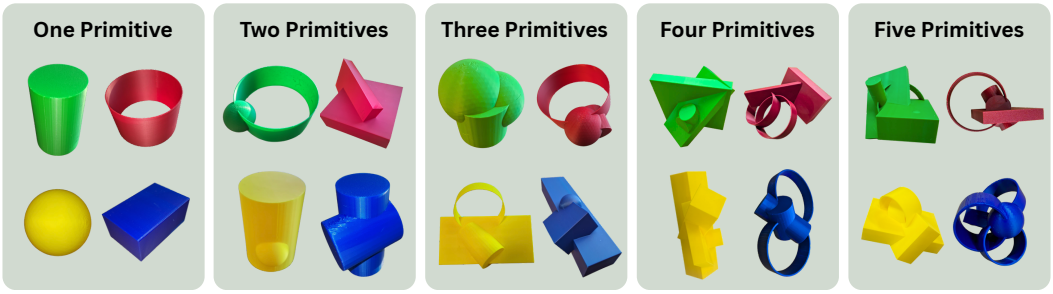

Figure 2: **Our Cézanne toys are composed of different number of primitives.** We generate each toy by randomly assembling 1-5 primitives and randomizing dimensions and colors.

However, they are often trained with a limited set of objects and environments and generalize poorly to new ones. One common approach to address this is through scaling up the training data (Brohan et al., 2022; Zitkovich et al., 2023; Intelligence et al., 2025; Eppner et al., 2021; Fang et al., 2020; Ye et al., 2025). In contrast, we show that strong generalization capabilities are still achievable even with training data of a couple of hours. Another line of works improves the generalizability of robotic model through heavy data augmentation (Hansen & Wang, 2021; Tobin et al., 2017; Sadeghi & Levine, 2016) while we achieve strong generalization without any augmentation. Other works improve the generalizability by learning better visual representations (Burns et al., 2023; Srirama et al., 2024). Compared to these approaches which usually require costly visual pretraining, we improve generalizability by obtaining object-centric visual representations via DetPool, which is light-weight and can be applied to any pretrained vision model.

Object-Centric Vision Models. Object-centric models are shown to improve performance and robustness in computer vision. Methods like Burgess et al. (2019); Engelcke et al. (2019); Locatello et al. (2020) learn object-centric representations from 2D images, typically for scene decomposition. Extending object-centric learning to the temporal domain, Herzig et al. (2021) introduce an object-region transformer that learns temporally coherent object-level features across video frames for tasks such as action recognition and dynamics prediction. Other approaches extend this idea into the 3D domain via world models (Ferraro et al., 2023; Jeong et al., 2025). Unlike these works, our approach focuses on the effect of object-centric representations on generalizable robotic manipulation.

When it comes to manipulation, a variety of object-centric grasping approaches have been explored (Chen et al., 2024; Zurbrügg et al., 2024; Mandikal & Grauman, 2020). Papers such as Devin et al. (2017) address generalizable robot learning through object-centric methods by using attention mechanisms. However, they only evaluate the generalization between similar objects and do not explore the limit of generalization between vastly different objects such as random toys and real objects. The most similar approach to ours is OTTER (Huang et al., 2025), which uses the vision-language attention map in CLIP to obtain object-centric visual representations. However, it is only limited to CLIP, while our method can be applied to any vision transformer.

3 A Cézanne Toy Grasping Dataset

To evaluate the generalization capabilities of robotic grasping policies, we explore a challenging zero-shot setting: training policies exclusively on a set of out-of-distribution (OOD) objects and testing on common real-world objects. To this end, we develop a systematic approach for generating a diverse set of random, OOD objects. We draw inspiration from Cézanne's classic idea that complex objects can be abstracted into compositions of simple shape primitives. We thus generate our training objects by randomly combining these primitives. This process efficiently creates a training set of "Cézanne toys" composed of random primitives, which ensures they are OOD, while still retaining structural properties that enable generalization. An overview of this process is presented in Figure 1. Next, we detail our primitives' designs, the toy generation process, and the resulting grasping dataset.

Designing the Primitives. Inspired by prior literature (Marr & Nishihara, 1978; Tulsiani et al., 2016; Li et al., 2019), we choose four primitive types: spheres, cuboids, cylinders, and rings (See

the left column of Figure 2). The primitive's scale is randomized within specific ranges. Cuboids range from 2–7.2 cm in width, 1–20 cm in height, and 2–28 cm in length; spheres range from 1–8 cm in diameter; cylinders range from 4–7 cm in diameter and 4–12 cm in height; and rings range from 6–20 cm in diameter, 0.6–1.8 cm in wall thickness, and 2–6 cm in height.

Generating Cézanne Toys. We generate Cézanne toys by randomly combining the primitives. Figure 2 illustrates some examples of the generated toys. We start by choosing a random number of primitives, ranging from 1 to 5. We then randomly choose the corresponding number of primitives from the four basic types and the dimensions of each instance are randomized. The sampled primitives are then sequentially assembled to form the final toy. Specifically, the first primitive is placed at the origin, and the centroid of each subsequent primitive is randomly positioned within a previous primitive. This ensures the primitives overlap and form a coherent structure rather than scattered components. Each primitive is also assigned a random 3D rotation. Finally, the toy is randomly assigned one of four colors: blue, red, green or yellow. By repeating this process, we generate a training set of 250 diverse toys, including 27 made of two primitives, 35 of three, 38 of four, and 47 of five, as well as individual primitives such as 46 cuboids, 18 balls, 20 cylinders, and 19 rings. All toys are both simulated and 3D printed for grasping data collection.

Collecting Grasping Data. We collect toy grasping trajectories in both simulation and real. In simulation, we use ManiSkill (Tao et al., 2025) with a Franka arm and gripper; in the real world, we use the same Franka arm with a Robotiq gripper, as well as a Unitree H1-2 humanoid equipped with Inspire RH56DFTP hands. We collect all data via teleoperation, except for grasping individual primitives in simulation, which is performed using motion planning. During collection, we ensure a diverse set of grasping poses per object, since individual objects can be grasped in many different ways. We collect 2,500 trajectories in simulation, 1,500 on the real Franka, and 500 on the H1-2.

4 THE LEGO METHOD

To enable a policy trained on our "Cézanne toys" to generalize to real-world objects, we introduce a novel object-centric approach. Our method's key distinction from full-scene architectures is its use of a detection pooling mechanism to obtain an object-centric visual representation, which we empirically show is the key to robust generalization. This section details our including preliminaries (Section 4.1), full architecture (Section 4.2), and the detection pooling method (Section 4.3).

4.1 PRELIMINARIES

Robotic Tasks. Robotic tasks can be represented as temporal sequences of observations and actions. The observations typically consist of visual observations $i_{1:T}$ and proprioceptive states $s_{1:T}$, where T is the episode length, $i_t \in \mathbb{R}^{N\times H\times W\times 3}$ denotes the images captured by N cameras at time step t, and $s_t \in \mathbb{R}^{d_s}$ is the proprioception (e.g., the joint positions) of the robot at step t. The actions $a_{1:T} \in \mathbb{R}^{d_a}$ represent how robot commands its joints (e.g., target joint positions) at step t.

Policy Learning. The objective is to learn a policy that maps a history of the past C steps—visual inputs $i_{t-C+1:t}$ and robot states $s_{t-C+1:t}$—to a future action sequence of length K, in order to successfully complete the task: $\pi(i_{t-C+1:t}, s_{t-C+1:t}) \rightarrow a_{t:t+K-1}$.

4.2 ARCHITECTURE

Below, we describe the different components of our policy architecture as well as the training objective. Figure 3 (a) illustrates the overall LEGO architecture.

Vision Encoder. Given the observations $i_{t-C+1:t}$ and $s_{t-C+1:t}$ from past C steps, our model uses a vision encoder to encode each set of visual observations i_t into visual embeddings $e_t^{1:N}$, where $e_t^n \in \mathbb{R}^{d_e}$ is the embedding of the n-th camera image at step t, and d_e is the hidden dimension of the vision encoder. We use a pretrained MVP (Xiao et al., 2022) as the vision encoder. These resulting features are then input into a transformer-based architecture for further processing.

Transformer Policy. We use a transformer-based architecture as our main policy network. It first concatenates the visual embeddings $e_t^{1:N}$ and the proprioception s_t along the channel dimension into a single token, and then projects it with an MLP. The transformer backbone then takes the projected

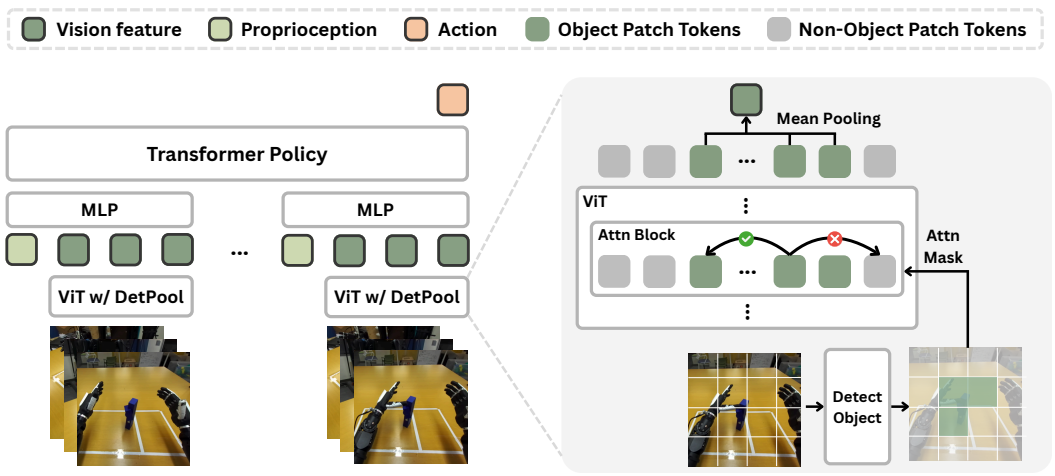

Figure 3: **The LEGO architecture with DetPool. (a)** LEGO uses a ViT with DetPool to extract features of the target object and uses a transformer to predict future actions based on the visual features and the proprioception. **(b)** The ViT extracts features that focus on the target object via DetPool which restrains the attention to the object patches using an attention mask and performs mean pooling on the output object patch tokens to get the final object-centric vision feature.

tokens from all past C steps and predicts the concatenated actions $a_{t:t+K-1}$ for the next K steps from the last token. The transformer is designed to have the same size as a ViT-B (Dosovitskiy et al., 2021) to get the best performance (see the ablation in Section 5.5).

Training Objective. Following the regular behavior cloning algorithm, the training loss is the mean ℓ_1 loss between the predicted actions $\hat{a}_{t:t+K-1}$ and the ground-truth actions $a_{t:t+K-1}$, i.e., $\mathcal{L} = \frac{1}{Kd_a}\|\hat{a}_{t:t+K-1} - a_{t:t+K-1}\|_1$.

To learn a policy that generalizes to novel objects, we design the vision encoder to be object-centric via detection pooling, which we introduce in the next section.

4.3 Detection Pooling

We design a detection pooling mechanism in the vision encoder such that the extracted visual feature is focused on the object to be grasped, as shown in Figure 3 (b). Specifically, we first obtain the object segmentation mask for each frame using SAM 2 (Ravi et al., 2024b). We then use the object mask to set the attention mask in the vision encoder such that there is no attention between object patch tokens and non-object patch tokens. In this way, we ensure that the object patch tokens only contain features from the object itself while ignoring features from non-object patch tokens. Note that this method still allows the vision encoder to understand where the object is in the scene due to the use of positional embeddings. At the end of the vision encoder, we obtain the object-centric visual feature by applying mean pooling on the object patch tokens, which is the final visual embedding we use for the policy model. We empirically find that DetPool is crucial for achieving strong zero-shot generalization compared to other pooling methods such as mean and attention pooling that do not restrict the attention mask within the ViT and only pool the final output tokens (Section 5.2).

5 Experiments

We evaluate LEGO on the YCB object benchmark (Calli et al., 2015) using the ManiSkill simulator (Tao et al., 2025). For comparison, we include vision-language-action (VLA) models such as π_0-FAST and OpenVLA-OFT, which aim to generalize through large-scale pretraining. We further analyze how performance scales with the number of unique toys and demonstrations. Beyond simulation, we test LEGO on two real-world setups: a 7-DoF Franka Emika Panda with a 1-DoF

Table 1: **Results of zero-shot grasping in simulation.** We compare our model with state-of-the-art models (OpenVLA-OFT and π_0-FAST) finetuned on our dataset in simulation, as well as different pooling baselines. Our model outperforms the finetuned baselines in simulation, with our DetPool proving key to generalization by boosting performance 22-48% over other pooling baselines.

Method	# Demos					
	250	500	1000	1500	2000	2500
OpenVLA-OFT (Kim et al., 2025)	30.10	36.35	22.31	15.38	14.71	12.79
π_0-FAST (Black et al., 2024)	8.85	7.60	7.69	8.56	4.23	4.13
Ours - Attn Pooling	34.71	40.10	44.23	48.27	49.81	51.63
Ours - CLS Pooling	24.71	20.29	36.92	41.44	42.40	49.81
Ours - Mean Pooling	32.98	30.38	36.15	39.90	40.29	40.58
Ours - Det Pooling	**56.63**	**68.17**	**71.15**	**74.62**	**76.83**	**80.00**

Robotiq 2F-85 adaptive gripper, where evaluation is done on the YCB benchmark, and an Unitree H1-2 humanoid with Inspire dexterous hands, evaluated on a 13-object set of everyday items. We include demonstration videos in the supplementary materials that show our collected data and the corresponding evaluation settings.

5.1 Implementation Details

Model and Training Setup. LEGO is implemented using PyTorch (Paszke et al., 2019). Its architecture consists of a ViT-L encoder from MVP (Xiao et al., 2022) for feature extraction and a ViT-Base transformer backbone. The policy is conditioned on a history of $C = 16$ timesteps to predict $K = 16$ future actions. For our DetPool mechanism, we use SAM 2 (Ravi et al., 2024a) to obtain object masks for real-world images and use ground-truth masks in simulation. The model is trained on eight NVIDIA A6000 GPUs and evaluated on a single A6000.

State and Action Parameterization. We parameterize the proprioceptive space using the joint angles of the robot arm used and a continuous gripper state (when applicable). This yields an 8-dimensional vector for the Franka setup, and a 40-dimensional vector for our H1-2 setup (which includes feedforward torques and finger joints). The model then conditions on state vectors from past timesteps and predicts action vectors for future timesteps, where state and action vectors are represented using absolute joint angles, rather than relative (delta) angles.

5.2 Simulation Evaluation

Experimental Setup. Our training set contains 2,500 demonstrations, comprising 10 successful grasps for each of our 250 unique toys. To analyze scaling laws, all models are also trained on subsets of this data. For evaluation, we use a set of 65 graspable objects from the YCB benchmark, selecting only those feasibly graspable by the Franka robot; each object is tested 16 times on a predefined grid, and we report the mean success rate across all trials.

Baselines. We compare LEGO (86M parameters) against two significantly larger, state-of-the-art VLAs that rely on large-scale pretraining: π_0-FAST (3B) (Black et al., 2024) and OpenVLA-OFT (7B) (Kim et al., 2025). Both models are fine-tuned on the same data as ours. To validate the contribution of our core DetPool mechanism, we also conduct ablation studies, replacing DetPool with standard alternatives like attention pooling, CLS pooling, and mean pooling.

Results. Our simulation results, summarized in Table 1, highlight the superior generalization and scalability of LEGO compared to baselines. While LEGO's performance scales reliably with more data—achieving a top success rate of 80% with 2,500 demonstrations, the state-of-the-art VLA baselines falter. We find that π_0-FAST is too data-hungry for the small dataset and struggles with a real-to-sim domain gap from its pretraining. Similarly, OpenVLA-OFT shows initial promise with 250–500 demonstrations but quickly overfits as more data is added, causing its performance to deteriorate. On the other hand, while attention pooling is the strongest baseline, it is still significantly outperformed by our DetPool mechanism. In contrast, DetPool enables robust and scalable generalization, underscoring the effectiveness of object-centric visual representation for generalizability.

Table 2: **Zero-shot grasping results on the real Franka robot.** We compare our model against ShapeGrasp, OpenVLA-OFT (finetuned), and state-of-the-art π_0-FAST (zero-shot and finetuned). Our model achieves a 66.67% success rate, outperforming all baselines except finetuned π_0-FAST.

Method	Pretraining	Tuned on Toys	# Parameters	# Demos / 1500
OpenVLA-OFT (Kim et al., 2025)	OXE	✔	7B	9.47
π_0-FAST (Black et al., 2024)	π Dataset + 75K DROID	✘	3B	61.82
π_0-FAST (Black et al., 2024)	π Dataset + 75K DROID	✔	3B	**76.56**
ShapeGrasp (Li et al., 2024b)	GPT4o	✘	-	26.56
Ours	✘	✔	86M	66.67

5.3 Franka Robot Evaluation

Experimental Setup. For real-world experiments, we use a 7-DoF Franka Emika Panda arm with a Robotiq 2F-85 gripper, consistent with the DROID benchmark. We 3D-print the 250 toys with the highest simulated success rates in simulation and collect 1,500 successful grasp demonstrations. All models are then evaluated on a test set of 64 YCB objects. Following the simulation protocol, each object is tested 16 times on a predefined grid, and we report the mean success rate.

Baselines. We compare LEGO with strong baselines. π_0-FAST (Black et al., 2024) is a state-of-the-art VLA model trained on in-domain data from the DROID setting as well as a large-scale robotics dataset. OpenVLA-OFT (Kim et al., 2025) is a 7B-parameter VLA model pretrained on the Open-X Embodiment (OXE) dataset (Collaboration et al., 2023). ShapeGrasp (Li et al., 2024b) is a training-free, LLM-based approach that uses pretrained language models to decompose objects geometrically before selecting a graspable part.

A common theme across these methods is reliance on large-scale pretraining: either extensive in-domain robot–object interaction data or internet-scale multimodal data (for ShapeGrasp). In contrast, LEGO is trained from scratch using only 2,500 demonstrations, yet achieves competitive performance despite being orders of magnitude smaller in both dataset size and model scale.

Results. As shown in Table 2, LEGO achieves the second-best performance among all models tested, highlighting the effectiveness of our approach. It outperforms OpenVLA-OFT, a large pretrained VLA model; ShapeGrasp, which leverages internet-scale multimodal data via an LLM; and π_0-FAST in its zero-shot setting, trained on 75K in-domain DROID grasping examples. This also demonstrates the strength of our object-centric representation, as LEGO attains superior performance using far less data and a smaller model architecture.

The finetuned version of π_0-FAST achieves the best overall performance, which we hypothesize is because finetuning on additional demonstrations from our DROID setup allows it to utilize its pretrained knowledge and adapt to the specific lighting and physical environment, improving performance. In contrast, OpenVLA-OFT is less effective. We observed that minor inaccuracies frequently caused grasp failures, indicating difficulty in generalizing to novel objects and settings.

5.4 H1-2 Dexterous Hands Evaluation

Experimental Setup. We also perform real-world experiments with the Unitree H1-2, a humanoid robot with 27 degrees of freedom (DoFs). Each 7-DoF arm is equipped with a 6-DoF Inspire RH56DFTP hand, which has 12 total joints. The 6 DoFs capture the independent motions of the thumb and fingers, while the 12 joints result from each DoF being implemented as a pair of mechanically linked joints driven by a single linear servo. This hand design mimics human-like dexterity more closely than traditional gripper end-effectors, making it well-suited for experiments requiring fine-grained grasping.

We evaluate our model and the baselines on 13 everyday objects using the left arm and hand. Each object is tested five times across a predefined grid, and each trial is scored in the same manner as in the Franka experiments.

Table 3: **Results on H1-2 humanoid robot.** We compare our model with state-of-the-art models (π_0-Fast and OpenVLA-OFT) that are finetuned on our data. We show our model achieves superior performance without any pretraining on real objects.

Method	Bell Pepper	Pink Cube	Baton Cookies	Solder Coil	Tomato	Pink Ribbed Ball	Piggles Stuffed Toy
OpenVLA-OFT	0	0	40	20	20	0	40
π_0-FAST	20	20	0	20	20	20	40
Ours	**60**	**40**	**60**	**40**	**60**	**60**	**60**

	Mike Wazowski Stuffed Toy	Red Tape Dispenser	Red Solo Cup	Paper Towel Roll	Hand Sanitizer	Yo-Yo	Average
OpenVLA-OFT	60	0	0	60	0	0	18.46
π_0-FAST	60	40	40	40	20	0	26.15
Ours	**60**	**60**	**20**	**60**	**60**	**20**	**50.77**

Baselines. We compare LEGO with OpenVLA-OFT and π_0-FAST (see Section 5.3 for details).

Results. As shown in Table 3, LEGO achieves the highest success rate of 50.77% in a more challenging setting than the Franka DROID experiments, despite being trained from scratch with only 500 demonstrations. In contrast, π_0-FAST struggles due to limited demonstrations and the likely absence of this embodiment in its pretraining data. OpenVLA-OFT also underperforms, having been pretrained on robot arm and gripper data without humanoid or dexterous-hand examples. These results underscore LEGO 's data efficiency and the role of DetPool in enabling robust generalization.

5.5 Ablation Studies

We present our ablation studies below, conducted within the ManiSkill simulation environment, which offers a stable and reproducible setting for rigorously evaluating the impact of various factors.

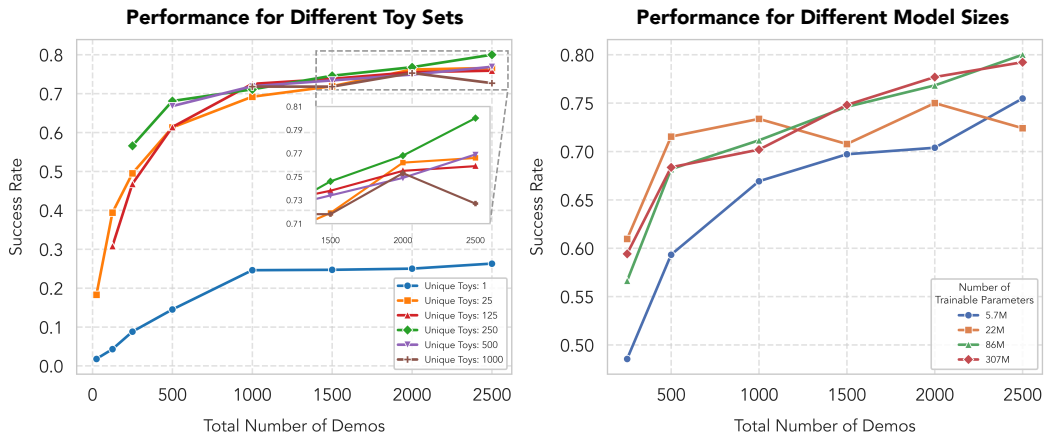

Figure 4: **Scaling studies. Left**: The zero-shot success rate scales with both the number of demos and the number of unique toys. We also find that once the number of demos is sufficient, 25 toys is already enough to achieve a robust zero-shot transfer. **Right**: The performances scales with the size of the policy transformer until it saturates at the size of 86M.

Effect of Number and Diversity of Demonstrations. We perform an ablation study to examine how both the number of unique toys in the training set and the number of grasping demonstrations influence performance. Specifically, we construct six object sets containing 1, 25, 125, 250, 500, and 1000 unique toys, respectively. For each set, we collect 2,500 grasping demonstrations and train our model using varying numbers of demonstrations per set. The results, shown in the left panel of Figure 4, indicate that increasing the number of unique objects improves performance, but with diminishing returns. In contrast, the number of demonstrations has a stronger impact on learning generalizable grasping, a result consistent with findings from cognitive science literature.

Effect of Model Size. To investigate how the size of the policy's transformer backbone affects performance, we conduct an ablation study. Using the 250-object set—which yields the best overall performance—we vary the transformer's size and evaluate the policy across different numbers of demonstrations. The results, shown in the right panel of Figure 4, indicate that ViT-Base is the best overall choice: it matches or slightly surpasses ViT-Large in performance while being significantly smaller and thus allowing for faster inference.

Importance of Individual Primitives. To assess the relative importance of each of the four primitives, we conduct an ablation study in which the training set excludes toys containing a given primitive. For each case, the model is trained with varying numbers of demonstrations. The results, presented in Table 4, show that the sphere is the most critical primitive, as its exclusion results in the largest performance degradation. In contrast, the ring and cylinder appear to be less important, with a relatively small performance drop when they are omitted.

Effect of Toy Complexity. We perform an ablation study to measure the relationship between toy complexity, quantified by the number of constituent primitives, and model performance. The model is trained on demonstrations containing toys with 2-5 primitives. As shown in Table 5, toys with two primitives contribute the most to performance, while toys with five primitives are still beneficial but less influential. This is likely due to the evaluation set's distribution of sizes, which contains more toys with two or three primitives; highly complex toys with five primitives are relatively rare.

Table 4: **Ablation of primitive types.** We study the importance of each primitive type by removing each one out of the primitive set.

Primitive Removed	100	200	500	1000
Cuboid	**37.88**	56.35	65.38	72.12
Sphere	44.13	**47.31**	**61.83**	**63.08**
Ring	44.23	**67.5**	68.56	**72.6**
Cylinder	**45.29**	57.6	**69.52**	72.31

Table 5: **Ablation of toy complexity.** We study the importance of each toy complexity level by training polices only on toys composed of a certain number of primitives.

Toy Complexity	25	125	250
Two Primitives	**9.04**	**32.6**	**44.42**
Three Primitives	7.31	15.77	23.17
Four Primitives	7.69	12.4	23.36
Five Primitives	4.32	10.87	10.19

6 Conclusion

In this work, we demonstrate that robots can acquire robust general-purpose grasping skills by learning from a simple set of objects composed from just four basic shape primitives: spheres, cuboids, cylinders, and rings. We show that training on these toys enables a policy to generalize to a wide range of real-world objects. Our method learns an object-centric visual representation using a detection pooling and transformer architecture, and is trained on a dataset of 250 toys with 1,500 demonstrations in the real Franka setting. This policy achieves a 67% zero-shot success rate on the YCB dataset, outperforming state-of-the-art models such as π_0-FAST and OpenVLA-OFT despite them being trained on more diverse and larger datasets. Our findings on grasping scaling laws highlight how we can efficiently optimize performance with limited data. Ultimately, this work demonstrates a scalable path to robotic manipulation by showing that real-world grasping generalization can emerge from learning on object composites of a few primitive shapes. We believe this work offers a promising path to scalable and generalizable learning in robotic manipulation.

7 Limitations and Future Work

While we show that our method offers a promising path toward generalized grasping, it is important to acknowledge its limitations to guide future research. One key limitation is the diversity of the training domain: the model's performance may degrade on objects with different physical properties. Furthermore, our current work focuses on simple, single-step grasping. Future work could extend this approach to complex, long-horizon tasks such as cloth folding and manipulation in dynamic scenes. Finally, the computational cost of the model's architecture presents a challenge for real-world deployment on resource-constrained hardware, pointing toward a need for future optimization.

REFERENCES

Jacopo Aleotti and Stefano Caselli. Manipulation planning of similar objects by part correspondence. *2011 15th International Conference on Advanced Robotics (ICAR)*, pp. 247–252, 2011. URL https://api.semanticscholar.org/CorpusID:18127179.

Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-Hsin Fang, Kunimatsu Hashimoto, Muhammad Zubair Irshad, Masha Itkina, et al. A careful examination of large behavior models for multitask dexterous manipulation. *arXiv preprint arXiv:2507.05331*, 2025.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi 0$: A vision-language-action flow model for general robot control. *ArXiv*, abs/2410.24164, 2024. URL https://api.semanticscholar.org/CorpusID:273811174.

James J. Bonaiuto and Michael A. Arbib. Learning to grasp and extract affordances: the integrated learning of grasps and affordances (ilga) model. *Biological Cybernetics*, 109:639 – 669, 2015. URL https://api.semanticscholar.org/CorpusID:17366201.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew M. Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *ArXiv*, abs/1901.11390, 2019. URL https://api.semanticscholar.org/CorpusID:59523721.

Kaylee Burns, Zach Witzel, Jubayer Ibn Hamid, Tianhe Yu, Chelsea Finn, and Karol Hausman. What makes pre-trained visual representations successful for robust manipulation? *arXiv preprint arXiv:2312.12444*, 2023.

Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pp. 510–517. IEEE, 2015.

Hanzhi Chen, Binbin Xu, and Stefan Leutenegger. Funcgrasp: Learning object-centric neural grasp functions from single annotated example object. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1900–1906, 2024. URL https://api.semanticscholar.org/CorpusID:267547826.

Yuanpei Chen, Yaodong Yang, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuan Jiang, Stephen Marcus McAleer, Yiran Geng, Hao Dong, Zongqing Lu, and Song-Chun Zhu. Towards human-level bimanual dexterous manipulation with reinforcement learning. *ArXiv*, abs/2206.08686, 2022. URL https://api.semanticscholar.org/CorpusID:249848184.

Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuan Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:2804–2818, 2023. URL https://api.semanticscholar.org/CorpusID:265801841.

Naomi Chukwurah, Abiodun Sunday Adebayo, and Olanrewaju Oluwaseun Ajayi. Sim-to-real transfer in robotics: Addressing the gap between simulation and real-world performance. *Journal of Frontiers in Multidisciplinary Research*, 2024. URL https://api.semanticscholar.org/CorpusID:276984724.

Vanya Cohen, Benjamin Burchfiel, Thao Nguyen, Nakul Gopalan, Stefanie Tellex, and George Dimitri Konidaris. Grounding language attributes to objects using bayesian eigenobjects. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1187–1194, 2019. URL https://api.semanticscholar.org/CorpusID:170078764.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

Coline Devin, P. Abbeel, Trevor Darrell, and Sergey Levine. Deep object-centric representations for generalizable robot learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7111–7118, 2017. URL https://api.semanticscholar.org/CorpusID:33758357.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *ArXiv*, abs/1907.13052, 2019. URL https://api.semanticscholar.org/CorpusID:198986015.

Clemens Eppner, Arsalan Mousavian, and Dieter Fox. Acronym: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6222–6227. IEEE, 2021.

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11444–11453, 2020.

Ali Farhadi, Ian Endres, Derek Hoiem, and David Alexander Forsyth. Describing objects by their attributes. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1785, 2009. URL https://api.semanticscholar.org/CorpusID:14940757.

Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and B. Dhoedt. Focus: Object-centric world models for robotics manipulation. *ArXiv*, abs/2307.02427, 2023. URL https://api.semanticscholar.org/CorpusID:259342267.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13611–13617. IEEE, 2021.

Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3138–3149, 2021. URL https://api.semanticscholar.org/CorpusID:238744000.

Daniel Ho, Kanishka Rao, Zhuo Xu, Eric Jang, Mohi Khansari, and Yunfei Bai. Retinagan: An object-aware approach to sim-to-real transfer. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10920–10926, 2020. URL https://api.semanticscholar.org/CorpusID:226278453.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Huang Huang, Fangchen Liu, Letian Fu, Tingfan Wu, Mustafa Mukadam, Jitendra Malik, Ken Goldberg, and Pieter Abbeel. Otter: A vision-language-action model with text-aware visual feature extraction. *arXiv preprint arXiv:2503.03734*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $pi_0.5$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Youngjoon Jeong, Junha Chun, Soonwoo Cha, and Taesup Kim. Object-centric world model for language-guided manipulation. *ArXiv*, abs/2503.06170, 2025. URL https://api.semanticscholar.org/CorpusID:276903201.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, P Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Ye Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sung Yul Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng

Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean-Pierre Mercat, Abdul Rehman, Pannag R. Sanketi, Archit Sharma, C. Blake Simpson, Quang Uyen Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Da Ling Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosa Maria Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J. Lim, Jitendra Malik, Roberto Mart'in-Mart'in, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. *ArXiv*, abs/2403.12945, 2024. URL https://api.semanticscholar.org/CorpusID:268531351.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *ArXiv*, abs/2406.09246, 2024. URL https://api.semanticscholar.org/CorpusID:270440391.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 378–383, 2016. URL https://api.semanticscholar.org/CorpusID:7586242.

Haosheng Li, Weixin Mao, Weipeng Deng, Chenyu Meng, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Hongan Wang, and Xiaoming Deng. Seggrasp: Zero-shot task-oriented grasping via semantic and geometric guided segmentation. *ArXiv*, abs/2410.08901, 2024a. URL https://api.semanticscholar.org/CorpusID:273323195.

Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2652–2660, 2019.

Samuel Li, Sarthak Bhagat, Joseph Campbell, Yaqi Xie, Woojun Kim, Katia P. Sycara, and Simon Stepputtis. Shapegrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10527–10534, 2024b. URL https://api.semanticscholar.org/CorpusID:268723780.

Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv preprint arXiv:2502.20396*, 2025.

Weiyu Liu, Jiayuan Mao, Joy Hsu, Tucker Hermans, Animesh Garg, and Jiajun Wu. Composable part-based manipulation. *ArXiv*, abs/2405.05876, 2024. URL https://api.semanticscholar.org/CorpusID:265154271.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *ArXiv*, abs/2006.15055, 2020. URL https://api.semanticscholar.org/CorpusID:220127924.

Priyanka Mandikal and Kristen Grauman. Learning dexterous grasping with object-centric visual affordances. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6169–6176, 2020. URL https://api.semanticscholar.org/CorpusID:233439776.

David Marr and Herbert Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200(1140):269–294, 1978.

Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:261685884.

Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei A. Efros, and Mathieu Aubry. Differentiable blocks world: Qualitative 3d decomposition by rendering primitives. *ArXiv*, abs/2307.05473, 2023. URL https://api.semanticscholar.org/CorpusID:259766537.

Amy Work Needham, Tracy M. Barrett, and Karen Peterman. A pick-me-up for infants' exploratory skills: Early simulated experiences reaching for objects using 'sticky mittens' enhances young infants' object exploration skills. *Infant Behavior & Development*, 25:279–295, 2002. URL https://api.semanticscholar.org/CorpusID:16427970.

Erhan Oztop, Nina S. Bradley, and Michael A. Arbib. Infant grasp learning: a computational model. *Experimental Brain Research*, 158:480–503, 2004. URL https://api.semanticscholar.org/CorpusID:8738077.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Karl Pertsch, Oleh Rybkin, Frederik Ebert, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. Long-horizon visual planning with goal-conditioned hierarchical predictors. *ArXiv*, abs/2006.13205, 2020. URL https://api.semanticscholar.org/CorpusID:219981151.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Ricardo Garcia Pinel, Robin Strudel, Shizhe Chen, Etienne Arlaud, Ivan Laptev, and Cordelia Schmid. Robust visual sim-to-real transfer for robotic manipulation. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 992–999, 2023. URL https://api.semanticscholar.org/CorpusID:260315942.

Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, 2021. URL https://api.semanticscholar.org/CorpusID:236986915.

Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. In *Conference on Robot Learning*, pp. 683–693. PMLR, 2023.

David H. Rakison and George Butterworth. Infants' use of object parts in early categorization. *Developmental Psychology*, 34:49–62, 1998. URL https://api.semanticscholar.org/CorpusID:210399125.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024a. URL https://arxiv.org/abs/2408.00714.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024b.

Philippe Rochat. Object manipulation and exploration in 2- to 5-month-old infants. *Developmental Psychology*, 25:871–884, 1989. URL https://api.semanticscholar.org/CorpusID:197658959.

Holly A. Ruff. Infants' manipulative exploration of objects: Effects of age and object characteristics. *Developmental Psychology*, 20:9–20, 1984. URL https://api.semanticscholar.org/CorpusID:201316353.

Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

Sheila Schneiberg, Heidi Sveistrup, Bradford J. McFadyen, P. Mckinley, and Mindy F. Levin. The development of coordination for reach-to-grasp movements in children. *Experimental Brain Research*, 146:142–154, 2002. URL https://api.semanticscholar.org/CorpusID:15714879.

Anthony Simeonov, Yilun Du, Beomjoon Kim, Francois Robert Hogan, Joshua B. Tenenbaum, Pulkit Agrawal, and Alberto Rodriguez. A long horizon planning framework for manipulating rigid pointcloud objects. In *Conference on Robot Learning*, 2020. URL https://api.semanticscholar.org/CorpusID:226964745.

Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. Hrp: Human affordances for robotic pre-training. *arXiv preprint arXiv:2407.18911*, 2024.

Yuyin Sun, Liefeng Bo, and Dieter Fox. Attribute based object identification. *2013 IEEE International Conference on Robotics and Automation*, pp. 2096–2103, 2013. URL https://api.semanticscholar.org/CorpusID:8413785.

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.

Esther Thelen, Daniela Corbetta, Kathi Kamm, John P. Spencer, Klaus Schneider, and Ronald F. Zernicke. The transition to reaching: mapping intention and intrinsic dynamics. *Child development*, 64 4:1058–98, 1993. URL https://api.semanticscholar.org/CorpusID:7142337.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1466–1474, 2016. URL https://api.semanticscholar.org/CorpusID:2380406.

Nikolaus Vahrenkamp, Leonard Westkamp, Natsuki Yamanobe, Eren Erdal Aksoy, and Tamim Asfour. Part-based grasp planning for familiar objects. *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 919–925, 2016. URL https://api.semanticscholar.org/CorpusID:12049991.

Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Fei-Fei Li, and Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *ArXiv*, abs/2403.07788, 2024. URL https://api.semanticscholar.org/CorpusID:268363547.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

Jianglong Ye, Keyi Wang, Chengjing Yuan, Ruihan Yang, Yiquan Li, Jiyue Zhu, Yuzhe Qin, Xueyan Zou, and Xiaolong Wang. Dex1b: Learning with 1b demonstrations for dexterous manipulation. *arXiv preprint arXiv:2506.17198*, 2025.

Hanako Yoshida and Linda B. Smith. What's in view for toddlers? using a head camera to study visual experience. *Infancy : the official journal of the International Society on Infant Studies*, 13 3: 229–248, 2008. URL https://api.semanticscholar.org/CorpusID:18371897.

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.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

René Zurbrügg, Yifan Liu, Francis Engelmann, Suryansh Kumar, Marco Hutter, Vaishakh Patil, and Fisher Yu. Icgnet: A unified approach for instance-centric grasping. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4140–4146, 2024. URL https://api.semanticscholar.org/CorpusID:267034845.

A TOYS DESIGN

A.1 REAL 3D TOYS DESIGN AND MANUFACTURING

Primitive Design. To design the toys, we wrote a Python script that uses the SAPIEN physics engine to generate random dimensions for a set of primitives in the amount desired, such as a cuboid and a cylinder for a two primitive toy. These primitives are assembled into a toy by placing them at random offsets between them that ensure the primitives are still physically connected to each other. Finally, we export the toy mesh into an STL file using the Trimesh library. We list out the dimension ranges of the primitives in Table 6.

Table 6: Dimension ranges for primitive shapes.

Shape	Diameter/Width (cm)	Height (cm)	Length (cm)
Cuboid	2–7.2	1–20	2–28
Sphere	1–8	N/A	N/A
Cylinder	4–7	4–12	N/A
Ring	6–20	2–6	0.6–1.8 (wall thickness)

Toy Manufacturing. We printed a total of 250 toys in PLA filament, in addition to multiple test prints to validate the toy geometry and print quality. This was done using a fleet of eight Bambu P1P printers over a span of four weeks, enabling a maximum throughput of 200 toys per week by printing multiple toys on a single print bed (excluding FivePrimitive toys, whose size meant that they took up the entire print bed and took significantly longer to print). The fleet was managed using the Bambu Farm Manager platform.

The biggest challenge with printing the toys was the delicate geometry of the rings. The original designs had very thin ring walls that would snap during removal from the print bed. To compensate, we redesigned the toys to have thicker ring walls to strengthen the print. In addition, the intersection of shape primitives often resulted in large overhanging bodies, which required large amounts of tree supports to be modelled and printed. Toys with larger primitive counts had significantly higher print times due to their increased volume and complexity. Certain FivePrimitive toys had to be scaled down in size by 20% to fit in the 256mm x 256mm x 256mm print volume.

We have provided the full Bambu printer settings used for our prints for ease of reproducibility in Tables 8, 9, 10, and 11. Any omitted settings are assumed to take the default value. Organizing the toys into boxes and using a label printer to label them with their names is important for keeping track of all the toys, such as if a reprint is needed.

B ADDITIONAL EXPERIMENTS

Table 7: **Effect of toy colors on zero-shot generalization.** We compare the zero-shot performance of model trained on single-color toys with multi-color ones. Training on toys with multiple colors boost performance by about 1%-4% although training on single-color toys still yields a strong generalization to real objects.

Toy Colors	250	500	1000	1500	2000	2500
Red	50.1	66.44	68.94	72.6	75.48	76.35
Red + Green + Blue + Yellow	**56.63**	**68.17**	**71.15**	**74.62**	**76.82**	**80**

Effect of Color. We measure the impact of toy color on performance by conducting an ablation study comparing a policy trained on a set of only red toys to our original set where toys were randomly assigned one of four colors (red, green, blue, yellow). As shown in Table 7, color diversity improves performance. This is likely because exposure to toys with varying colors during training helps the model learn more robust visual features so it can generalize better to real-world objects.

C Real Robot Hardware Configuration

C.1 Franka Emika Panda

We deploy our policy on a Franka Panda Robot with 7 DoFs equipped with a RobotiQ gripper and a ZED 2i wrist camera. The 7 DoFs allow for precise and dexterous manipulation of the gripper to grasp various types of objects from every part. Two additional ZED 2i cameras are positioned to the left and right sides of the robot. Each camera provides an RGB stream at 720p and 30 FPS, without depth information. The hardware configuration is shown in Figure 5.

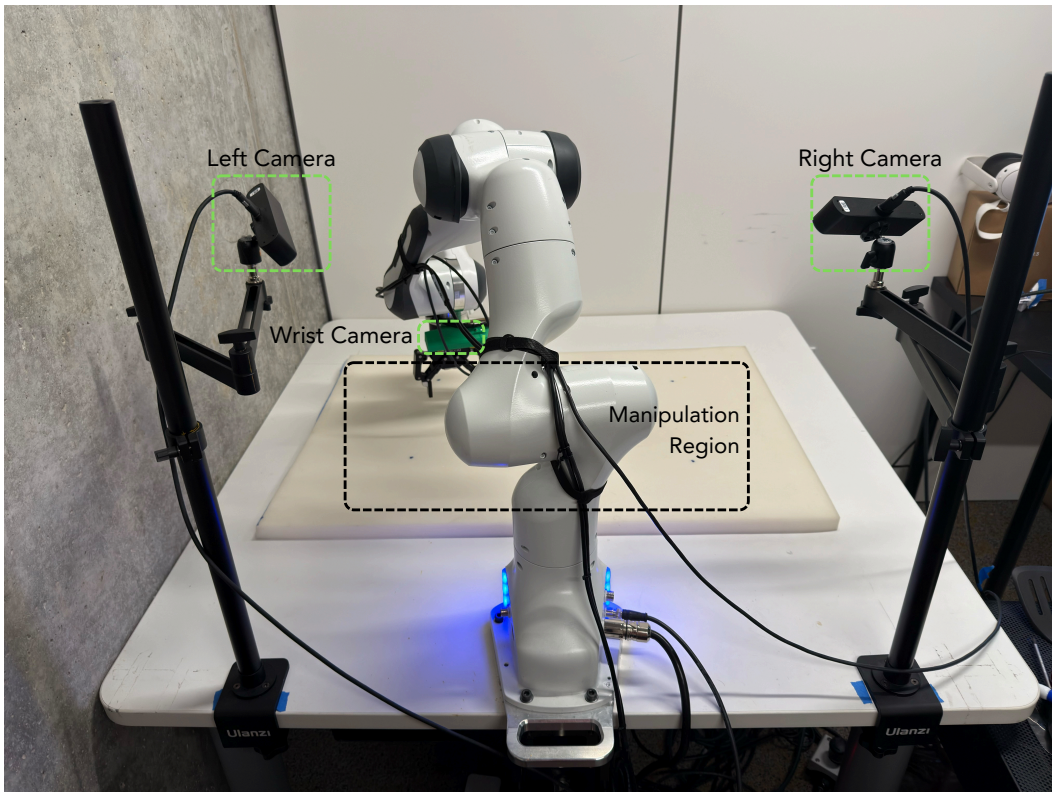

Figure 5: Hardware Configuration for Franka Emika Panda with Robotiq Gripper.

C.2 H1-2 Humanoid with Dexterous Hands

We also deploy our policy to a Unitree H1-2 humanoid robot. The robot is equipped with two Inspire RH56DFTP dexterous hands, each with 6 DoFs, 12 motors and a linear drive design with six miniature linear servo drives and six pressure sensors integrated inside. Given these characteristics, the hands are a good fit to emulate real dexterous operations by a human. The robot is also equipped with a ZED 2i head camera mounted below the original head camera to improve the quality of the egocentric data captured. Two ZED 2i cameras are positioned to the side of the robot, creating a similar setup to the one used for the Franka arm. Each camera provides an RGB stream at 720p resolution and 30 FPS, without depth information. The hardware configuration is shown in Figure 6.

D Robot Demonstrations Collection

D.1 ManiSkill Simulation Motion Planning

ManiSkill (Tao et al., 2025) is a simulation environment built on the SAPIEN framework. We generated a dataset of demonstrations for a Franka arm grasping and lifting single primitive objects using scripted planners.

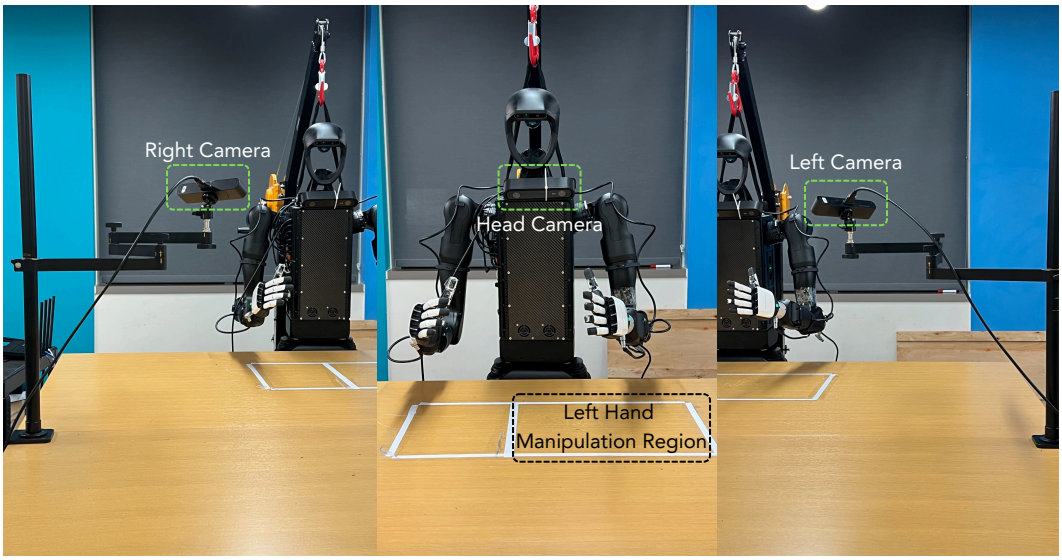

Figure 6: Hardware Configuration for H1-2 Humanoid with Inspire Dexterous Hands.

D.2 ManiSkill Simulation Teleoperation

Using ManiSkill, we also designed a simulation environment to collect data via human teleoperation. The teleoperation data collection process was then standardized as follows: the arm was first positioned slightly above the target grasp pose, then moved down to the grasp position, its gripper was closed to secure the object, and the object was then lifted upward. While it is possible to fully automate the data generation pipeline using grasping planners for more complex toys, we encountered engineering challenges that ultimately led us to rely on teleoperation.

D.3 Franka Real Robot Teleoperation

We teleoperated the Franka robot using a Meta Quest 3 headset, with only the right-hand controller mapped to arm control. Each pick-up demonstration was executed in one smooth motion on a foam-covered table to protect the objects. We recorded videos from the left and right ZED cameras as well as the wrist camera, and additionally logged the robot's proprioceptive states. We adopt the Franka-DROID robot settings provided by the DROID dataset (Khazatsky et al., 2024).

D.4 H1-2 with Dexterous Hands Teleoperation

To collect real-world data for the H1-2 humanoid robot, we used a teleoperation setup with the Apple Vision Pro (AVP) VR headset, built on Unitree's XR Teleoperate platform. The headset provides an RGB 2D view from the head camera, giving the operator a human-like perspective via the Vuer visualization toolkit. Our tracking script controls both dexterous hands and monitors the arms' poses; however, due to hardware limitations, we restricted data collection to the left arm and hand. Each recorded episode corresponds to a single toy-grasping demonstration.

E DetPooling

Creating Attention Masks. To pool visual features, we first extract the target object's segmentation mask from camera views. In the ManiSkill Franka simulation, ground truth object masks are directly available and used to identify vision encoder patches overlapping with the object. For the real Franka and H1-2 dexterous hand setups, we manually annotated 200 toy images with bounding boxes to train a Faster R-CNN detector with a ResNet-101 backbone [1] . The detector's bounding boxes are

[1] https://github.com/facebookresearch/detectron2

Table 8: Print Quality Settings

Setting	Value
Layer Height	0.3 mm
Initial Layer Height	0.3 mm
Line Width (All)	0.62 mm
Seam Position	Aligned
Smart Scarf Seam Application	On
Scarf Application Angle	155°
Scarf Steps	10
Scarf Joint for Inner Walls	On
Role-based Wipe Speed	On
Slice Gap Closing Radius	0.049 mm
Resolution	0.012 mm
Arc Fitting	On
Elephant Foot Comp.	0.15 mm
Ironing Type	No Ironing
Initial Layer Density	90%

Table 9: Print Speed Settings

Setting	Value
Initial Layer Speed	35 mm/s
Initial Layer Infill	55 mm/s
Outer Wall Speed	120 mm/s
Inner Wall Speed	150 mm/s
Top Surface Speed	150 mm/s
Sparse Infill Speed	100 mm/s
Travel Speed	500 mm/s
Normal Printing Accel.	10000 mm/s^2
Travel Acceleration	10000 mm/s^2
Initial Layer Travel Accel.	6000 mm/s^2
Initial Layer Accel.	500 mm/s^2
Inner Wall Accel.	0 mm/s^2
Outer Wall Accel.	5000 mm/s^2
Top Surface Accel.	2000 mm/s^2
Sparse Infill Accel	100%

Table 10: Print Strength Settings

Setting	Value
Wall Generator	Classic
Order of Walls	Inner/Outer
Bridge Flow	1
Wall Loops	2
Top/Bottom Shell Pattern	Monotonic
Top Shell Layers	3
Top Shell Thickness	0.8 mm
Bottom Shell Layers	3
Bottom Shell Thickness	0 mm
Internal Infill Pattern	Rectilinear
Sparse Infill Density	10%
Sparse Infill Pattern	Triangles
Infill/Wall Overlap	15%
Infill Direction	45°
Ensure Vertical Shell	Enabled

Table 11: Print Support Settings

Setting	Value
Enable Support	On
Type	Tree(auto)
Style	Default
Threshold Angle	30°
Remove Small Overhangs	On
Raft Layers	0
Top Z Distance	0.2 mm
Bottom Z Distance	0.2 mm
Top Interface Layers	2
Top Interface Spacing	0.5 mm
Support/Object XY Distance	0.35 mm
Support/Object First Layer Gap	0.2 mm
Tree Support Branch Distance	5 mm
Tree Support Branch Diameter	2 mm
Tree Support Branch Angle	45°

then used as input to SAM 2 to obtain segmentation masks, from which the attention masks are constructed in the same manner as in simulation.

Pooling Visual Features. For detector-based pooling, we follow a standard vision processing pipeline. The image is first patchified and passed through Transformer blocks. From the final block, we obtain spatial feature maps, and then apply the attention mask obtained above to pool the corresponding spatial features, yielding the final pooled features. For the visual encoder, we adopt the off-the-shelf ViT-L MVP model, which was pre-trained with a masked autoencoder objective and has been demonstrated to be effective for robotic control in prior work (Radosavovic et al., 2023).

F Robotic Policy Training Details

Observation. For the simulated Franka robot setting, we use three camera views as visual inputs: two fixed cameras mounted on the tabletop and one wrist-mounted camera. For the real Franka robot, the hardware configuration follows the standard DROID setup, with two tabletop-mounted cameras and one wrist-mounted camera. For LEGO policy training, we use only the two tabletop-mounted cameras as visual inputs.

Action Space. The LEGO policy is conditioned on the previous and current states, represented by the 7-DoF arm joint positions and the 1-DoF gripper state. The policy is trained to predict future action chunks, consisting of joint poses and gripper states.

Training Details. We adopt a learning rate of 5×10^{-4} with a weight decay of 0.01. Training is conducted for 900 epochs with a 30-epoch warm-up and a global batch size of 512. In comparison to foundation VLA models such as π_0-FAST (Black et al., 2023) and OpenVLA-OFT (Kim et al., 2024), our approach demonstrates substantially lower GPU memory requirements and achieves faster convergence, highlighting the efficiency of the proposed architecture.

G Baselines Implementation Details

G.1 π_0-FAST

We adopt π_0-FAST (Black et al., 2024) as a baseline for our simulated Franka, real-world Franka, and real-world H1-2 Dexterous Hands experiments, following the official code and instructions [2] .

Simulated Franka Robot. On the ManiSkill simulation platform, we fully finetuned the released base autoregressive π_0-FAST model on our simulated toy dataset. We use joint position control, adapting the pretrained model to predict the absolute 7-DoF joint pose and 1-DoF gripper status. We use left camera view and wrist camera view as visual inputs, and use "pick the toy" as the language instruction. We follow the default learning rate in the original implementation and finetune the model for 10K steps with a batch size of 32 for each setting reported in Table 1.

Real Franka Robot. For the real-world Franka robot, we use the DROID setting. Instead of velocity control, we adopt joint position control and finetune the released base autoregressive π_0-FAST on our teleoperated toy dataset. The pretrained model is adapted to predict the absolute 7-DoF joint pose and 1-DoF gripper status. We use left camera view and wrist camera view as visual inputs, and use "pick the toy" as the language instruction. Following the default learning rate, we train for 10K steps under both the 500-demonstration and 1500-demonstration settings shown in Table 2.

Real H1-2 Robot with Dexterous Hands. We also extend the setting to include humanoid arms with dexterous hands. Specifically, we finetune the released base autoregressive π_0-FAST on our 500-demonstration teleoperated toy dataset using delta joint control. We experiment with both absolute joint control and delta joint control for the 7-DoF right arm, 6-DoF wrist torque, and 6-DoF finger angles (totally a 20-dim action). We use left camera view and head camera view as visual inputs, and use "pick the toy with dual arms" as the language instruction. Results show that delta control outperforms absolute control.

However, in this new embodiment-specific setting (compared with DROID setting, which the pretrianing covers it), we find that π_0-FAST tends to overfit with limited data, likely due to its large model size. To mitigate overfitting, we select an early checkpoint where the cross-entropy loss reaches a reasonable value greater than 1 (but for DROID, it will not overfit even with a $1e-2$ loss probably since its pretrained on large amount of DROID data). For the reported results in this paper, we follow the default learning rate and train for 1K steps using 500 demonstrations, as summarized in Table 2.

G.2 OpenVLA-OFT

We use OpenVLA-OFT (Kim et al., 2025) as a baseline for both simulation and real-world experiments, following the official implementation and finetuning instructions [3]. We use LoRA (Hu et al., 2021) finetuning with a rank of 32 for all experiments.

Simulated Franka Robot. On the ManiSkill simulation platform, we use delta joint position control and input images from the front, base, and wrist cameras. The model is trained with a batch size of 2 and an initial learning rate of 1.25e-4, decayed to 1.25e-5 after 100,000 steps. Training runs for a total of 150,000 steps, with checkpoints at every 20,000 steps evaluated to select the best-performing model for each experiment.

[2] https://github.com/Physical-Intelligence/openpi

[3] https://github.com/moojink/openvla-oft

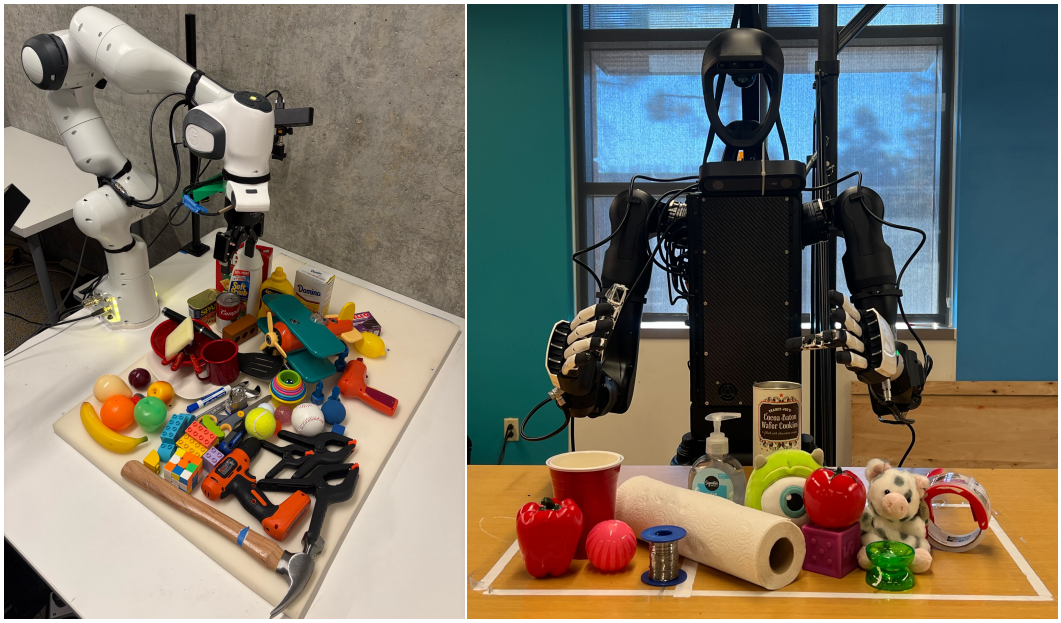

Figure 7: **Real-world Evaluation Settings.** We have DROID Franka setting with YCB dataset on the left and H1-2 robot with dexterous hands and 13 everyday objects.

Real Franka Robot. In the real Franka DROID setting, we use delta joint position control, consistent with the simulation experiments. The model receives images from the left, right, and wrist cameras. Training uses a batch size of 2 and an initial learning rate of 1.25e-4, decayed to 1.25e-5 after 100,000 steps, for a total of 150,000 steps. We used the last checkpoint for evaluation.

Real H1-2 Robot with Dexterous Hands. The model is conditioned on images from the left, right, and head cameras. It receives a 26-dimensional state vector—corresponding to 7 DoF per arm and 6 DoF per hand—and predicts a 40-dimensional output, which includes absolute joint targets for all joints as well as feedforward torques for both arms. Training uses a batch size of 2 and an initial learning rate of 1.25e-4, decayed to 1.25e-5 after 100,000 steps, for a total of 150,000 steps. We used the last checkpoint for evaluation.

G.3 SHAPEGRASP

We evaluate ShapeGrasp on our real Franka setup using the official implementation[4] . ShapeGrasp uses GPT-4o to identify a graspable part from a decomposition graph, where nodes represent object parts (modeled as convex shapes) and their spatial relationships. It outputs a pixel location along with a z-axis rotation for a top-down grasp. Using a calibrated Intel RealSense D435 camera, we project the pixel-level grasp prediction into 3D space. An executable grasp trajectory is then generated by interpolating between the robot's current pose and the predicted grasp pose.

H EVALUATION DETAILS

For the simulated Franka robot, we use the default task environment "PickClutterYCB-v1" for evaluation, with details available in the official documentation. For the real-world experiments, we consider two settings, as shown in Figure 7. The left panel illustrates the standard DROID setup with the YCB dataset used for evaluation, while the right panel shows the H1-2 robot equipped with Inspired dexterous hands and the 13 everyday objects used for evaluation.

[4]https://github.com/samwli/ShapeGrasp

H.1 ManiSkill Simulation Evaluation

To evaluate policies in simulation, we defined a 0.15×0.15 m square workspace, subdivided into a 4×4 grid. The grid was constructed from the Cartesian product of the sets $\{-0.075, -0.025, 0.025, 0.075\}$ along both the x and y axes, resulting in 16 evenly spaced placements. For each trial, the object was placed at one grid location with its z-rotation initialized using a random seed. Each object was tested across all 16 placements, and success rates were averaged across objects and placements. A trial was considered successful (1) if the robot lifted the object above a height threshold of 0.3 m. For OpenVLA-OFT policies, we reduced the success threshold to 0.15 m, as the gripper would often prematurely open after grasping the object for these policies. Trials in which the object was not lifted above the threshold were marked unsuccessful (0).

H.2 Franka Robot Evaluation

To evaluate our policy on the Franka Panda arm, we defined a 0.5×0.28 m rectangular workspace on the table, subdivided into a 4×4 grid. For each trial, the object was placed in one of the 16 grid cells, with its z-axis orientation randomized. We evaluated policies by testing each object across all 16 placements and averaged the results to compute the final success rate. A trial was considered successful (1) if the robot securely lifted the object above a height threshold of 0.2 m, and unsuccessful (0) otherwise.

H.3 H1-2 Humanoid Dexterous Hands Evaluation

To evaluate our policy on the H1-2, we defined a grasping workspace by taping off a $40\,\text{cm} \times 36\,\text{cm}$ rectangular zone on the table, positioned within the head-mounted ZED camera's field of view and centered between the two Inspire hands. This rectangle was subdivided into six equally sized $3\,\text{in} \times 3\,\text{in}$ squares. For each object tested, we conducted five grasping trials, placing the object in a different square for each trial. Performance was scored as 1 if the robot successfully picked up the object and 0 otherwise. All trials were executed using the left arm and hand. During evaluation, we encountered technical issues with the Inspire hands—most notably unresponsive thumb joints on both sides—which limited the scope of humanoid grasping experiments we were able to carry out.