HAMSTER: HIERARCHICAL ACTION MODELS FOR OPEN-WORLD ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large models have shown strong open-world generalization to complex problems in vision and language, but they have been relatively more difficult to deploy in robotics. This challenge stems from several factors, the foremost of which is the lack of scalable robotic training data since this requires expensive on-robot collection. For scalable training, these models must show considerable transfer across domains, to make use of cheaply available "off-domain" data such as videos, hand-drawn sketches, or data from simulation. In this work, we posit that hierarchical vision-language-action models can be more effective at transferring behavior across domains than standard monolithic vision-language-action models. In particular, we study a class of hierarchical vision-language-action models, where high-level vision-language models (VLMs) are trained on relatively cheap data to produce semantically meaningful intermediate predictions such as 2D paths indicating desired behavior. These predicted 2D paths serve as guidance for low-level control policies that are 3D-aware and capable of precise manipulation. In this work, we show that separating prediction into semantic high-level predictions, and 3D-aware low-level predictions allows such *hierarchical* VLA policies to transfer across significant domain gaps, from simulation to the real world or across scenes with widely varying visual appearance. Doing so allows for the usage of cheap, abundant data sources beyond teleoperated on-robot data thereby enabling broad semantic and visual generalization. We demonstrate how hierarchical architectures trained on such cheap off-domain data can enable robotic manipulation with semantic, visual, and geometric generalization through experiments in simulation and the real world.

1 INTRODUCTION

Developing general robot manipulation policies has been notoriously difficult. With the advent of large vision-language models (VLMs) that display compelling generalizations, there is an optimism that similar techniques can be helpful for robotic manipulation. Several prior works (Team et al., 2024; Kim et al., 2024; Gu et al., 2023) build open-world vision-language-action models (VLAs) by finetuning off-the-shelf, pretrained VLMs. The recipe for training many of these VLA models has been to collect and curate a large-scale robotics-specific dataset, complete with images and corresponding on-robot actions, and then finetune a VLM to directly produce actions (Kim et al., 2024; Brohan et al., 2023a). Such VLAs have shown robustness on simple tasks and controlled environmental variations. However, these models display limited generalization in terms of environment, object, task, and semantic variation. This issue could be attributed to the scarcity of diverse, in-domain training data. The data needed to train these models is expensive since it requires end-to-end image-action pairs that must all be collected directly on-robot. A solution for training VLA models must be developed to bring down the cost of data collection or learn from easy-to-collect "cheap" sources of data.

On the other hand, relatively "small" imitation learning models have shown impressive dexterity and geometric robustness. Such models have demonstrated promise across a range of complex tasks involving contact-rich manipulation and 3D reasoning, spanning domains from tabletop manipulation (Shridhar et al., 2023; Goyal et al., 2023) to fine dexterous manipulation (Zhao et al., 2023). Trained on relatively small datasets, these models show local robustness and stable control but typically lack semantic or visual generalization. They are often brittle to changes in the envi-

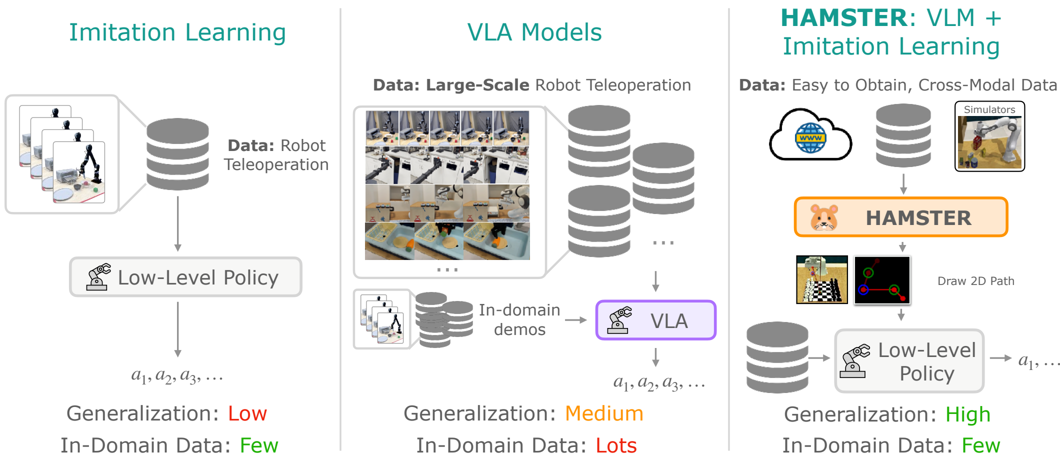

Figure 1: Overview of HAMSTER, VLAs and "smaller" imitation learning methods. HAMSTER's hierarchical design results in better generalization with a small amount of in-domain data. HAMSTER is able to utilize cheap training sources such as videos or simulations for enhanced generalization.

ronment, semantic description of the tasks, or changes in the objects being manipulated (Pumacay et al., 2024). This fragility can also be boiled down to scarce in-domain data collected on a robot. Reliable, generalizable robotic learning techniques must marry the generalization benefits of large VLMs, with the efficiency, local robustness and dexterity of small imitation learning policies, all while being able to train from abundant and cheap sources of data. In this work, we ask – can we design VLA models that train on relatively abundant and cheap data sources, showing broad visual and semantic generalization, while capturing the low-level geometric and 3D understanding displayed by small imitation learning models?

We propose that a hierarchical architecture for vision-language-actions models, HAMSTER (**H**ierarchical **A**ction **M**odels with **S**epara**TE**d Path **R**epresentations), can serve as an effective way to learn from abundant and cheap sources of data such as videos or simulation. We study a family of HAMSTERs, where finetuned VLMs are connected to low-level 3D policy learning methods via intermediate 2D path representations. Since these 2D paths can easily be obtained in abundance from data sources such as videos or simulations (either with point tracking, hand-sketching, or proprioceptive projection), these can be used to finetune the larger higher-level VLM in HAMSTER. These 2D paths can then serve as guidance for a low-level policy that operates on rich 3D and proprioceptive inputs, alleviating the burden of long-horizon planning and semantic reasoning, allowing low-level policies to focus on robustly generating precise, spatially-aware actions.

Representations similar to 2D paths has been explored in the robot learning literature (Gu et al., 2023), primarily as a technique for flexible task specification. However, the key hypothesis explored in this paper is distinct – we posit that using cheap data such as videos or simulation to finetune *hierarchical* path generating VLMs can enable a surprising degree of cross-domain transfer as compared to the direct transfer of monolithic vision-language-action models (Brohan et al., 2022; Kim et al., 2024). Here the focus is less on using paths as a scalable technique for task specification, and more on using hierarchy as a mechanism for robust cross-domain transfer across settings with considerable visual and semantic differences. Specifically, we find that VLMs trained to predict 2D path representation can transfer to the real world from simulations that look very different from the real world, or across real-world scenarios with widely varying appearance. Hence, the hierarchical design of HAMSTER provides a way to utilize cheaper, but perceptually varying sources of "off-domain" data (such as simulation or cross-embodiment data) to benefit real-world control policies.

The hierarchical design presented in HAMSTER can also offer additional advantages through the decoupling of VLM training and low-level action prediction. Specifically, since the higher-level VLM is predicting semantically meaningful trajectories from monocular RGB camera inputs, the lower-level control policies can operate from rich 3D and proprioceptive inputs. In doing so, HAMSTER inherits the semantic reasoning benefits of VLMs along with the 3D reasoning and spatial awareness benefits of 3D imitation learning policies (Goyal et al., 2024; Ke et al., 2024). Finally, since HAMSTER is built on both open-source VLMs and low-level policies, it can serve as a fully open-sourced enabler for the community-building vision-language-action models.

2 RELATED WORK

LLMs and VLMs for robotics. Early attempts in leveraging LLMs and VLMs for robotics are through pretrained language (Jang et al., 2022; Shridhar et al., 2023; Singh et al., 2023) and visual (Shah & Kumar, 2021; Parisi et al., 2022; Nair et al., 2023; Ma et al., 2023) models. However, these are not sufficient for complex semantic reasoning and generalization to the open world (Brohan et al., 2022; Zitkovich et al., 2023). Recent research has focused on directly leveraging open world reasoning and generalization capability of LLMs and VLMs, by prompting or fine-tuning them to, e.g., generate plans (Huang et al., 2022; 2023b; Lin et al., 2023; Liang et al., 2023; Singh et al., 2023; Brohan et al., 2023b), construct value (Huang et al., 2023a) and reward functions (Kwon et al., 2023; Sontakke et al., 2023; Yu et al., 2023; Ma et al., 2024; Wang et al., 2024). Our work is more closely related to the literature on VLA models, summarized below.

Monolithic VLA models as language-conditioned robot policies. Monolithic VLA models have been proposed to produce robot actions given task description and image observations directly (Brohan et al., 2022; Jiang et al., 2023; Zitkovich et al., 2023; Team et al., 2024; Kim et al., 2024; Radosavovic et al., 2023). Monolithic VLA models are often constructed from VLMs (Liu et al., 2024b; Bai et al., 2023; Driess et al., 2023; Lin et al., 2024), and are trained on large-scale robot teleoperation data (Brohan et al., 2022; Collaboration et al., 2023; Khazatsky et al., 2024) to predict actions as text or special tokens. However, due to the lack of coverage in existing robotics datasets, they must be finetuned in-domain on expensive teleoperated data. The most relevant monolithic VLA model is LLARVA (Niu et al., 2024), which predicts end-effector trajectories in addition to robot actions. However, LLARVA does not use trajectory prediction to control the robot; rather, it uses it as an auxiliary task to improve action prediction. Therefore, LLARVA still suffers from the limitations of monolithic VLA models. In contrast, our work takes a hierarchical approach, drastically reducing the number of demonstrations needed for learning downstream tasks and the number of VLM calls per episode. Moreover, our proposed hierarchical architecture enables training from cheaper data sources while enabling considerable cross-domain transfer.

VLMs for predicting intermediate representations Our work bears connections to prior methods using vision-language models for intermediate prediction. These methods can be categorized by the choice of predicted representation:

Point-based predictions: A common intermediate prediction interface has been keypoint affordances (Stone et al., 2023; Sundaresan et al., 2023; Nasiriany et al., 2024; Yuan et al., 2024). Some examples include using open-vocabulary detectors (Minderer et al., 2022), iterative prompting of VLMs (Nasiriany et al., 2024), or fine-tuning detectors to identify certain parts of an object by semantics (Sundaresan et al., 2023). Perhaps most related, Yuan et al. (2024) finetunes a VLM to predict objects of interest as well as free space for placing an object, and Liu et al. (2024a) propose a mark-based visual prompting procedure to predict keypoint affordances as well as a fixed number of waypoints. As opposed to these, our work finetunes a VLM model to not just predict points but rather entire 2D paths, making it more broadly applicable across robotic tasks.

Trajectory-based predictions: The idea of using trajectory-based task specifications to condition low-level policies was proposed in RT-trajectory (Gu et al., 2023), largely from the perspective of flexible task specification. This work also briefly discusses the possibility of combining RT-Trajectory with trajectory sketches generated from prompting a pre-trained vision language model. Complementary to RT-Trajectory, the focus of this work is less on the use of trajectory sketches for task specification, but rather on the abilities of a hierarchical VLA model to finetune the high-level VLM on cheap and abundant sources. This could include training data such as videos or simulation data, and show transfer to test scenarios of interest with considerable visual and semantic variation. While RT-trajectory uses human effort or off-the-shelf pre-trained models to generate trajectories, we show that finetuning VLM models on cheap data sources can generate more accurate and generalizable trajectories (see Table. 2). Moreover, our instantiation of this architecture enables the incorporation of rich 3D and proprioceptive information, as compared to monocular 2D policies (Gu et al., 2023).

Leveraging simulation data for training robot policies. There has been extensive work on leveraging simulation for robot learning. Simulation data is popular in reinforcement learning (RL), as RL on real robotic systems is often impractical due to high sample complexity and safety concerns (Lee et al., 2020; Handa et al., 2023; Torne et al., 2024). Recently, simulation has been also exploited to directly generate (Fishman et al., 2022) or bootstrap (Mandlekar et al., 2023) large-scale datasets for imitation learning, to reduce the amount of expensive robot teleoperation data needed. Our work takes a different approach - using simulation data to finetune a VLM, and showing that VLM is able to transfer the knowledge learned from simulation data to real robot systems, despite considerable visual differences. A related observation is recently made by (Yuan et al., 2024), but they use keypoint affordances as the interface between the VLM and the low-level policy as opposed to more general expressive 2D path representations.

3 BACKGROUND

Imitation Learning via Supervised Learning: The goal of imitation learning is to train a probabilistic policy $\pi_\theta(a \mid s, o, z)$ from an expert-provided dataset. This policy π_θ outputs the probability of producing action a conditioned on proprioceptive states s, perceptual observations o, and language instructions z that specify the task. In the typical imitation learning setting, a dataset of expert in-domain trajectories is provided, consisting of observation-action-language tuples $\mathcal{D} = \{(s_i, a_i, o_i, z_i)\}_{i=1}^N$. This dataset can be utilized to learn the parameters of the policy π_θ. While π can take on a variety of architectures with various training objectives (Goyal et al., 2023; Ke et al., 2024; Zhao et al., 2023; Chi et al., 2023), most imitation learning algorithms are trained via supervised learning to maximize the objective: $\mathbb{E}_{(s_i,a_i,o_i,z_i)\sim\mathcal{D}}\left[\log \pi_\theta\left(a_i \mid s_i, o_i, z_i\right)\right]$. This core objective can be modified with rich architectural choices such as 3D policy architectures (Goyal et al., 2023; Ke et al., 2024) or more expressive policy distribution classes (Zhao et al., 2023; Chi et al., 2023), but generalization to out-of-domain to settings with semantic or visual variations is still challenging. We study how vision-language models can be used to aid the generalization of such low-level imitation learning-based policies, discussed in Section 4.1.

Vision Language Models: Typical vision language models (VLMs) (Lin et al., 2024; Liu et al., 2024b) are large transformers (Vaswani et al., 2023) that take vision & text tokens as input and produce text responses. These models are pre-trained on large multimodal datasets (Zhu et al., 2023; Byeon et al., 2022), and then finetuned on targeted high-quality datasets (Shen et al., 2021; Lu et al., 2022). These models tokenize each modality into a shared space to produce a sequence of output tokens corresponding to text or other output modalities. In this work, we assume access to a pre-trained, text and image input VLM (Lin et al., 2024; Liu et al., 2024b), that autoregressively outputs a sequence of text tokens conditioned on an image and previous text tokens. These pretrained VLMs can typically be finetuned using a supervised prediction loss that minimizes the negative log-likelihood of the answer text tokens.

4 HAMSTER: HIERARCHICAL ACTION MODELS FOR ROBOTIC LEARNING

In this work, we examine how VLA models can be trained on relatively abundant data to demonstrate cross-domain transfer capabilities, as opposed to training on expensive image-action data collected on a robot. HAMSTER is a family of hierarchical action models designed for this purpose, exhibiting generalizable and robust manipulation. It consists of two interconnected models: first, a higher-level VLM that is fine-tuned on large-scale, cross-modal data to produce intermediate guidance (detailed in Section 4.1), and second, a low-level policy that produces actions conditioned on the VLM's predicted guidance (detailed in Section 4.2). The finetuned VLM and the low-level policy communicate using a 2D path representation. Figure 2 provides an overview of HAMSTER's design. Crucially, we study the ability of such a hierarchical design to enable training on cheap, abundantly available data such as simulation and videos.

Problem Definition. Rather than operating in the pure imitation learning setting as described in Section 3, we study a scenario where cross-domain data is utilized to train VLA models. While the typical imitation learning setting uses a dataset of optimal in-domain, on-robot tuples $\mathcal{D} = \{(s_t, a_t, o_t, z_t)\}_{t=1}^N$ to learn a near-optimal policy π_θ, in this setting we additionally assume access to a much larger dataset(s) of "off-domain" approximately optimal data $\mathcal{D}_{\text{off}} = \{(o_i^o, z_i^o)\}_{i=1}^M$, where $M \gg N$, such as video or simulation data. This "off-domain" data $\mathcal{D}_{\text{off}}$ is different from in-domain data $\mathcal{D}$ in several important ways: *1)* Off-domain perceptual observations o_i^o may be considerably different than in-domain perceptual observations o_i, even when the underlying physical state of the system is similar An illustrative example of this is the marked difference between simu-

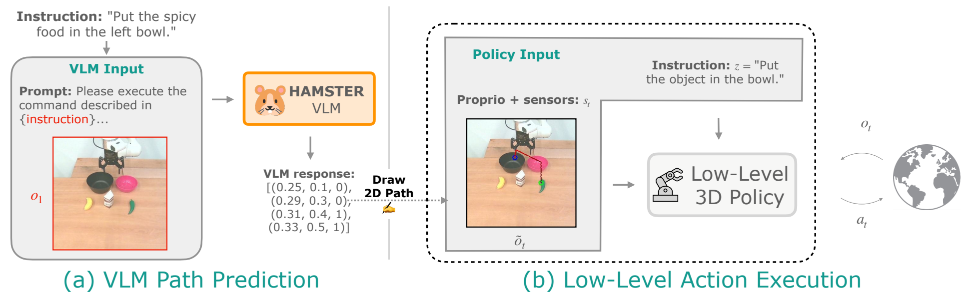

Figure 2: Depiction of HAMSTER's execution. The high-level VLM is called once to generate the 2D path. The low-level policy is conditioned on the 2D path and interacts with the environment sequentially to execute low-level actions. The path predicted by the VLM enhances the low-level policy generalization capability.

lation and real-world scene appearance (see Figure. 5). *2)* The underlying physical dynamics of the system can be potentially different, i.e the transition dynamics $\mathbb{P}(s'|s, a)$ between states s', s may be different between off-domain sources such as video or simulation than the test-time deployment setting. While the dynamics may show level differences, we assume the higher-level coarse strategies to solve the task remain invariant. *3)* Off-domain data may not have access directly to actions a or proprioception p, for instance in video based datasets. This poses challenges to directly applying the standard imitation learning paradigm for these datasets.

The goal is to leverage the combination of a small amount of "expensive" in-domain data $\mathcal{D}$ and a large amount of relatively "cheap" off-domain data $\mathcal{D}_{\text{off}}$ to obtain a generalizable policy π_θ that can be successfully deployed over various initial conditions, task variations, and visual variations in the in-domain robot environment. Without additional assumptions, this problem is arduous due to the lack of alignment between the in-domain and off-domain settings. In this work, we assume access to an intermediate *path-labeler* $p_i = h(o_i, z_i)$ at training time, that accepts an observation o_i and a language instruction z_i from either the off-domain or in-domain datasets, to produce an intermediate path label p_i that indicates *how* to optimally perform the task z_i from the observation o_i. In this work, we choose this intermediate path label p_i to be a sequence of points, a 2D path, on the image that indicates coarse end-effector motion to solve the designated task. This path-labeler at training time can come from different sources – a projection of known proprioception if available, human-drawn trajectory annotations on images, point-tracked end-effector or hand positions from video, and so on. Applying such a path labeler to the off-domain dataset yields $\tilde{\mathcal{D}_{\text{off}}} = \{(o_i^o, z_i^o, p_i^o)\}_{i=1}^M$.

4.1 HAMSTER'S VLM FOR PRODUCING 2D PATHS TRAINED FROM OFF-DOMAIN DATA

The first stage of building a HAMSTER VLA model is finetuning a high-level VLM that predicts coarse 2D paths p given a language instruction z and observation o. This path represents the approximate trajectory of the robot end-effector on the input camera image. It also contains information about the gripper state (where to open the gripper and where to close it) as subsequently explained.

Although, conceptually, any VLM can be used to predict such a 2D path by casting an appropriate prompt, we find that standard pre-trained VLMs struggle with predicting such a path in a zero-shot manner (see Table 2). Therefore, we finetune pre-trained VLMs on datasets that ground VLMs to robot scenes and path predictions collected from easier-to-obtain sources, i.e., internet visual-question-answering data, robot data from other modalities, and simulation data. The primary advantages of finetuning such a hierarchical VLM that produces intermediate representations $\hat{l}_i$, as opposed to directly producing actions a with a monolithic model (Kim et al., 2024; Zitkovich et al., 2023) are twofold: 1) the lack of actions in certain off-domain datasets (such as videos) makes it impossible to even train monolithic pixel-to-action models, 2) we find empirically that hierarchical VLMs producing intermediate cross-domain predictions generalize more effectively than monolithic VLA models.

Finetuning Objective and Datasets We use VILA-1.5-13b (Lin et al., 2024), a 13-billion-parameter vision language model trained on interleaved image-text datasets and video captioning data, as our base VLM. We then curate a multi-domain dataset to finetune this model for effective 2D path prediction. Predicting the 2D path of the end-effector requires understanding *what* objects to manipulate in a given task in terms of their pixel positions, but also reasoning about *how* a robot should perform the task. To enable this understanding, we collate a diverse off-domain dataset $\mathcal{D}_{\text{off}}$ from a wide range of modalities, including real-world data, visual question-answering data, and simulation data. Importantly, *none* of this off-domain data used to train the VLM comes from the deployment environment, thereby emphasizing generalizability. However, as outlined in Section 4.2, the predictions of this trained VLM are used to guide a low-level policy at inference time.

We assemble a dataset $\tilde{\mathcal{D}}_{\text{off}} = \{(o_i^o, z_i^o, p_i^o)\}_{i=1}^M$ of image inputs o_i^o, language prompts z_i^o, and path labels p_i^o consisting of three types of data: (1) pixel point prediction tasks (*what*); (2) simulated robotics tasks (*what and how*); (3) a real robot dataset consisting of trajectories (*what and how*). We detail each dataset below; see Figure 6 for visualization of each dataset's prompts and labels.

Pixel Point Prediction. For pixel point prediction, we use the dataset released by RoboPoint (Yuan et al., 2024) with 1.4 million VQA tasks, with most answers represented as a list of 2D points corresponding to locations on the image. A sample consists of a prompt z^o like `Find all instances of cushions`, an input image o^o and labels p^o like $[(0.25, 0.11), (0.22, 0.19), (0.53, 0.23)]$.[1] This dataset consists of data automatically generated in simulation and collected from existing real-world datasets; its diversity and tasks enable the HAMSTER VLM to reason about pixel-object relationships across diverse scenes while retaining its semantic generalization capabilities.

Robot Simulation Data. We additionally generate a dataset of simulated robotics tasks from RLBench (James et al., 2020), a simulator of a Franka robot performing tabletop manipulation for a wide array of both prehensile and non-prehensile tasks. We use the simulator's built-in planning algorithms to automatically generate successful manipulation trajectories and construct ground-truth 2D path labels p^o. Each trajectory contains a sequence of 3D coordinates of the robot's gripper in world space, as well as whether the gripper is open or closed at a given time step. We use known camera intrinsics and extrinsics to project these points on the front image and construct labels $p^o = [(x_{\text{image}}, y_{\text{image}}, \texttt{gripper\_open}), \ldots]$ where $x_{\text{image}}, y_{\text{image}} \in [0, 1]$ are relative pixel locations of the end effector's position on the image. The front camera image of the initial state forms the image input o^o and the prompt z^o for the VLM is to provide a sequence of points denoting the trajectory of the robot gripper to achieve the given instruction (see Figure 2) We generate 1000 episodes for each of 79 robot manipulation tasks in RLBench, each episode with $\sim$4 language instructions, for a total of $\sim$300k (o^o, z^o, p^o) tuples for $\tilde{\mathcal{D}_{\text{off}}}$.

Real Robot Data. Using real robot data allows us to ensure the VLM can reason about objects and robot gripper paths when conditioned on scenes, including real robot arms. We use existing, online robot datasets *not from the deployment environment* to enable this VLM ability. We source 10k trajectories from the Bridge dataset (Walke et al., 2023; Collaboration et al., 2023) consisting of a WidowX arm performing manipulation tasks and $\tilde{4}$5k trajectories from DROID (Khazatsky et al., 2024). For both datasets, we use the given end-effector trajectories and given (or estimated) camera matrices to convert robot gripper trajectories to 2D paths p^o. We use a camera image from the first timestep of each robot trajectory as o^o and a similar text prompt z^o as the simulation dataset. Note that we essentially utilize the robot data as video data, where the end effector is tracked over time. In principle, this could be done with any number of point-tracking methods (Doersch et al., 2023) on raw video as well, with no action or proprioceptive labels.

VLM Training. We finetune the HAMSTER VLM on all three datasets by randomly sampling from all samples in the entire dataset with equal weight. One problem with directly training on the path labels p^o is that many paths may be extremely long, e.g., exceeding one hundred points. Since we want the HAMSTER VLM to reason at a *high level* instead of on the same scale as the low-level control policy. Therefore, we simplify the paths p^o with the Ramer-Douglas-Peucker algorithm (Ramer, 1972; Douglas & Peucker, 1973) that reduces curves composed of line segments to similar curves composed of fewer points. We train with the standardized supervised prediction loss to maximize the log-likelihood of the language labels p^o $\mathbb{E}_{(o_i^o, z_i^o, p_i^o) \sim \tilde{\mathcal{D}_{\text{off}}}} \log \text{VLM}\left(p_i^o \mid z_i^o, o_i^o\right)$.

[1]Note that this is not a temporally ordered path, but rather simply a set of unordered points of interest in an image. We overload notation here for the sake of notational convenience.

4.2 Path Guided Low-Level Policy Learning

After training the HAMSTER VLM to predict paths, we train a low-level policy to utilize these paths to predict actions. While a low-level control policy *can* learn to solve the task without access to 2D path predictions, providing it with 2D paths can make the task easier. The paths allow the low-level policy to forgo long-horizon and semantic reasoning and focus on local and geometric predictions to produce low-level actions. As we find empirically (see Figure. 3), 2D paths allow for considerably improved visual and semantic generalization of low-level policies. We train low-level policies based on rich 3-D perceptual information, available at test time on a robotic platform with standard depth cameras. Then the question becomes—how do we incorporate 2D path information $\hat{p}$ produced by the VLM in Section 4.1 onto the 3D inputs to enable generalizable robot manipulation?

Conditioning on Paths. We convert 2D paths of the form $p = \{(x_i, y_i, \texttt{gripper\_open})\}_{t=1}^{L}$ into a format that is easy to incorporate into any language (z), proprioception (s), and image (o) conditioned policy $\pi_\theta(a \mid s, o, z)$. While one could concatenate the path with the proprioception or language input, paths are of varied lengths, and this could prevent the integration of such paths into existing policy architectures that cannot take in varied proprioceptive or language inputs. Instead, we directly draw the 2D path points onto the image input to the policy, which is not only generalizable across policy architectures but also may provide easier-to-follow path guidance as the policy does not have to learn how to associate path points with their corresponding image locations (Gu et al., 2023). During training, we use oracle paths constructed by projecting end-effector points to the camera plane as described for simulation and real robot data in Section 4.1.

Formally, we iterate through each trajectory $\tau_i = \{s_i^t, a_i^t, o_i^t, z_i)\}_{t=1}^{T}$ on the in-domain dataset $\mathcal{D}$ to obtain the path p_i. Gu et al. (2023) proposed using colored trajectories to guide a policy's actions, and we largely follow their method of coloring trajectories to indicate gripper status and progression through time. These paths are drawn onto all images in the trajectory $o_i^1 ... o_i^T$ by drawing points at each (x, y) and connecting them with line segments to obtain $\{\tilde{o}_i^t\}_{t=1}^{T}$. We use a color gradient to indicate progression through time (see Figure 2(b) for an example). We plot circles for change in gripper status: e.g., green for closing the gripper and blue for opening. This constructs the final in-domain path-labeled dataset $\mathcal{D}_{\text{path}} = \{(s_i, a_i, \tilde{o}_i, z_i)\}_{i=1}^{N}$.

Imitation Learning. Finally, we train a policy $\pi_\theta(a \mid s, \tilde{o}, z)$ conditioned on proprioception and other sensor information s, path-annotated image observations $\tilde{o}$, and a task language instruction z on $\mathcal{D}_{\text{path}}$. HAMSTER's general path-conditioning framework allows for using arbitrary lower-level control policies as they do not need to condition on the same inputs as the VLM. Therefore, we train 3D low-level policies, such as RVT-2 (Goyal et al., 2024) and 3D-DA (Ke et al., 2024), for low-level control. Here, we assume s includes additional sensor information (i.e., depth), which 3D-DA and RVT-2 utilize to construct point clouds and virtual camera renderings, respectively, for more accurate control and data-efficient imitation learning. We directly train these policies, with no necessary architectural changes,[2] with their supervised imitation learning objectives on $\mathcal{D}_{\text{path}}$ to maximize log-likelihoods of the dataset actions: $\mathbb{E}_{(s_t, a_t, \tilde{o}_t, z_t) \sim \mathcal{D}_{\text{path}}} \log \pi_\theta(a \mid s_t, \tilde{o}_t, z_t)$. For further implementation details, see Appendix B.

Online Evaluation Standard VLA architectures query the VLM for every low-level action (Kim et al., 2024; Brohan et al., 2023a), which can be very expensive with large VLMs—for example, OpenVLA's 7B param VLA only runs at 6Hz on an RTX 4090 (Kim et al., 2024). Instead, HAMSTER's hierarchical design allows us to query the VLM just once at the beginning of the episode to generate a 2D path $\hat{l}$ that we draw onto every subsequent image.[3] Therefore, HAMSTER can be scaled to large VLM backbones without needing end-users to be concerned about inference speed.

5 Experimental Evaluation

To test the hypotheses proposed in Section 4, we perform empirical evaluations in both simulation and the real world. The experiments primarily aim to answer the following questions: (1) do hierarchical VLA models enable behavioral generalization to unseen scenarios? (2) do hierarchical VLA

[2] We ignore the language instruction for RVT-2 it is already encoded in the 2D path.

[3] HAMSTER is not inherently limited to being queried once per episode, but for simplicity and computational efficiency we query just once per episode in our experiments.

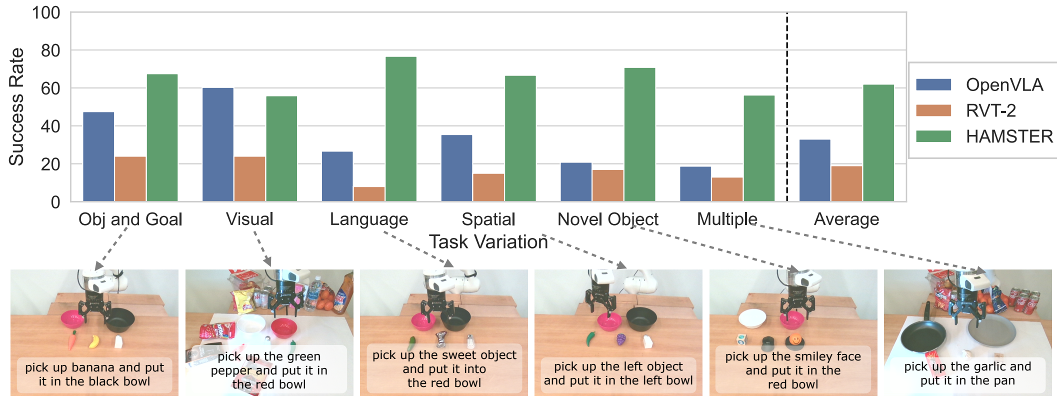

Figure 3: Depiction of quantitative real-world policy execution results on a real-world robot, evaluated across different axes of generalization. Across all generalization axes, HAMSTER outperforms monolithic VLAs and 3D imitation learning policies.

models show more effective cross-domain generalization than monolithic VLA models or low-level imitation learning methods? (3) is behavior learned by hierarchical VLA models robust to significant degrees of visual and semantic variations? (4) does including cross-domain data from settings like simulation really help with model generalization? (5) does explicitly finetuning the high-level VLM yield benefits in terms of spatial and semantic reasoning?

5.1 Real World Evaluation on Tabletop Manipulation

Our real-world evaluation experiments aim to test the generalization capability of hierarchical VLA models across significant semantic and visual variations. In particular, we consider a variant of HAMSTER that uses a VLM (VILA-1.5-13b) finetuned on the data mixture in Section 4.1 as the high-level predictor, with a 3-D policy architecture - RVT-2 (Goyal et al., 2024) as the choice of low-level policy, as described in Section 4.2. The low-level 3D policy is trained with 320 episodes collected via teleoperation directly on the table-top manipulation setup shown in Fig. 6. Importantly, the high-level VLM in HAMSTER is not finetuned on any in-domain data and is directly transferred only from the cheap data sources described in Section 4.1. This suggests that any generalization that the VLM sees does not result from in-domain training data rather than from cross-domain transfer.

Baseline comparisons: We compare HAMSTER to a state-of-the-art monolithic VLA, OpenVLA (Kim et al., 2024), as well as a non-VLM 3D imitation learning policy. For fair comparison, we finetune OpenVLA on the collected in-domain trajectory data described above, since OpenVLA showed poor zero-shot generalization to the testing domain. For the 3D imitation learning policy, we use RVT-2 (Goyal et al., 2024) as our baseline, as it is effective in learning robust policies with few demonstrations. The RVT-2 baseline is trained with the same teleoperation data used to train the low-level policy in HAMSTER but without the intermediate 2D path representation from HAMSTER's VLM.

Figure 3 summarizes our real-world results. We compile results for multiple tasks, including 'pick and place', 'push buttons', and 'knock down objects'. Similar to prior works (Kim et al., 2024), we test generalization across various axes, including: *Arrangement:* novel rearrangement of seen objects; *Visual:* visual generalization with pose changes in the target object and changes in table texture, lighting, and distractors; *Language:* semantic generalization with unseen language instructions to describe the task and target objects; *Novel Object:* testing scenarios with objects that are unseen in the training data; and lastly combinations of multiple of the above factors applied simultaneously. In total, we do 132 tests across all the models.

We find that HAMSTER is able to outperform monolithic VLA models as well as 3D imitation learning methods. This is significant because this improved performance is in the face of considerable visual and semantic changes in the test setting, showing the ability of HAMSTER to transfer much more effectively than monolithic VLA models or non-VLM base models. We refer readers to the supplementary website for additional details on the failure modes and evaluation conditions.

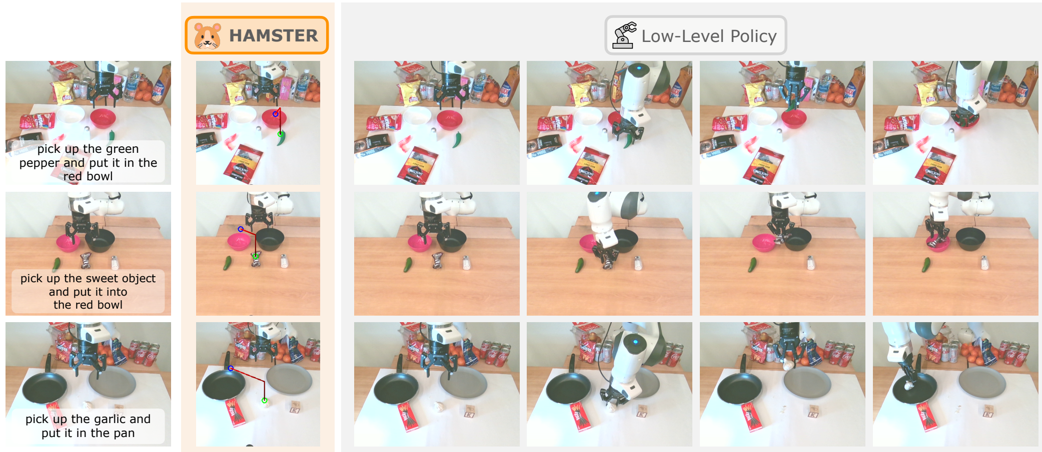

Figure 4: Examples of real-world executions of HAMSTER demonstrate its strong performance in novel complex scenes, achieved by leveraging the generalization capabilities of VLMs and the robust execution of low-level 3D policies.

	Avg.	no var	bac tex	cam pos	distractor	lig col	man obj col	man obj siz
3D-DA[Ke et al.]	0.44	**0.53** $\pm$ 0.04	0.37 $\pm$ 0.07	**0.45** $\pm$ 0.38	0.28 $\pm$ 0.24	0.62 $\pm$ 0.30	0.55 $\pm$ 0.36	0.33 $\pm$ 0.21
HAMSTER (w 3D-DA)	**0.52**	0.48 $\pm$ 0.32	**0.46** $\pm$ 0.07	0.41 $\pm$ 0.11	**0.43** $\pm$ 0.10	**0.66** $\pm$ 0.20	**0.61** $\pm$ 0.10	**0.66** $\pm$ 0.25

	man obj tex	rec obj col	rec obj siz	rec obj tex	rlb and col	rlb var	tab col	tab tex
3D-DA[Ke et al.]	**0.54** $\pm$ 0.18	0.3 $\pm$ 0.08	0.26 $\pm$ 0.16	0.45 $\pm$ 0.18	0.08 $\pm$ 0.01	0.59 $\pm$ 0.33	0.37 $\pm$ 0.23	**0.41** $\pm$ 0.19
HAMSTER(w 3D-DA)	0.46 $\pm$ 0.12	**0.44** $\pm$ 0.17	**0.29** $\pm$ 0.14	**0.59** $\pm$ 0.2	**0.12** $\pm$ 0.04	**0.65** $\pm$ 0.2	**0.68** $\pm$ 0.13	0.38 $\pm$ 0.11

Table 1: Simulation evaluation of HAMSTER across different visual variations. We test vanilla 3D Diffuser Actor and HAMSTER across variations in Colosseum (Pumacay et al., 2024) and find that HAMSTER generalizes more effectively than 3D Diffuser Actor. Avg. indicates mean across variations, including no variation. For details about each variation, please refer to Pumacay et al. (2024).

5.2 Simulation Evaluation

We also perform controlled experiments in simulation. We use Colosseum (Pumacay et al., 2024) as the benchmark as it displays considerable visual and semantic variations. In simulation, we paired our high-level VLM, with 3D Diffuser Actor (Ke et al., 2024) as the low-level policy, since this is one of the state-of-the-art models on RLBench. We compare HAMSTER with a vanilla 3D Diffuser Actor implementation without path guidance. Table. 1 summarizes our results in simulation. HAMSTER significantly outperforms vanilla 3D-DA (0.43 vs 0.52). This shows that the 2D paths produced by the VLM in HAMSTER can help low-level policies to generalize better to novel unseen variations. We refer readers to Colosseum (Pumacay et al., 2024) for details on the variations.

5.3 Impact of Design Decisions on VLM performance

To better understand the transfer and generalization performance of the proposed hierarchical VLA model, we analyze the impact of various decisions involved in training the high-level VLM. We conduct a human evaluation of different variants of a trained high-level VLM on a randomly collected dataset of real-world test images, as shown in Figure 5. We ask each model to generate 2D path traces corresponding to instructions such as “move the block on the right to Taylor Swift” or “screw the light bulb in the lamp” (the full set is in Appendix C.1). We then provide the paths generated by each method to human evaluators who have not previously seen any of the models’ predictions. The human evaluators rank the predictions for different methods and the average rank across the samples is reported in Table 2.

We evaluate the following VLM models (listed in the order seen in Table 2): (1) zero-shot state-of-the-art closed-source models such as GPT-4o using a similar prompt to ours (shown in Figure 7) (2) zero-shot state-of-the-art closed-source models such as GPT-4o but using Code-as-Policies (Liang et al., 2023) to generate paths as described in Gu et al. (2023) (3) finetuned open-source models (VILA-1.5-13b) on the data sources described in Section 4.1, but excluding the simulation trajectories from the RLBench dataset, (4) finetuned open-source models (VILA-1.5-13b) on the data

Method	VLM	Finetuning Data	Rank Exc. Real RLB.	Rank Real RLB.	Rank All
RT-Traj.	0-shot GPT4-o	-	3.40	3.63	3.52
RT-Traj.	CaP GPT4-o	-	3.57	3.36	3.46
HAMSTER	VILA	Our Exc. Sim RLB.	1.78	2.39	2.11
HAMSTER	VILA	Our	**1.59**	**1.28**	**1.42**

Table 2: Ranking-based human evaluation of different VLMs, averaged across various real-world evaluation tasks. Results indicate that HAMSTER including simulation data is most effective since it captures both spatial and semantic information across diverse tasks from RLBench. This significantly outperforms zero-shot VLM-based trajectory generation, as described in Gu et al. (2023)

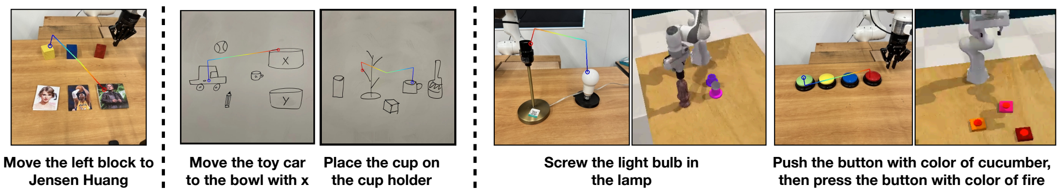

Figure 5: HAMSTER's VLM demonstrates considerable generalization and cross-domain learning to scenarios not encountered in the training set. From left to right: (a) it can effectively utilize world knowledge to generalize to tasks specified by people; (b) it generalizes to highly out-of-domain input images, such as human-drawn sketches; (c) when trained on diverse simulated data it shows transfer to related, but visually distinct tasks in the real world.

sources described in Section 4.1, including path sketches from the RLBench dataset. The purpose of these evaluations is to first compare with closely related work that generates 2D trajectories using pretrained closed source VLMs Gu et al. (2023) (Comparison (1) and (2)). The comparison between (3) and (4) (our complete method) is meant to isolate the impact of including the simulation path sketches from the RLBench dataset. In doing so, we analyze the ability of the VLM to predict intermediate paths to transfer across significantly varying domains (from RLBench to the real world).

The results suggest that: (1) zero-shot path generation, even from closed-source VLMs Gu et al. (2023) such as GPT-4o with additional help through Code-as-Policies (Liang et al., 2023), underperforms VLMs finetuned on cross-domain data as in HAMSTER; (2) inclusion of significantly different training data such as low-fidelity simulation during finetuning improves the real-world performance of the VLM. This highlights the transferability displayed by HAMSTER across widely varying domains. These results emphasize that the hierarchical VLA approach described in HAMSTER can effectively utilize diverse sources of cheap prior data for 2D path predictions, despite considerable perceptual differences.

6 Conclusion and Limitations

In summary, HAMSTER studies the potential of hierarchical VLA models, achieving robust generalization in robotic manipulation. It consists of a finetuned VLM that accurately predicts 2D paths for robotic manipulation and a low-level policy that learns to generate actions using the 2D paths. This two-step architecture enables visual generalization and semantic reasoning across considerable domain shifts, while enabling data-efficient specialist policies, like ones conditioned on 3D inputs, to perform low-level action execution.

This work represents an initial step towards developing versatile, hierarchical VLA methods, with numerous opportunities for future improvement and expansion. The proposed work only generates points in 2D space, without making native 3D predictions. This prevents the VLM from having true spatial 3D understanding. Moreover, the interface of just using 2D paths is a bandwidth limited one, which cannot communicate nuances such as force or rotation. In the future, investigating learnable intermediate interfaces is a promising direction. Moreover, training these VLMs directly from large-scale human video datasets would also be promising.

REFERENCES

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Adeola Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin D. Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll L. Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report. In *arxiv preprint*, 2023. URL `https://arxiv.org/pdf/2303.08774`.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023a.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pp. 287–318. PMLR, 2023b.

Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`, 2022.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu (eds.), *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL `https://doi.org/10.15607/RSS.2023.XIX.026`.

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, 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, 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.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10061–10072, 2023.

David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica*, 10(2):112–122, 1973. doi: 10.3138/FM57-6770-U75U-7727. URL `https://doi.org/10.3138/FM57-6770-U75U-7727`.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, pp. 8469–8488. PMLR, 2023.

Adam Fishman, Adithyavairavan Murali, Clemens Eppner, Bryan Peele, Byron Boots, and Dieter Fox. Motion policy networks. In Karen Liu, Dana Kulic, and Jeffrey Ichnowski (eds.), *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pp. 967–977. PMLR, 2022. URL `https://proceedings.mlr.press/v205/fishman23a.html`.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pp. 694–710. PMLR, 2023.

Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt2: Learning precise manipulation from few demonstrations. *RSS*, 2024.

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.

Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5977–5984. IEEE, 2023.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pp. 9118–9147. PMLR, 2022.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning*, pp. 540–562. PMLR, 2023a.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, pp. 1769–1782. PMLR, 2023b.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

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. In *Conference on Robot Learning*, pp. 991–1002. PMLR, 2022.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. In *International Conference on Machine Learning*, 2023.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. 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 Martín-Martín, 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. 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

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. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26689–26699, June 2024.

Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365, 2023.

Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. *arXiv preprint arXiv:2403.03174*, 2024a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024b.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2023.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, pp. 1820–1864. PMLR, 2023.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pp. 728–755. Springer, 2022.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pp. 892–909. PMLR, 2023.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. In *International Conference on Machine Learning*, 2024.

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. In *8th Annual Conference on Robot Learning*, 2024. URL `https://openreview.net/forum?id=Q2lGXMZCv8`.

Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *international conference on machine learning*, pp. 17359–17371. PMLR, 2022.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. *arXiv preprint arXiv:2402.08191*, 2024.

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.

Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972. ISSN 0146-664X. doi: https://doi.org/10.1016/S0146-664X(72)80017-0. URL `https://www.sciencedirect.com/science/article/pii/S0146664X72800170`.

Rutav M Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. In *International Conference on Machine Learning*, pp. 9465–9476. PMLR, 2021.

Zejiang Shen, Kyle Lo, Lucy Lu Wang, Bailey Kuehl, Daniel S. Weld, and Doug Downey. Incorporating visual layout structures for scientific text classification. *ArXiv*, abs/2106.00676, 2021. URL `https://arxiv.org/abs/2106.00676`.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pp. 785–799. PMLR, 2023.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530. IEEE, 2023.

Sumedh Anand Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Biyik, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. In *NeurIPS*, 2023.

Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Sean Kirmani, Brianna Zitkovich, Fei Xia, et al. Open-world object manipulation using pre-trained vision-language models. In *Conference on Robot Learning*, pp. 3397–3417. PMLR, 2023.

Priya Sundaresan, Suneel Belkhale, Dorsa Sadigh, and Jeannette Bohg. Kite: Keypoint-conditioned policies for semantic manipulation. In *Conference on Robot Learning*, pp. 1006–1021. PMLR, 2023.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *Robotics: Science and Systems*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL `https://arxiv.org/abs/1706.03762`.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. In *International Conference on Machine Learning*, 2024.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montserrat Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. In *Conference on Robot Learning*, pp. 374–404. PMLR, 2023.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction in robotics. In *8th Annual Conference on Robot Learning*, 2024. URL `https://openreview.net/forum?id=GVX6jpZOhU`.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu (eds.), *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.016. URL `https://doi.org/10.15607/RSS.2023.XIX.016`.

Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. Multimodal C4: An open, billion-scale corpus of images interleaved with text. *arXiv preprint arXiv:2304.06939*, 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.

For extended supplementary details and results, please see `https://sites.google.com/view/hamster-iclr`.

A VLM Finetuning Dataset Details

Pixel Point Pred Data. Our point prediction dataset comes from Robopoint (Yuan et al., 2024). Most data in our point prediction dataset contains labels given as a set of unordered points such as $p^o = [(0.25, 0.11), (0.22, 0.19), (0.53, 0.23)]$. However, data in RoboPoint also contains answers that are instead in natural language for VQA queries such as "what is the person feeding the cat?" We keep these data as is because these VQA queries are likely to benefit a VLM's semantic reasoning an visual generalization capabilities; we fine-tune HAMSTER's VLM on the entire Robopoint dataset as given.

Simulation Data. We selected 79 RLBench tasks out of 100 to generate data by removing the tasks with poor visibility on the `front_cam` view in RLBench. We use the first image in each episodes combined with each language instruction. THe final dataset is around 320k.

Real Robot Data. For the Bridge Walke et al. (2023) dataset, which only provides RGB images, we extract trajectories by iteratively estimating the extrinsic matrix for each episode. In each scene, we randomly sample a few frames and manually label the center of the gripper fingers. Using the corresponding end-effector poses, we compute the 3D-2D projection matrix with a PnP (Perspective-n-Point) approach. We then apply this projection matrix to the episodes and manually check for any misalignments between the projected gripper and the actual gripper. Episodes exhibiting significant deviations are filtered out and start a new round to estimate their extrinsic.

For DROID (Khazatsky et al., 2024), a large portion of the dataset contains noisy camera extrinsics information that do not result in good depth alignment. Therefore, we filter out trajectories with poor-quality extrinsics as measured by the alignment between the projected depth images and the RGB images. This results in ∼45k trajectories (∼22k unique trajectories as trajectories each have 2 different camera viewpoints) which we use for constructing the VLM dataset $\mathcal{D}_{\text{off}}$ as described in Section 4.1.

B Implementation and Architecture Details

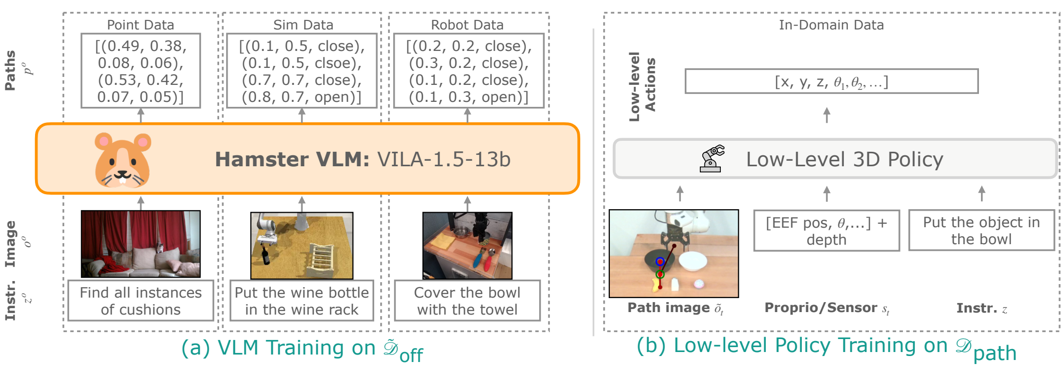

Figure 6: (a): Examples of training data in $\tilde{\mathcal{D}}_{\text{off}}$ used to train HAMSTER's VLM. (b): The data used to train HAMSTER's low-level policies.

B.1 VLM Implementation Details

VLM Prompt. We list the prompt for both fine-tuning on sim and real robot data and evaluation in Figure 7. We condition the model on an image and the prompt, except when training on Pixel Point Prediction data (i.e., Robopoint (Yuan et al., 2024) data) where we used the given prompts from the dataset.

HAMSTER Prompt

In the image, please execute the command described in ⟨quest⟩{quest}⟨/quest⟩. Provide a sequence of points denoting the trajectory of a robot gripper to achieve the goal. Format your answer as a list of tuples enclosed by ⟨ans⟩ and ⟨/ans⟩ tags. For example: ⟨ans⟩[(0.25, 0.32), (0.32, 0.17), (0.13, 0.24), ⟨action⟩Open Gripper⟨/action⟩, (0.74, 0.21), ⟨action⟩Close Gripper⟨/action⟩, ...]⟨/ans⟩ The tuple denotes point x and y location of the end effector of the gripper in the image. The action tags indicate the gripper action. The coordinates should be floats ranging between 0 and 1, indicating the relative locations of the points in the image.

Figure 7: The full text prompt we use to train HAMSTER with on simulation and real robot data (Section 4.1). We also use this prompt for inference.

VLM Trajectory Shortening. As mentioned in Section 4.1, one problem with directly training on the path labels p^o is that many paths may be extremely long. Therefore, we simplify the paths p^o with the Ramer-Douglas-Peucker algorithm (Ramer, 1972; Douglas & Peucker, 1973) that reduces curves composed of line segments to similar curves composed of fewer points. We run this algorithm on paths produced by simulation and real robot data to generate the labels p^o for $\mathcal{D}_{\text{off}}$. We use tolerance $\epsilon = 0.05$.

VLM Training Details. We train our VLM, VILA1.5-13B Lin et al. (2024), on a node equipped with eight NVIDIA A100 GPUs, each utilizing approximately 65 GB of memory. The training process takes about 30 hours to complete. We use an effective batch size of 256 and a learning rate of 1×10^{-5}. During fine-tuning, the entire model—including the vision encoder—is updated.

B.2 Low-level Policy Training Details

We train RVT2 (Goyal et al., 2024) and 3D-DA (Ke et al., 2024) as our lower-level policies. We keep overall architecture and training hyperparameters the same as paper settings. Specific details about how the inputs were modified other than the 2D path projection follow.

For low-level policy training, we train the policies on ground truth paths constructed by projecting trajectory end-effector points to the camera image. In order to also ensure the policies are robust to possible error introduced by HAMSTER VLM predictions during evaluation, we add a small amount of random noise ($N(0, 0.01)$) to the 2D path (x, y) image points during training to obtain slightly noisy path drawings.

RVT2. We remove the language instruction for RVT-2 when conditioning on HAMSTER 2D paths.

3D-DA. No changes such as removing language were performed for 3D-DA's inputs as we saw a performance drop when removing language from the tasks.

C Extended Results

C.1 VLM Real World Generalization Study

The full list of task descriptions for this study is below (see Section 5.3 for the main experiment details). Duplicates indicate different images for the same task. We plot some additional comparison examples in Figure 8. Note that the path drawing convention in images for this experiment differ from what is given to the lower-level policies as described in Section 4.2 as this multi-colored line is easier for human evaluators to see.

1. screw in the light bulb on the lamp
2. screw in the light bulb on the lamp
3. screw in the light bulb on the lamp
4. screw out the light bulb and place it on the holder

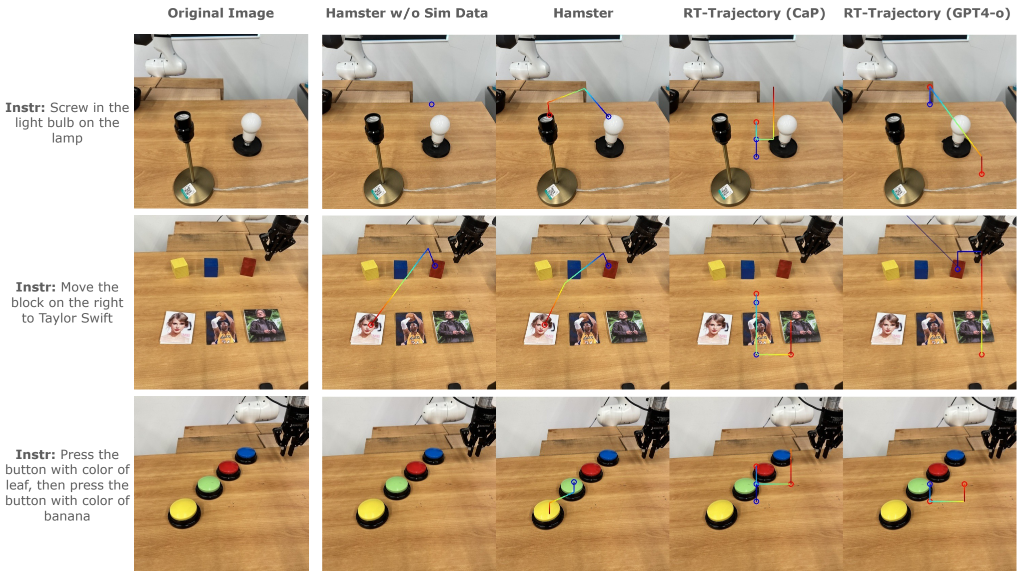

Figure 8: Human VLM evaluation example images and instructions along with corresponding trajectories from HAMSTER without any finetuning on (RLBench) simulation data, HAMSTER finetuned on all the data in Section 4.1, RT-Trajectory (Gu et al., 2023) with Code-as-Policies (Liang et al., 2023) powered by GPT-4o (Achiam et al., 2023), and RT-Trjaectory powered by GPT-4o directly.

5. screw out the light bulb and place it on the holder
6. screw in the light bulb
7. screw in the light bulb on the lamp
8. move the blue block on Taylor Swift
9. pick up the left block and put it on Jensen Huang
10. move the block on the right to Taylor Swift
11. place the yellow block on Kobe
12. pick up the blue block and place it on Jensen Huang
13. move the red block to Kobe
14. press the button on the wall
15. press the button to open the left door
16. press the button to open the right door
17. open the middle drawer
18. open the bottom drawer
19. open the top drawer
20. open the middle drawer
21. open the bottom drawer
22. press the button
23. press the button
24. press the orange button
25. press the orange button with black base
26. press the button
27. pick up the SPAM and put it into the drawer
28. pick up the orange juice and put it behind the red box

29. pick up the tomato soup and put it into the drawer
30. pick up the peach and put it into the drawer
31. move the mayo to the drawer
32. move the dessert to the drawer
33. pick up the object on the left and place it on the left
34. pick up the fruit on the left and put it on the plate
35. pick up the milk and put it on the plate
36. press the button with the color of cucumber, then press the button with color of fire
37. press the button with color of banana
38. press the button with color of leaf
39. press the button with color of leaf, then press the one with color of banana
40. press left button
41. pick up the left block on the bottom and stack it on the middle block on top
42. make I on top of C
43. put number 2 over number 5
44. stack block with lion over block with earth
45. pick up the left block on the bottom and stack it on the middle block on top
46. stack the leftest block on the rightest block
47. stack the block 25 over block L
48. put the left block on first stair