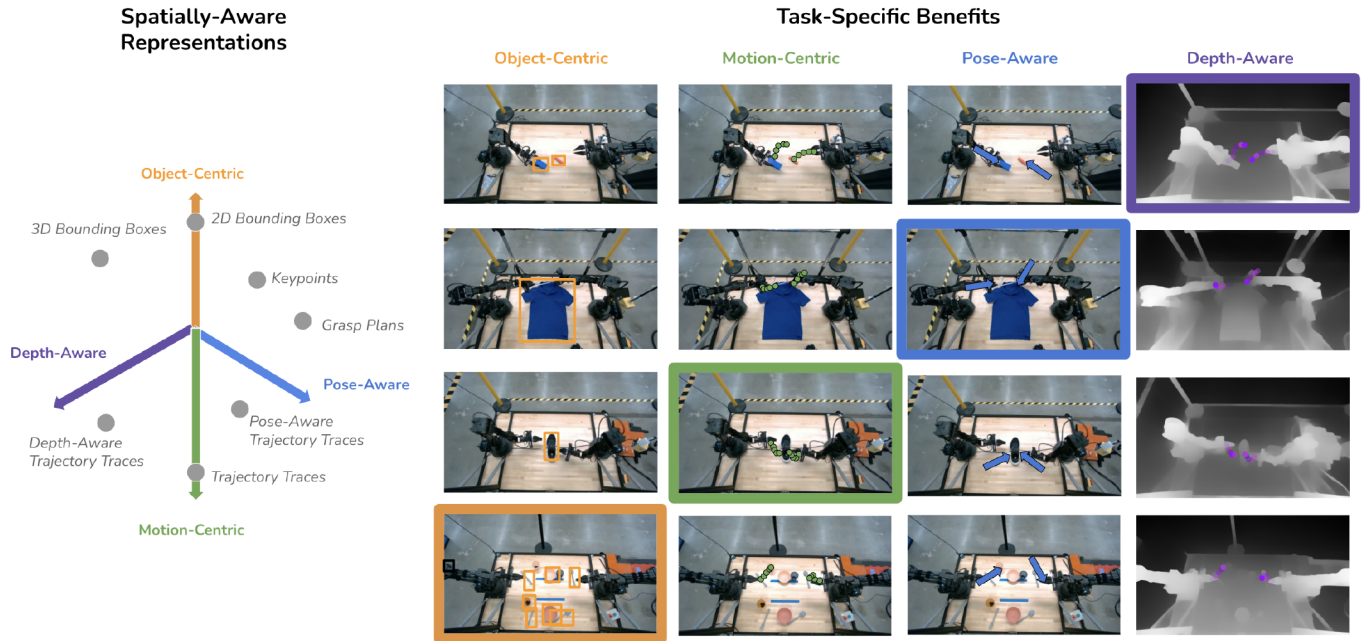


# Bridging Perception and Action: Spatially-Grounded Mid-Level Representations for Robot Generalization

Author Names Omitted for Anonymous Review. Paper-ID 750



**Fig. 1: Bimanual, dexterous manipulation requires task-specific grounding.** The left depicts various axes for spatial grounding as well as qualitative categorizations of different mid-level representations. Different representations lead to different levels of improvement depending on the task.

**Abstract**—In this work, we investigate how *spatially-grounded* auxiliary representations can provide both broad, high-level grounding, as well as direct, actionable information to help policy learning performance and generalization for dexterous tasks. We study these mid-level representations across three critical dimensions: object-centricity, pose-awareness, and depth-awareness. We use these interpretable mid-level representations to train *specialist* encoders via supervised learning, then use these representations as inputs to a diffusion policy to solve dexterous bimanual manipulation tasks in the real-world. We propose a novel mixture-of-experts policy architecture that can combine multiple specialized expert models, each trained on a distinct mid-level representation, to improve the generalization of the policy. This method achieves an average of 11% higher success rate on average over a language-grounded baseline and a 24% higher success rate over a standard diffusion policy baseline for our evaluation tasks. Furthermore, we find that leveraging mid-level representations as supervision signals for policy actions within a weighted imitation learning algorithm improves the precision with which the policy follows these representations, leading to an additional performance increase of 10%. Our findings highlight the importance of grounding robot policies with not only broad, perceptual tasks, but also more granular, actionable representations.

## I. INTRODUCTION

Large pre-trained robotics models have made significant progress in recent years towards improving robotic generalization capabilities by leveraging large-scale pre-training datasets. However, these models still face challenges in adapting to slight scene variations such as different spatial locations, unseen objects, and different lighting conditions. An increasingly popular approach to address this challenge is explicitly establishing deeper connections between robot policies and the abstract patterns and relationships that govern the physical world. For example, vision-language-action models (VLAs) make an attempt to benefit from semantic and visual knowledge from vision-language models (VLMs) by fine-tuning these models with robot data. Other works use explicit *mid-level representations* such as low-level language instructions [2], key-points [25, 36], or trajectories [14, 31, 38] to provide additional grounding to the robot policy. Despite the success of these methods, the generalization properties that these additional forms of grounding enable are still high-level in nature, and their benefits for simpler tasks may not necessarily transfer to more complex tasks – ones that require further dexterity or object interactions.

We hypothesize that the choice of mid-level representations is highly dependent on task-specific requirements. For instance, for a robot tasked with folding a shirt, a bounding box may help locate a shirt’s general position but fails to provide actionable information on how to manipulate it. Similarly, depth-informed representations are crucial for contact-rich tasks such as handing over objects, but may be less important for overhead pick-and-place tasks. In order to give models stronger generalization capabilities for a wide variety of dexterous tasks, it is essential to explore representations that balance high-level abstraction and low-level actionable detail. These representations must not only encode spatial and geometric reasoning but also offer adaptability across diverse and dynamic environments.

In this paper, we find a set of mid-level representations that enhances the adaptability of a robot policy across a wide variety of environments. We first systematically study these representations across four critical dimensions: **object-centricity**, or understanding of the locations and geometry of objects on the scene; **motion-centricity** or understanding of the future motion of the robot; **pose-awareness**, or understanding of spatial orientations; and **depth-awareness**, or understanding about three-dimensional structure and geometry. Through these experiments, we identify how different forms of spatial grounding align with task-specific requirements. We then propose a method to leverage these representations through a diffusion policy-based model conditioned on multiple experts outputting interpretable representations. We show that while different mid-level representations excel at different tasks, our method can leverage these task-specific benefits to achieve consistently higher performance on a wide range of environments.

In addition, we further investigate how robot policies utilize the aforementioned representations. We find that reliance on structured signals presents a trade-off: policies that depend heavily on these representations can become more susceptible to overfitting and reduced robustness in noisy environments. To mitigate this, we introduce key architectural design choices that balance sensitivity to mid-level representations with resilience against spurious noise in these representations. Finally, we incorporate these representations as additional training signal. We refer to the alignment of the demonstrations with these mid-level representation as *self-consistency*, which can provide a weighting scheme for weighted imitation learning – upweighting data with self-consistent representations. This approach further refines the policy’s ability to execute mid-level plans with greater precision and reliability.

Our policy, Mid-Level MoE, achieves a 24% higher success rate than a standard diffusion policy baseline and an 11% improvement over a baseline using language representations across a series of bimanual, dexterous tasks. Furthermore, our weighted imitation learning method enhances the policy’s reliance on mid-level representations while preserving robustness to perturbations, resulting in a 10% higher success rate compared to standard training. These results highlight that a deeper understanding of robot grounding can lead

to significant improvements in robot success in dexterous, bimanual environments.

## II. RELATED WORKS

Training *generalist* robot policies has been typically approached as a multi-task learning problem, where a single machine learning model is optimized for different behaviors and objectives. Many prior works have involved scaling up the breadth and diversity of robot data [8, 12, 21, 22, 26, 30, 40]. A key challenge with the multi-task policy learning regime is in obtaining policies that *generalize* to new objects, task variants, environmental factors and so on. Towards this, a significant body of work has focused on generalizing robot policies to grasp new objects [19, 28, 32]. In this work, we are interested in achieving a similar form of generalization for dexterous manipulation [6, 39]. Prior works in learning multi-task dexterous policies have struggled with generalizing to entirely new objects due to the high-dimensional observation and action spaces.

While the typical recipe to obtaining generalizable policies is to scale robot data, collecting such data remains prohibitively expensive. A promising alternative is to introduce structure into the end-to-end pixels-to-actions mapping. To provide robot models with a greater understanding of commonalities in robot tasks, planning, and behavior, several previous works have considered conditioning robot policies with higher-level representations of robot behavior [16, 27, 33, 37]. To increase the interpretability and controllability of robot policies, many works have instead conditioned on *explicit* representations. These representations typically been specified either through goal images [9, 29], video demonstrations [11, 35] or language [13, 20, 41]. While many earlier works have used higher-level conditioning information at the task specification level, recent works have worked towards getting robots to achieve more specific goals, often specified in language [1, 3, 17, 18].

One potential drawback to the hierarchical learning framework is its rigidity in structure. Recently, many works have started viewing adding structure between robot perception and action as adding a more general "grounding" to the policies. A modern instantiation of this class of methods has been distilling additional knowledge into robot policies in the form of auxiliary tasks. For instance, some methods pre-train robot policies using regularization tasks such as visual question-answering (VQA) [4, 7, 10, 23], language planning [15, 24, 34], or spatial reasoning [5]. Other approaches explicitly condition on higher-level representations, including language [2, 38] or image annotations [14, 25, 31]. However, many of these representations are disconnected from the physics of a robot’s interaction with the world, and fail to capture the precise spatial and contextual details required for dexterous manipulation. For example, providing a language caption or object bounding box for a robot’s workspace is not particularly informative about the object’s geometry, orientation, affordances, and so on. In this paper, we investigate spatial mid-level representations that bridge the gap



between high-level inputs (e.g., language commands or simple object markers) and the low-level action space of a robot. By grounding policies in richer spatial details, we aim to achieve better generalization and more reliable performance across a wide range of environments and tasks.

### III. SPATIALLY-GROUNDED MID-LEVEL REPRESENTATIONS

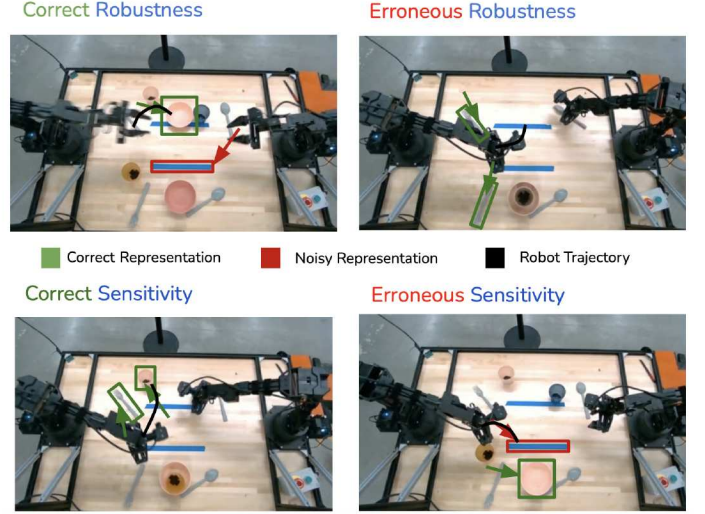
Robots that effectively generalize across diverse environments require an understanding of broad, high-level abstractions intrinsic to the real world, such as object geometry, spatial relationships, and motion dynamics. While one can hope to learn these relationships directly from end-to-end data, current large-scale robot policies that try to scale up imitation learning still struggle with performing dexterous tasks in environments that involve slight shifts in these properties. To address this issue, instead of implicitly relying on black-box feature extraction, we propose to explicitly incorporate representations that capture these mid-level abstractions, enabling robot policies to adapt more robustly to variations in real-world settings.

Concretely, we assume access to a dataset  $\mathcal{D} = \{\tau_1, \tau_2, \dots, \tau_n\}$ , where each trajectory  $\tau$  is a series of states and actions  $(s_1, a_1), (s_2, a_2), \dots, (s_t, a_t)$ . While typical end-to-end imitation learning aims at finding a mapping  $\pi(a|s)$ , we instead add a set of experts that generate mid-level representations  $\{E_i(s)\}_{i=1}^k$ , each of which provides a specific type of grounding. We then aim to learn a policy  $\pi(a|E_1(s), E_2(s), \dots, E_k(s), s)$  which can leverage these representations to perform more robustly across diverse scenarios.

What representations would lead to the best performance for robot policies? There exists a hierarchy of representations, spanning from low-level geometric features to high-level symbolic structures such as language-based subtasks. For instance, language subtasks provide flexible, interpretable scaffolding that allows policies to be decomposed into modular instructions, facilitating transfer and reuse across different tasks. In parallel, higher granularity representations—like identified grasp locations—provide the precision required for dexterous manipulation and robust adaptation to slight variations in object geometry or environment layout. In this work, we concentrate on representations that hold the potential to enhance both the versatility and generality with which robots can perform dexterous tasks. By focusing on mid-level abstractions that capture critical factors—such as object geometry, spatial relationships, and motion dynamics—our goal is to equip robots with the adaptability needed to handle subtle variations in real-world environments more robustly.

We first investigate the utility of representations on four main axes: object-centricity, motion centricity, depth-awareness, and pose-awareness to provide a comprehensive spatial grounding for robot policies. In addition,

- 1) **Object-Centric Representations:** These experts focus on extracting information pertinent to each object in the scene, such as object poses, sizes, shapes, and



**Fig. 3: The sensitivity-robustness tradeoff.** Policies need to follow their mid-level representations while being robust in erroneous noise to these representations.

potential interaction points (e.g., grasp points, keypoints for alignment).

- 2) **Motion-Centric Representations:** These experts capture and interpret the dynamic aspects of the environment by focusing on how objects, the robot, or other agents move and interact over time. For instance, they may encode object velocities, accelerations, potential collision points, or kinematic constraints.
- 3) **Depth-Aware Representations:** These experts leverage depth information to infer spatial relationships and physical constraints in the environment. For example, they can identify occlusions, measure distances between objects, and compute volumetric properties. By incorporating depth-aware reasoning, we enable robots to make more accurate and reliable decisions in 3D environments.
- 4) **Pose-Aware Representations:** These experts encode the relative and absolute poses of objects, as well as the robot's own pose, in a way that supports precise manipulation tasks. For example, pose-aware experts can compute alignment requirements for assembly tasks or predict optimal configurations for stable grasps.

These four axes—object-centric, motion-centric, depth-aware, and pose-aware—work together to provide a comprehensive spatial grounding for robot policies.

### IV. THE SENSITIVITY-ROBUSTNESS TRADEOFF

Previous works have predominantly focused on how various forms of grounding can enhance the performance of robotic policies. However, we argue that investigating mid-level relationships—specifically, how policies adhere to and utilize their representations—is equally crucial. By analyzing this relationship, we can view the mid-level representations as a *bridge* between the sensory inputs of the policy and the lower-level joint actions. This enhances the interpretability

of our policy by disentangling errors with the policy’s mid-level decision making processing with its lower-level action generation. Through exploring this relationship, we identify a fundamental tradeoff between the sensitivity with which a robot follows its representations and its robustness to errors in these representations.

Consider a household robot tasked with cleaning kitchenware, which receives mid-level representations from  $k$  different experts  $E_1, E_2, \dots, E_k$  (see Figure 3). Suppose  $E_1(s)$  provides the locations of objects of interest in the scene. A policy which utilizes these representations must consistently use these locations to output actions that move towards the correct target. The degree to which the policy follows this representation can be understood as *sensitivity*. If one of the representations has an error, such as  $E_1$  incorrectly detecting an object, the robot may attempt to interact with the wrong item, leading to improper handling or placement of kitchenware. The degree to which the policy is able to take reasonable actions in the presence of these perturbations can be understood as *robustness*.

This sensitivity-robustness tradeoff underscores the necessity of developing robot policies that balance adherence to mid-level representations with the ability to remain adaptable and resilient in the face of environmental variations. This balance ensures that, while policies leverage detailed, task-relevant information for precise manipulation, they do not become overly dependent on specific features that may change or vary in different contexts.

Our usage of *spatially-aware* mid-level representations allows us to directly quantify the tradeoff between generalization and executability in robotic policies. We propose two key metrics to measure this tradeoff:

- 1) **Sensitivity Score (SS):** The Sensitivity Score quantifies the extent to which the policy adheres to the provided mid-level representations during task execution. Specifically, it measures how variations or perturbations in the representations influence the resulting trajectories of the robot. Formally, let  $f(s, E, \tau)$  represent a function that evaluates the adherence of the trajectory  $\tau$  to the representations  $E = \{E_1(s), E_2(s), \dots, E_k(s)\}$  given the state  $s$ . The Sensitivity Score is defined as:

$$SS(E) = \mathbb{E}_{s, \tau} [\text{Adherence}(E(s), \tau)]$$

where  $\text{Adherence}(\cdot)$  is a metric quantifying the alignment between the policy’s trajectory and a particular mid-level representation. A lower SS indicates that the policy closely follows the representations. Conversely, a higher SS implies that the policy is less reliant on specific representations. Further information on how  $\text{Adherence}(\cdot)$  is computed for each representation can be found in Appendix C.

- 2) **Robustness Index (RI):** The Robustness Index measures the policy’s ability to maintain stable and effective performance when perturbations are introduced to the mid-level representations. For each state  $s$  in a trajectory

$\tau$ , and for each mid-level representation  $E_i(s)$  generated by the experts, we apply a Gaussian perturbation:

$$\tilde{E}_i(s) = E_i(s) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

The perturbed representations  $\tilde{E}_i(s)$  are then used as inputs to the policy:

$$\pi(a \mid \tilde{E}_1(s), \tilde{E}_2(s), \dots, \tilde{E}_k(s), s)$$

Let  $P_j^{(\text{perturb})}(\sigma)$  represent the policy’s performance under the  $j$ -th Gaussian perturbation with standard deviation  $\sigma$ , and  $P_0$  its performance with the original, unperturbed representations. The Robustness Index is defined as the mean of the performance ratios across all perturbations:

$$RI(\sigma) = \frac{1}{m} \sum_{j=1}^m \frac{P_j^{(\text{perturb})}(\sigma)}{P_0}$$

where  $m$  is the number of distinct Gaussian noise realizations applied to the representations. A higher RI signifies that the policy remains resilient and maintains its effectiveness despite perturbations in the representations. Conversely, a lower RI suggests that the policy is prone to performance degradation when the representations are altered, highlighting potential overfitting to the exact representations provided during training.

## V. ARCHITECTURE

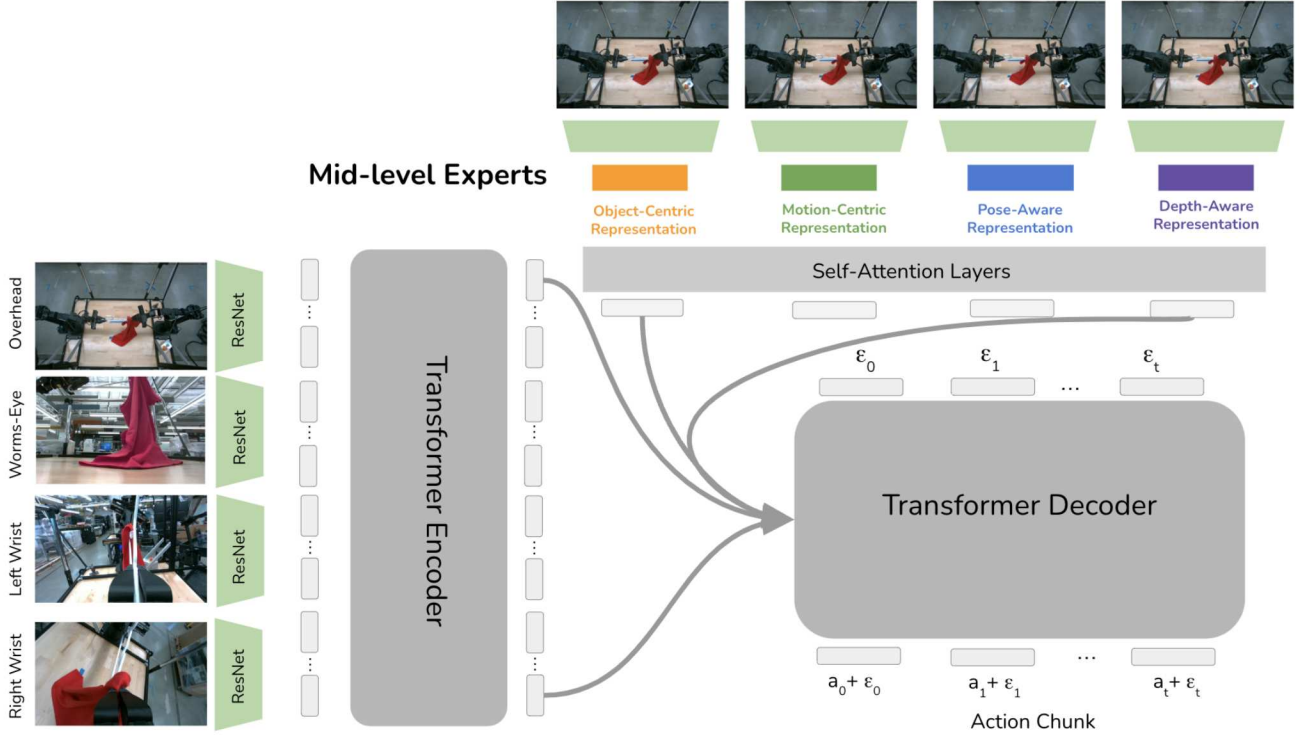
We implement our method on a diffusion policy similar to the one proposed in [40]. The policy takes as input 4 images from different viewpoints (2 third-person images and 2 wrist images) and directly outputs 12 absolute joint positions—6 for each arm—as well as a continuous gripper value per each gripper. Each image is fed through a separate ResNet50 encoder, before being processed with a transformer encoder to obtain image embeddings. At each state, we denoise the decoder predicts  $t = 10$  action chunks simultaneously with a transformer. See Figure 4 for a depiction of our architecture.

### A. Mid-level Experts

We design our architecture motivated by the sensitivity-robustness tradeoff. At each state, the robot must discern which representations are pertinent for the task at hand and output actions which follow these representations. Meanwhile, if any of the representations have noise, the policy must output reasonable actions that maintain task performance despite these disturbances. To achieve this, we employ three key design choices:

- 1) **Diverse Mid-level Experts** We employ  $k = 4$  mid-level experts to output representations corresponding to our aforementioned axes of grounding. In particular, we have separate object-centric, motion-centric, pose-aware, and depth-aware representations. These correspond to bounding boxes, trajectory traces, grasp plans, and depth-aware traces.





**Fig. 4: Policy Architecture.** Four images are passed into a transformer encoder. In addition, an image is fed into each individual mid-level expert. The results embeddings are passed into the transformer decoder through cross-attention.

- 2) **Attention-based Mid-level Gating:** In order to combine our experts, we use multi-headed attention to provide an early form of gating. Specifically, the embeddings are processed into a multidimensional tensor  $z = \text{MultiHead}(R_1, R_2, R_3, R_4)$ , where each  $R_i$  is the representation from expert  $E_i$ . This gating strategy dynamically weights each expert’s contribution based on the current state, enabling effective integration of diverse representations.
- 3) **Cross-Attention Mechanism:** We then perform cross-attention between the processed mid-level embeddings  $z$  and the image embeddings. This allows the policy to further use information from its images to determine which representations to emphasize for action selection. This ensures that the policy dynamically prioritizes the most relevant representations based on the current visual and contextual information, as well as be more robust to errors in these representations.

## B. Training

To train our policy, we adopt a two-stage approach. In the first stage, the expert mid-level representations are trained separately on data tailored to each representation. This data is purely relabeled from demonstrations and proprioception. For motion-centric representations, we use proprioception data, which refers to the sensor data related to the arm’s position and movement, to process the arm’s trajectories at future points in time. We utilize a trajectory length of 10

points, with one annotation every 10 timesteps. We project these representations into the observation frame using the camera intrinsics and extrinsics. In simulation, these values are provided by the simulator, while in the real world, they are calibrated using an AprilTag board.

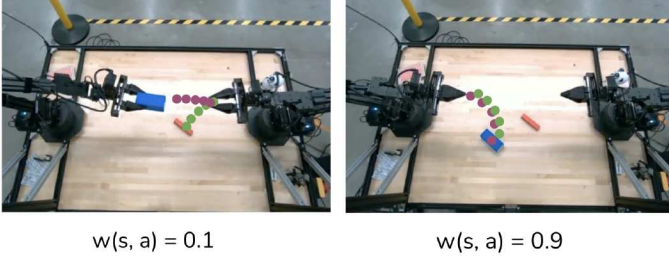
We base our object-centric representations on an off-the-shelf OWL-ViT (153M params). The language prompts are carefully tuned for each task to minimize noise as much as possible. For pose-aware representations on top of bounding boxes, we combine proprioception by processing states where the arm comes into contact with objects. We then retrain ResNet34 models (21M params) on top of this relabelled data. We find that that for more dexterous tasks, this is important in order to avoid significant decreases in the frequency of robot control by using an OWL-ViT in-the-loop.

Once the expert modules are trained independently, their parameters are frozen. Then, the policy network trained end-to-end with a noise prediction loss. During inference time, each of the expert models are executed asynchronously.

## C. Self-Consistency

A primary challenge in integrating mid-level representations into robot policies lies in ensuring that the policy consistently follows the guidance provided by these representations. Object-centric attributes, depth-aware insights, or pose-aware signals serve as mid-level “expert outputs” that the policy is incentivized to replicate. However, direct supervision through standard behavioral cloning (BC) can lead to inconsistencies, especially when the mid-level predictions are noisy or only

Mid-level Representation Achieved Trajectory



**Fig. 5: Self-Consistency.** On the left image, the robot’s achieved trajectory doesn’t match its mid-level representation, which leads to a lower weight. In the right, the robot follows its representation, leading to a higher weight.

partially correct. Such inconsistencies ultimately degrade the policy’s ability to effectively utilize the expert signals.

Our proposed method is analogous to reinforcement learning with a hand-crafted advantage function. In RL, the loss function typically incorporates an advantage term, given by:

$$\mathcal{L}_{\text{RL}} = \mathbb{E}_{(s,a) \sim \mathcal{D}} [A(s,a) \cdot \mathcal{L}_{\text{PG}}(\pi(a|s))]$$

where  $A(s,a)$  represents the advantage function, which modulates the policy gradient loss  $\mathcal{L}_{\text{PG}}$  based on the estimated benefit of selecting action  $a$  in state  $s$ . Similarly, our approach integrates mid-level expert outputs as implicit guidance in scenarios where no explicit reward signal is available. Instead of an advantage function, we introduce *self-consistency weights*  $w(x)$ , which serve to emphasize reliable expert guidance. The corresponding loss function is:

$$\mathcal{L}_{\text{policy}} = \mathbb{E}_{(x,a) \sim \mathcal{D}} [w(x) \cdot \mathcal{L}_{\text{BC}}(\pi(a|x), a)]$$

where  $w(x)$  reflects how well the mid-level expert outputs align with ground-truth or improve task success. Notably, our representations encode the policy’s desired future behavior, similar to how an advantage function models a policy’s expected future reward. By structuring policy learning in this way, our method ensures that mid-level expert outputs provide meaningful guidance, akin to an advantage function in reinforcement learning. This approach prioritizes high-quality supervision, improving the policy’s ability to effectively utilize expert-generated representations. By iteratively refining the training data and adjusting the weighting of consistent samples, our method creates a feedback loop that promotes tighter self-consistency between policy actions and mid-level expert outputs. See Algorithm 1 for more details. The exact method to compute the weights can be viewed in Appendix ??

## VI. EXPERIMENTS

Our goal is to evaluate the effectiveness of mid-level representations as grounding for training robotic policies, focusing on their ability to improve task performance across diverse scenarios. Specifically, we aim to assess how different forms of spatial grounding—object/motion-centricity, pose-awareness, and depth-awareness—impact a robot’s ability to generalize, execute precise actions, and recover from noisy inputs.

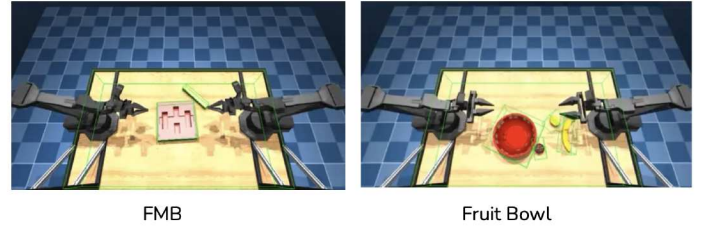
### Algorithm 1 Weighted Self-Consistency Training

**Require:** Policy  $\pi_\theta$  (with parameters  $\theta$ ), Mid-level experts  $f_{\text{expert}}$ , Training dataset  $D = \{(s_i, a_i)\}$ , Self-consistency threshold  $\delta$ , Learning rate  $\eta$

- 1: Initialize policy parameters  $\theta$
- 2: **repeat**
- 3:   Sample a minibatch of  $B$  states/actions  $\{(s_i, a_i)\}_{i=1}^B$  from  $D$
- 4:   Compute mid-level outputs  $m_i = f_{\text{expert}}(s_i)$  for each sample
- 5:   Determine self-consistency weights  $w_i = W(m_i, a_i, s_i) = 1 - \exp(-\lambda_E \cdot \text{Adherence}(E(s), \tau))$
- 6:   Compute weighted BC loss:

$$\mathcal{L}_{\text{BC}} = \frac{1}{B} \sum_{i=1}^B w_i \ell(\pi_\theta(s_i), a_i)$$

- 7:   Update policy parameters:  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{BC}}$
- 8: **until** convergence



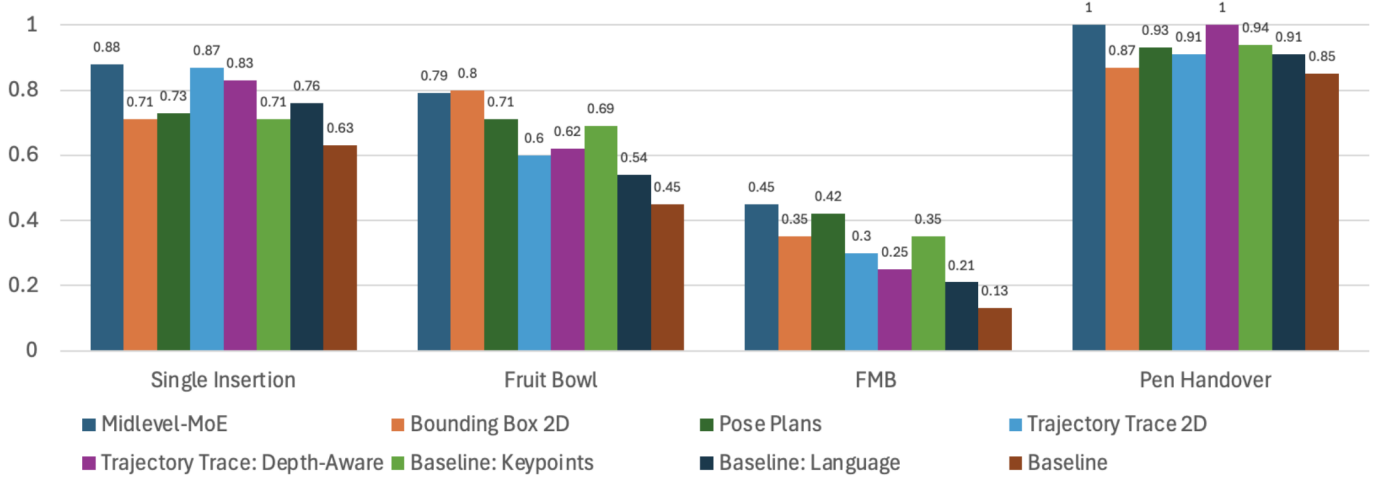
**Fig. 6: Simulation Tasks.**

- 1) Are mid-level representations effective in improving policy generalization performance across a range of tasks and environments?
- 2) What types of tasks benefit the most from specific mid-level representations, such as object/motion-centricity, pose-awareness, or depth-awareness, and how do these align with task-specific requirements?
- 3) Can a policy effectively utilize multiple sources of mid-level representations, and how does the integration of these diverse signals impact task performance and generalization?
- 4) Which policy architecture offers the best tradeoff between responsiveness to structured mid-level representations and robustness to noise or spurious inputs?
- 5) Can mid-level representations be effectively used as supervision signals during training to enhance policy precision and generalization across tasks?

#### A. Simulation Environment

We evaluate our method on the Aloha Unleashed simulation environment [40]. This environment consists of a bimanual parallel-jaw gripper robot workcell. The observations contain images from 4 different points of view: the overhead camera, worms-eye camera (facing the robot), and two wrist cameras.





**Fig. 7: Simulation Results.** Mid-level MoE achieves a 24% higher success rate over a standard diffusion policy baseline. It performs consistently well over different tasks by leveraging different representations.

- **Single Insertion:** The robot must pick up a peg with one arm, and a block containing a hole with another. Then, it must align the peg with the block and insert it into the hole.
- **FMB Assembly:** The robot must pick up a multiple blocks and place it into its appropriate slots. Each of the blocks must be placed in precise locations that require the robot to have good understanding of object geometry.
- **Fruit Bowl:** The robot must arrange a variety of fruits into a bowl, ensuring efficient use of space or adherence to a specific pattern. This task tests the robot’s ability to handle objects of different shapes and sizes while reasoning about spatial configurations.
- **Pen Handover:** The task requires the robot to perform a smooth and reliable pen handover to a human. This includes grasping the pen, orienting it correctly for comfortable use, and transferring it to the human’s hand without dropping or misalignment. The task tests precision grip, human-robot interaction, and timing coordination.

### B. Real-World Environment

We evaluate our method on a similar real-world environment consisting of two 6-DoF ViperX robot arms with parallel-jaw grippers. The observations perspectives are the same as in simulation.

- **Kitchen Stack:** The robot must organize and stack a set of kitchen items, such as bowls, plates, and cups, in an orderly manner on a designated shelf or surface. The task emphasizes spatial reasoning, stability prediction, and careful object placement.
- **Cup Stacking:** The robot must pick up multiple cups on the scene and stack the cups. This tasks tests the ability of the robot to generalize across different combinatorial object locations across the workstation.
- **Shirt Hanging:** The task requires the robot to hang a shirt on a hanger. The steps include flattening the shirt,

picking up a hanger from a rack, moving the shirt to a desirable location, placing the hangar onto the shirt, then picking up the hangar and putting it back onto the rack.

- **Shoelace Tying:** The robot must tie the laces of a shoe into a bow. This involves centering the shoe, straightening the laces, crossing them over, forming loops, and tightening the bow. The task emphasizes a robot’s fine motor control.

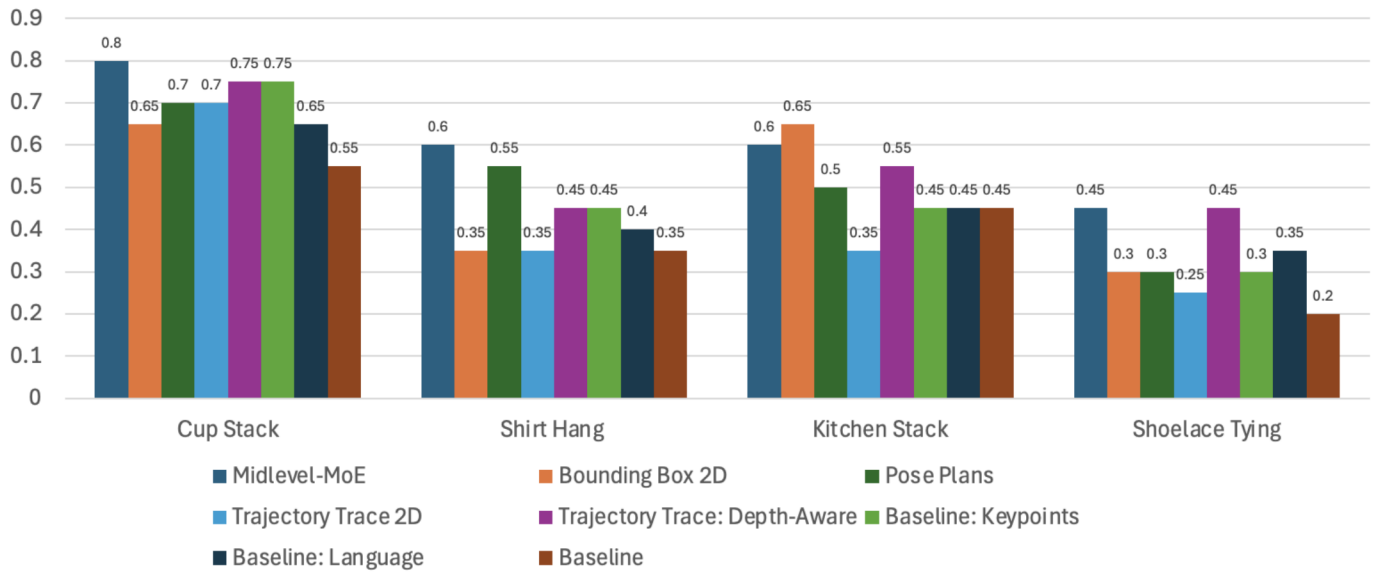
### C. Experiment Setup

To evaluate the impact of a singular representation on our policy, we ablate our method by conditioning on a single expert. In addition, we provide two ablations based on prior works investigating a single representation: a keypoints-based ablation based on MOKA [25] and a language baseline based on RT-H [2]. In the keypoint ablation, we identify important points of interest in the image by querying a VLM. For RT-H, we relabel robot demonstrations with the language "move the arm left/right/up/down." For each environment in simulation and the real-world, we vary the object locations, add distractor objects, and change the lighting conditions.

## VII. ANALYSIS

### A. Mid-level Representations have Task-Specific Benefits

Figures 7 and 9 present our experimental results in both simulation and real-world environments. We find that while all representation ablations outperform the baseline with no representations, their effectiveness varies depending on the task. More specifically, we observe that **motion-centric** representations (trajectory trace) achieve higher success rates in *Single Insertion* and *Cup Stack*, **object-centric** representations (bounding box) perform better in *Fruit Bowl* and *Kitchen Stack*, **pose-aware** representations (pose plans) excel in *FMB* and *Shirt Hang*, and **depth-aware** representations (trajectory trace: depth-aware) yield the highest success in *Pen Handover* and *Shoelace Tying*. These findings are consistent with our



**Fig. 8: Real-World Results.** There are clear differences in the benefits that different representations provide for tasks in the real world.

qualitative understanding of the grounding required for each task. For instance, tasks like *Fruit Bowl* and *Kitchen Stack* involve rearranging multiple objects within a cluttered scene, making object-centric representations particularly effective. In contrast, *Pen Handover* and *Shoelace Tying* necessitate a precise understanding of object relationships in 3D space, emphasizing the importance of depth-aware representations for accurate spatial reasoning and fine-grained manipulation.

Representation	Single Insertion	FMB
Bounding Box 2D	0.85	0.51
Grasp Plans	0.87	0.67
Trajectory Trace 2D	0.92	0.45
Trajectory Trace: Depth-Aware	0.97	0.57

**TABLE I: Performance Across Representations with Ground Truth.** Even with ground-truth mid-level representation, the success rates policies with different mid-level experts differ by at least 12%

The advantages of different mid-level representations become more evident when analyzing their impact with ground-truth data depicted in Table I. For *Bounding Box 2D* and *Grasp Plans*, values are directly derived from the simulation, while for *Trajectory Trace 2D* and *Trajectory Trace: Depth-Aware*, future trajectory values are estimated based on the robot arm’s current pose and the pose of the object it interacts with. Since trajectory estimation becomes more complex when multiple objects are involved, we evaluate performance in two representative tasks: *Single Insertion* and *FMB*. Our results show that in *Single Insertion*, motion-centric representations, such as trajectory traces, outperform object-centric representations (94% average success rate vs 85%. Meanwhile, in *FMB*, pose-aware representations like *Grasp Plans* yield significantly better performance 67% success rate. This highlights the importance of selecting representations that align with task-specific requirements.

*B. Mid-Level MoE can effectively utilize multiple different representations to generalize across a broad range of tasks.*

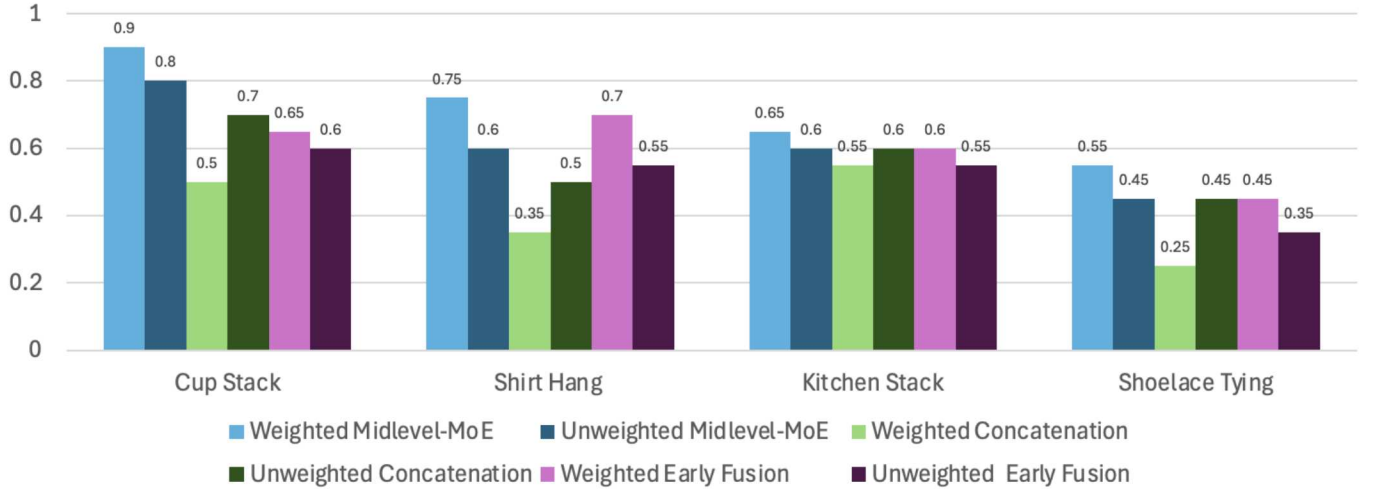
Figures 7 and 9 also compare Mid-Level MoE to a language baseline with lower-level language commands similar to [2], a keypoints baseline with object-centric points of interest similar to [25], and a simple diffusion policy baseline with no mid-level representations. Our method has an average of 69.6% success over all tasks compared to 45.1% success rate for the no-representation baseline, 51.5% success rate for the language baseline, and 58% success rate for the keypoints baseline. This highlights the usefulness of having more granular spatial grounding compared to higher-level representations like language for more dexterous tasks.

In addition, our results indicate that Mid-Level MoE can leverage the task-specific benefits of different representations across different tasks. While Mid-Level MoE doesn’t always perform better than the best representation for a particular task (such as compared to bounding boxes in the *Kitchen Stack* task), it consistently performs at a high across all evaluated tasks. Likewise, although we find some tasks that the keypoints and language baselines do well on (*Cup Stack* and *Single Insertion* respectively), since different tasks require different grounding, they don’t consistently lead to large increases in policy performance.

Method	Mid-Level MoE	Concatenation	Early Fusion
Simulation	0.78	0.72	0.65
Real	0.61	0.56	0.52

**TABLE II: Architecture Ablation** Average success rates for different architectures overs simulation and real tasks.





**Fig. 9: Architecture and Self-Consistency Ablation.** Weighted Mid-Level MoE achieves an average of 10% higher success rate than unweighted over 4 real-world tasks.

### C. Different Architectures offer Different Tradeoffs between Sensitivity and Robustness

We ablate different policy architectures in Tables II and III. Table II records the average success rates for our architecture across all simulation and real-world tasks. Meanwhile, Table II records the sensitivity scores for each of our mid-level experts as well as the robustness index. The robustness index is computed by adding Gaussian noise with standard deviation 0.1 to all of the representations and finding the ratio between the success rate with and without perturbation.

In particular, we find that our method outperforms early fusion and concatenation by an average of 11% and 7.5%, respectively. Notably, these performance enhancements are strongly correlated with the sensitivity scores observed across different architectures. Mid-Level MoE achieves an average sensitivity score of 0.2175, which is significantly lower than those of concatenation (0.2575) and early fusion (0.315). This reduction in sensitivity suggests that Mid-Level MoE is more adept at focusing on relevant mid-level representations, thereby enhancing policy performance and increasing success rates. Furthermore, our analysis of robustness indices reveals that concatenation methods exhibit lower robustness (0.75) compared to attention-based approaches, including Mid-Level MoE (0.86). This higher robustness indicates that attention-based methods are better at maintaining stable performance under varying conditions and potential perturbations. The combination of lower sensitivity and higher robustness in Mid-Level MoE not only contributes to its superior average performance but also ensures more reliable and consistent outcomes across diverse tasks.

Overall, these findings underscore the effectiveness of Mid-Level MoE in leveraging task-specific representations through attention mechanisms. By selectively attending to pertinent information and maintaining robustness, Mid-Level MoE demonstrates a balanced approach that enhances both the efficiency and reliability of policy execution across multiple

tasks.

### D. Self-Consistency Leads to Higher Sensitivity Scores

Table III demonstrates that policies trained with self-consistency exhibit increased sensitivity scores, indicating an enhanced ability to effectively leverage relevant mid-level features. In contrast, employing the Weighted Imitation Learning algorithm described in Algorithm 1 results in an average sensitivity score that is 25.5% lower than that of unweighted approaches across the three architectures. This reduction suggests that weighted imitation learning enables policies to focus more precisely on pertinent features, potentially minimizing the influence of noisy or irrelevant information.

However, the robustness indices reveal a trade-off: weighted imitation learning also leads to a 9% lower average robustness index. This decrease implies that while the policies become more selective in their feature utilization, they may be slightly less resilient to variations or adversarial perturbations in the input data. Balancing sensitivity and robustness is crucial, as excessive sensitivity can make models vulnerable to minor input changes, whereas reduced robustness might compromise performance in dynamic or unpredictable environments.

We find that policy architecture plays a crucial role in performance. Figure 5 shows results for various weighted and unweighted architectures across 4 real-world tasks. Interestingly, Weighted Mid-Level MoE has an average of 10% higher performance than Mid-Level MoE across the 4 tasks. This suggests that the benefits of more targeted feature utilization outweigh the slight decrease in robustness. However, self-consistency doesn't always improve performance. Notably, we see a decrease in performance for the weighted concatenated architecture versus unweighted (an average of 26% drop in performance). A reasonable explanation for this is a significant lower robustness index for the concatenated architecture (0.51). While self-consistency does indeed lower the sensitivity score for *Weighted Concatenation*, the policy struggles

Method	SS: Bounding Box	SS: Grasp Plan	SS: Trace 2D	SS: Trace Depth-Aware	RI
Weighted Mid-Level MoE	0.06	0.21	0.12	0.20	0.80
Unweighted Mid-Level MoE	0.1	0.36	0.16	0.25	0.86
Weighted Concatenation	0.10	0.24	0.17	0.23	0.51
Unweighted Concatenation	0.15	0.41	0.20	0.27	0.75
Weighted Early Fusion	0.14	0.30	0.20	0.27	0.73
Unweighted Early Fusion	0.19	0.47	0.25	0.35	0.80

**TABLE III: Sensitivity and Robustness.** Comparison of average sensitivity scores and robustness indices across various architectures. Policies with lower sensitivity scores and higher robustness indices tend to perform better.

with being robust to noise and spurious correlations in these representations.

## VIII. CONCLUSION

In this paper, we examine the types of grounding necessary for generalization in dexterous, bimanual robotic tasks. Our findings reveal that the effectiveness of different representations in facilitating generalization is inherently task-dependent. To address this challenge, we first identify four key representations—object-centric, motion-centric, pose-aware, and depth-aware—that achieve strong performance across a diverse set of tasks. We then introduce Mid-Level MoE, a novel policy that integrates multiple mid-level experts, each specializing in a distinct representation. Through our investigation, we uncover a fundamental tradeoff between sensitivity—the degree to which a policy adheres to a mid-level representation—and robustness—its resilience to noise within these representations. We demonstrate that applying weighted imitation learning to the Mid-Level MoE architecture effectively reduces sensitivity while preserving a high level of robustness. Ultimately, this work takes a step toward a deeper understanding of the grounding strategies necessary for scaling robot policies to broader and more complex environments.

**Limitations and Future Direction.** Our methodology comes with several limitations. First, due to frequency constraints in dexterous tasks, we distill our mid-level representations into smaller policies. Training policies to generate more generalist representations can introduce challenges related to inference speed. A promising future direction is to enable policies to condition on representations asynchronously while maintaining both high frequency and strong performance on individual tasks. In addition, our self-consistency method relies on hand-designed metrics for each task. While this is feasible for the tasks we have evaluated, scaling this approach to a broader range of tasks would require either extensive manual tuning or automated metric discovery methods.

## REFERENCES

- [1] Michael Ahn et al. Do as i can, not as i say: Grounding language in robotic affordances, 2022. URL <https://arxiv.org/abs/2204.01691>.
- [2] Suneel Belkale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debiddatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language, 2024. URL <https://arxiv.org/abs/2403.01823>.
- [3] Kevin Black, Mitsuhiro Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models, 2023. URL <https://arxiv.org/abs/2310.10639>.
- [4] Anthony Brohan et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL <https://arxiv.org/abs/2307.15818>.
- [5] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities, 2024. URL <https://arxiv.org/abs/2401.12168>.
- [6] Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuang Jiang, Stephen Marcus McAleer, Yiran Geng, Hao Dong, Zongqing Lu, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning, 2022. URL <https://arxiv.org/abs/2206.08686>.
- [7] Open X-Embodiment Collaboration et al. Open x-embodiment: Robotic learning datasets and rt-x models, 2024. URL <https://arxiv.org/abs/2310.08864>.
- [8] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning, 2020. URL <https://arxiv.org/abs/1910.11215>.
- [9] Yiming Ding, Carlos Florensa, Mariano Phielipp, and Pieter Abbeel. Goal-conditioned imitation learning, 2020. URL <https://arxiv.org/abs/1906.05838>.
- [10] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model, 2023. URL <https://arxiv.org/abs/2303.03378>.
- [11] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation, 2023. URL <https://arxiv.org/abs/2302.00111>.
- [12] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper,

- Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets, 2021. URL <https://arxiv.org/abs/2109.13396>.
- [13] Prasoon Goyal, Raymond J. Mooney, and Scott Niekum. Zero-shot task adaptation using natural language, 2021. URL <https://arxiv.org/abs/2106.02972>.
- [14] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, Priya Sundaresan, Peng Xu, Hao Su, Karol Hausman, Chelsea Finn, Quan Vuong, and Ted Xiao. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches, 2023. URL <https://arxiv.org/abs/2311.01977>.
- [15] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition, 2023. URL <https://arxiv.org/abs/2307.14535>.
- [16] Kouros Hakhmaneshi, Ruihan Zhao, Albert Zhan, Pieter Abbeel, and Michael Laskin. Hierarchical few-shot imitation with skill transition models, 2022. URL <https://arxiv.org/abs/2107.08981>.
- [17] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents, 2022. URL <https://arxiv.org/abs/2201.07207>.
- [18] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models, 2022. URL <https://arxiv.org/abs/2207.05608>.
- [19] Stephen James, Michael Bloesch, and Andrew J. Davison. Task-embedded control networks for few-shot imitation learning, 2018. URL <https://arxiv.org/abs/1810.03237>.
- [20] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning, 2022. URL <https://arxiv.org/abs/2202.02005>.
- [21] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video, 2024. URL <https://arxiv.org/abs/2410.24221>.
- [22] Alexander Khazatsky et al. Droid: A large-scale in-the-wild robot manipulation dataset, 2024. URL <https://arxiv.org/abs/2403.12945>.
- [23] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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, 2024. URL <https://arxiv.org/abs/2406.09246>.
- [24] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control, 2023. URL <https://arxiv.org/abs/2209.07753>.
- [25] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-world robotic manipulation through mark-based visual prompting, 2024. URL <https://arxiv.org/abs/2403.03174>.
- [26] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation, 2024. URL <https://arxiv.org/abs/2410.07864>.
- [27] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1113–1132. PMLR, 30 Oct–01 Nov 2020. URL <https://proceedings.mlr.press/v100/lynch20a.html>.
- [28] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics, 2017. URL <https://arxiv.org/abs/1703.09312>.
- [29] Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals, 2018. URL <https://arxiv.org/abs/1807.04742>.
- [30] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024. URL <https://arxiv.org/abs/2406.02523>.
- [31] Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning, 2024. URL <https://arxiv.org/abs/2406.11815>.
- [32] Ashutosh Saxena, Justin Driemeyer, Justin Kearns, and A. Ng. Robotic grasping of novel objects. In *Neural Information Processing Systems*, 2006. URL <https://api.semanticscholar.org/CorpusID:8682350>.
- [33] Tanmay Shankar, Shubham Tulsiani, Lerrel Pinto, and Abhinav Gupta. Discovering motor programs by recomposing demonstrations. In *Proceedings of (ICLR) International Conference on Learning Representations*, April 2020.
- [34] Pratyusha Sharma, Antonio Torralba, and Jacob Andreas. Skill induction and planning with latent language, 2022. URL <https://arxiv.org/abs/2110.01517>.
- [35] Sumedh A Sontakke, Jesse Zhang, Sébastien M. R. Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demon-



- stration is enough to learn robot policies, 2023. URL <https://arxiv.org/abs/2310.07899>.
- [36] Priya Sundaresan, Suneel Belkhale, Dorsa Sadigh, and Jeannette Bohg. Kite: Keypoint-conditioned policies for semantic manipulation, 2023. URL <https://arxiv.org/abs/2306.16605>.
  - [37] Ziyu Wang, Josh Merel, Scott Reed, Greg Wayne, Nando de Freitas, and Nicolas Heess. Robust imitation of diverse behaviors, 2017. URL <https://arxiv.org/abs/1707.02747>.
  - [38] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning, 2024. URL <https://arxiv.org/abs/2407.08693>.
  - [39] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL <https://arxiv.org/abs/2304.13705>.
  - [40] Tony Z. Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity, 2024. URL <https://arxiv.org/abs/2410.13126>.
  - [41] Hongkuan Zhou, Xiangtong Yao, Oier Mees, Yuan Meng, Ted Xiao, Yonatan Bisk, Jean Oh, Edward Johns, Mohit Shridhar, Dhruv Shah, Jesse Thomason, Kai Huang, Joyce Chai, Zhenshan Bing, and Alois Knoll. Bridging language and action: A survey of language-conditioned robot manipulation, 2024. URL <https://arxiv.org/abs/2312.10807>.