Vision-Language Model for Generalized 3D Robotic Manipulation

Pranay Junare, Aditya Bidwai, Samra Huseynova
Team name : Pickachu
junar002@domain, bidwa001@umn.edu, husey012@umn.edu

Abstract

Generalized language-conditioned robotic manipulation remains a key challenge in the integration of natural language in robotics or embodied AI. While previous approaches have leveraged vision-language models (VLMs) like Contrastive Language-Image Pretraining (CLIP), they often struggle with generalization across diverse tasks and environments. In this project, we focus on improving language-conditioned robotic manipulation across varied 3D tasks, object types, and environments. We use Sigmoidal Language-Image Pre-Training 2 (SigLIP2), a more recent VLM, as our vision-language backbone and condition a transformer-based policy network to predict robot actions. The model is trained in an end-to-end fashion using Imitation Learning. The main objective of our project is to improve generalization across varied 3D tasks, object types, and environments, advancing the robustness of VLM-guided robotic manipulation. We want to understand whether multi-modal learning can help in performing complex 3D manipulation tasks. Refer our project website.

Research Areas: Embodied AI, Vision-Language Models(VLMs), VLAs, Multi-modal learning, Imitation learning, RLBench, Robot learning, etc

1 Motivation

Recent advancements in large-scale foundational models for vision, language, and action have proven to provide generalization by reasoning about perception and action in a unified manner. Furthermore, foundational models for language (BERT, RoBERTa, GPT, LLaMA) and VLMs/VLA (CLIP, ALIGN, OpenVLA, RT-X, Flamingo) have demonstrated remarkable capabilities in extracting semantic information and excelling at high-level reasoning and planning tasks. Moreover, recent developments in Imitation learning and generalist policies have opened up new frontiers for the foundational robotics model at the intersection of language, vision, and action. We want to build upon these recent developments and architect a comprehensive methodology to unify Language, Vision and Action for various robot manipulation tasks.

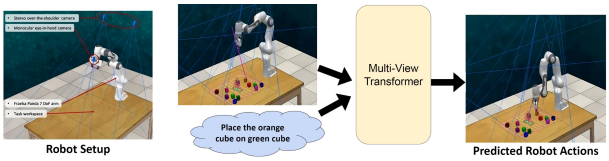

Figure 1: Task: Language conditioned object stacking.

2 Problem Statement

Humans can easily understand and perform actions like 'scooping coffee beans', 'placing objects in a drawer', 'turning the right water tap', or 'folding a cloth' without needing explicit calculations about the shape, size, pose, or movement of the objects. We just intuitively know how to do it based on our past experiences. Robots, on the other hand, tend to struggle with this. They usually need precise models, poses, and structures of the objects and often fail at generalization. The challenge is: How can we teach robots to understand abstract language instructions coupled with visual input to predict actions required to precisely complete multiple real-world tasks? Moreover, an important thing to note here is, inorder to predict 3D robot actions, we want to jointly reason over multi-modal input of vision and language, instead of traditional discrete architectures where language is processed independently to extract semantics of the task and then vision is used to determine the spatial structure of the scene to perform robot action.

3 Literature Review

Seminal work CLIPort (Shridhar et al., 2021) integrates CLIP's (Hafner et al., 2021) broad semantic understanding with Tranporter Network's (Zeng

et al., 2021) spatial precision to enable robots to perform various manipulation tasks specified via natural language instructions. The model learns from demonstrations and generalizes well to unseen tasks without explicit object pose estimation. CLIPort exhibits strong data efficiency and adaptability, but its reliance on 2D feature extraction might limit performance in complex 3D environments.

3D-LOTUS (Garcia et al., 2025) is a new work that extends the vision-language paradigm by incorporating 3D scene representations for action prediction. The model is evaluated on GemBench, a benchmark for assessing generalization in manipulation tasks. Although 3D-LOTUS outperforms previous methods in familiar tasks, it does struggle with new scenarios. To address this, 3D-LOTUS+ (Garcia et al., 2025) integrates task planning with large language models and improves object detection through vision-language models, achieving state-of-the-art generalization.

RVT (Goyal et al., 2023) overcomes the computationally expensive voxel-based 3D object manipulation by a scalable multi-view transformer-based approach. Some key features of RVT are an attention mechanism to aggregate information across views and re-rendering of the camera input from virtual views around the robot workspace. RVT explicitly reasons over the 3D scenes and trains almost 26x faster while achieving 26% higher success rate when compared to the PerAct. RVT-2 (Goyal et al., 2024) the successor of RVT archives 6x faster training time and 82% success rate by incorporating a series of architectural and system-level improvements such as multi-stage transformer pipeline, parameter optimization, and a custom point-renderer.

On the other hand, SAM2Act (Fang et al., 2025) proposes an architecture for tasks that specifically require spatial memory. It uses a memory-based architecture inspired by SAM2 (Ravi et al., 2024), which incorporates a memory bank, an encoder, and an attention mechanism to enhance spatial memory. The paper also introduces a novel benchmark MemoryBench, to assess spatial memory and action recall in robotic manipulation.

Lack of generalization to varying conditions is one of the challenges faced in robotic manipulation that has been addressed in HAMSTER (Li et al., 2025) by implementing a hierarchical VLA (Vision-Language-Action) model. Unlike monolithic VLA models that directly predict actions by fine-tuning VLMs, HAMSTER improves generalization by completing the task in 2 steps. Instead of fine-tuning VLMs for direct action prediction, HAMSTER gets 2D path predictions from VLM as an output and feeds them as input to models like RVT-2 and 3D-DA (Ke et al., 2024) for completion of low-level policy actions. This approach eliminates the need to teach VLM fine-grained control, reducing reliance on in-domain data and improving generalization. ARP+ (Zhang et al., 2024), on the other hand forms a model that enhances the adaptability of the system for both high and low frequency control tasks.

Similarly, the aim of PerACT2 (Grotz et al., 2024), which is a language-conditioned imitation learning model, is to extend and innovate the RLBench task set by introducing bimanual tasks.

In most cases manipulator interactsan with rigid objects, however research work such as (Deng et al., 2024) brings variation by replacing rigid objects with deformable ones. Language commands, graph and image embeddings are passed as inputs to the CLIP integrated transformer based model. Although the object variation is limited and model is not tested in real world scenarios, this paper considerably contributes to gaps in research field related to deformable object manipulation while also exploring the capabilities of relatively new simulation framework Softgym for robotic manipulation. Moreover, it is strictly restricted to the unimanual robot arm thus further restricting its applicability to the limited number of tasks.

4 Novelty

This work intends to improve existing approaches by addressing their limitations identified during the literature review.

Exploring a New Vision-Language Model (VLM): Many recent works utilize state-of-the-art VLM models, such as *CLIP* and *VILA* (Wu et al., 2024). However, a new VLM model, *SigLIP2* (Tschannen et al., 2025) is an improved version of CLIP, where the contrastive loss function has been replaced with a pairwise sigmoid loss. The application and evaluation of this model in robotic manipulation remain unexplored, presenting an opportunity to contribute significantly to the field.

Adaptability to Multiple tasks and in-task variations: A significant limitation of current approaches is the lack of noticeable variations in task complexity, objects, and orientations.

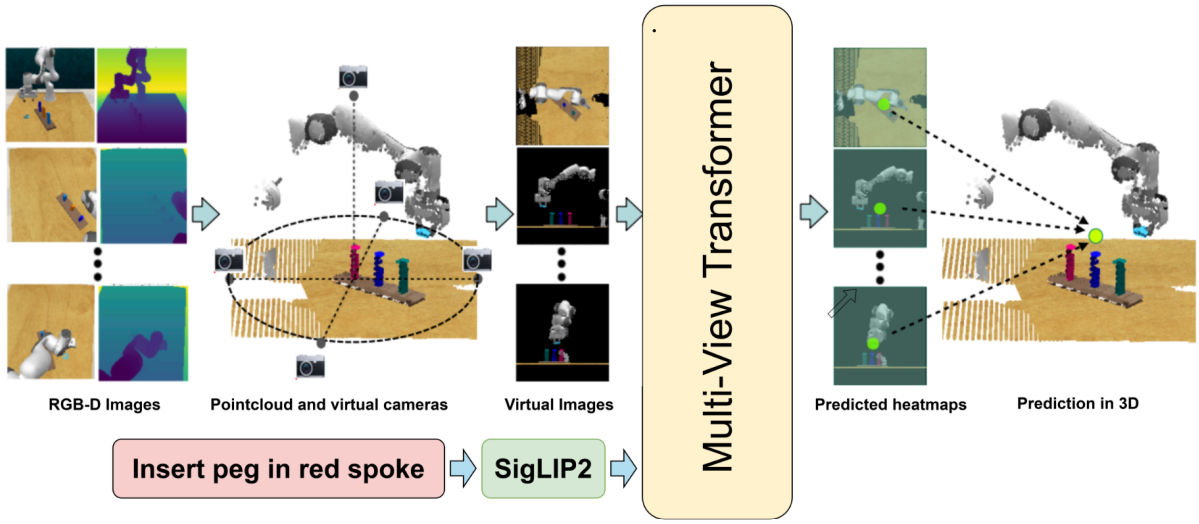

Figure 2: Network Architecture.

Most existing task sets and interactive objects remain largely the same across different studies. This raises concerns about whether robots are truly learning generalizable policies from demonstrations. Thus, the development of an approach that can adapt to at various task variations variations—such as lighting, camera viewpoints, spatial changes in the object pose, is important. Our approach thus evaluates its robustness across multiple task variations. Moreover, our method uses a single learning pipeline and can adapt to various multiple tasks without any architectural changes, meaning we only need to re-train our multi-modal model on other task demonstration datasets without any changes to the perception, learning and control parts of the robotic system.

5 Methods

Our project seeks to address the challenges and limitations of existing research in language-guided generalized robotic manipulation.(Ref. Figure-2 for complete model architecture.) Our first step is to develop a system capable of handling unimanual tasks, such as placing objects into a cupboard/box. In the next phases, we plan to extend the model by incorporating a dataset from all 18 RLBench tasks. The main contribution and challenge of this work lies in integrating these varied tasks into a single multi-task policy network, rather than using separate architectures for each individual task. Multi-task 3D policies have become a highly significant area of research within the robotics community in recent years, owing to the large-scale developments in multi-modal learning.

While many approaches attempt to achieve multi-task generalization by employing a unified action representation, these methods are often limited to the same task category or closely related task domains. Our goal is to develop a single policy capable of handling a diverse set of tasks with large domain variations. To accomplish this, we propose incorporating natural language as an additional modality, alongside the visual input from four cameras, to guide the manipulation across various tasks. Our method takes, Language and Vision as input and jointly learns to predict the action output. The model outputs an 8-dimensional action, including the 6-DoF target end effector pose (3-DoF for translation and 3-DoF for rotation), 1-DoF gripper state (open or close), and a binary indicator for whether to allow collision for the low-level motion planner. Another thing that we aimed to explore is the recent SigLIP2 model's performance over the baseline CLIP model. SigLIP2, pre-trained on millions of image-caption pairs from the internet, offers a robust prior for grounding semantic concepts commonly shared across tasks, such as categories, parts, shapes, colors, text, and other visual attributes. Furthermore, combined visual and language will be input to the multimodal Transformer encoder proposed in ViT. (Dosovitskiy et al., 2020) and the model is trained end-to-end using Imitation learning.

Our multi-view transformer architecture is based on RVT-2(Goyal et al., 2024). RVT-2 takes the current scene and a task instruction, and predicts the next key-frame pose. It consists of two stages. The first stage uses fixed virtual views around the robot to predict the area of interest. The second stage

uses zoomed-in views from the area of interest to predict the gripper pose. To predict the key-frame pose, RVT-2, similar to RVT, first reconstructs a point cloud of the scene using the input RGB-D images. The scene is then rendered from virtual cameras along orthogonal directions. While RVT renders five virtual views, including the top, front, left, back, and right views, RVT-2 renders only three views, ie front, top, and right, and does not sacrifice the performance. RVT-2 adopts a multi-stage design, where, in the first or coarse stage, it predicts the area of interest using a fixed set of views. RVT2 then zooms in on the area of interest and re-renders images around it. These virtual images are then passed to a multi-view transformer model that jointly reasons over all the views.

5.1 Dataset

We trained our proposed model on the pre-generated RLBench (simulations) demonstrations open-sourced by (Shridhar et al., 2022). The authors have divided this dataset into train, validate, and test, containing 100, 25, and 25 episodes, respectively. Due to the limitations on compute power, we trained our model architecture on one single task dataset, out of the available 18 task demonstration datasets. Since each task demonstration contains images of the scene from multiple different views the dataset size even for single demonstration, goes up to 100s of GBs.

5.2 Tasks

For our project, we will be evaluating the model's performance on RLBench's 18 tasks in simulation. In real-world experiments, we first plan to work on the task of pick and place. Specifically, our first representative task will be picking the specified object and placing it in the cupboard. Later on if the time permits, we plan to implement RLBench's subsequent tasks in real-world.

5.3 Benchmark Models for RLBench 18 Tasks

The performance of the proposed approach will be compared with the results obtained by the following benchmark models for RLBench 18 tasks:

- **SAM2Act** (Fang et al., 2025): SAM2Act has achieved the best performance on RLBench tasks. It integrates CLIP, one of the benchmark Vision-Language Models (VLMs), into its architecture. However, the performance of this model has only been evaluated on unimanual RLBench tasks. The Colosseum simulator is used for evaluation.

- **ARP/ARP+ (Autoregressive Policy)** (Zhang et al., 2024): ARP is a recent benchmark model that applies imitation learning for training. The core component of this model is a chunking-causal transformer that incorporates vision features obtained from ResNet50 (for Push-T and ALOHA tasks) and a Multi-View Transformer (for RLBench tasks). However, its performance has not been evaluated on bimanual RLBench tasks. Additionally, the model architecture does not utilize any VLMs, which are two key differences from the methodology we aim to apply.

- **RVT-2** (Goyal et al., 2023): RVT-2 is another high-performing benchmark model for RLBench tasks. Its architecture integrates CLIP as a VLM model; however, like SAM2Act, this model has only been evaluated in a unimanual setup.

- **PerACT2** (Grotz et al., 2024): The proposed architecture's performance on bimanual tasks can be compared against PerACT2, which has been adapted and tested specifically for bimanual RLBench tasks.

5.4 Simulation framework

We plan to use the RLBench simulation environment build on top of CoppeliaSim physics engine, along with PyRep and YARR for the initial model implementation and evaluation purposes.
CopeliaSim: CoppeliaSim (formerly V-REP) is a robot simulation software used for research, education, and industrial applications. It provides physics-based simulations for robotic tasks and is widely used for reinforcement learning (RL) and robotic control. Suitable for robot manipulation, autonomous navigation, and RL training in simulated environments.
PyRep: PyRep (James et al., 2019) is a high-level Python API for CoppeliaSim, making it easier to interact with the simulation environment programmatically. It is a toolkit for robot learning research, built on top of CoppeliaSim.
RLBench: RLBench(James et al., 2020) is a large-scale benchmark and learning environment designed to facilitate research in a number of vision-guided manipulation research areas, including rein-

forcement learning, imitation learning, multi-task learning, geometric computer vision, and in particular, few-shot learning.
YARR: YARR is "Yet Another Robotics and Reinforcement" learning framework for PyTorch. The framework allows for asynchronous training (i.e. agent and learner running in separate processes), which makes it suitable for robot learning.

5.5 Hardware

To train our model, we were planning to use the resources allocated on the MSI (Minnesota Supercomputer Institute) as a part of the course. But unfortunately, students were not given access to the MSI and we had to restrict to our own PCs. Fortunately, one of us had access to the GeForce RTX Nvidia 4090 but still we were heavily restricted to using only a subset of the dataset. Nevertheless, our method validates true and can be scaled even when a large number of task demonstrations are present in the dataset.

6 Experiments

The model was initially trained on 25 variations of the "put groceries into cupboard" task. In each episode, the orientations of the objects and the cupboard are modified to ensure better generalization. The robot must select a specified item from among eight objects and place it on a specified shelf (upper, middle, or bottom). Training ran for 15 epochs, and the accompanying graphs show the rotation, collision, gripper, and total losses.

Each training step begins by computing a translation loss that measures how accurately the network's predicted heatmap matches the ground-truth pick location. The model's raw logits over every pixel are compared to a one-hot (or Gaussian-blurred) target using standard cross-entropy, then averaged across all images and spatial locations to yield a single scalar. When the extra "RGC" heads are enabled, three additional cross-entropy losses are computed on the discretized Euler angles (x, y, z), one on the binary gripper open/close prediction, and one on the binary collision-ignore flag; each of these takes the appropriate slice of the network's logits, compares it to integer class labels derived from quaternions or boolean flags, and averages over the batch. Finally, all terms, including the translation plus any rotation, gripper, and collision losses, are summed without weighting to produce the total loss, which is backpropagated, stepped by the optimizer, and logged at each iteration. All of the loss curves exhibit exponential-decay pattern over the 15 epochs. Gripper loss goes to 0 almost after 5 epochs, which shows that model is quick to learn this open/close gripper skill. Collision loss starts around 0.34 and steadily falls to 0.04, indicating the network learns to predict "ignore collision" flags at a somewhat slower but still consistent pace. Coming to the rotation loss, y-axis drops fastest, whereas loss about x and z axis are a bit harder to nail down than y. Train loss is the sum of translation, rotation, gripper, and collision terms, it falls smoothly from about 11 down to 5.8, mirroring the component-wise declines. Tensorboard uses EMA (exponential moving average) to give the smoothed graph of results over all epochs. The dark blue lines on graphs stand for this, whereas pale blue lines show the actual per epoch loss.

During evaluation, the robot runs each trial in the simulator under the same conditions as training, and the simulator assigns a numerical score at every step based on task-specific success criteria. At the end of each run, the final score indicates task completion (a perfect score denotes full success). These end-of-trial scores are collected over many episodes to compute overall success rates, average episode lengths, and other summary statistics; any trial whose final score exceeds the success threshold counts as a successful attempt.

7 Results

We evaluated the model in two ways on the same task. When trained on "put groceries into cupboard" with simple language instructions, it achieved an 84 % success rate. In that setting, the model consistently handled all the commands 'leave X out' and 'do not put anything', but struggled whenever it had to pick up and place something (especially coffee or tuna in the cupboard) and also had difficulty with some "move X to bottom shelf" tasks (soup and strawberry jello). In a second evaluation, we issued more challenging commands that require understanding of shape or material, such as 'Put everything in spherical shape in the cupboard now'. The model struggled with these and instructions that contained grammatical errors or abstract rules (for example, 'pick the item whose name starts with the letter 'c'). In some failure cases the model recognized the correct object but lacked the precision to grasp and place it properly. However, when given a color-based command like 'Put the item

that is the same color as the sky', it successfully identified and retrieved the appropriate object.

To test generalization beyond pick-and-place groceries, we evaluated the same model on a "stack blocks" task, which requires picking up a cube and placing it on another cube. Despite simple language instructions, the model's success rate was 0 %. It could recognize and pick the correctly colored cube but failed to place it, because the manipulator has been trained only to look for a cupboard after grasping, however, it found no cupboard in the environment and could not complete the instruction.

8 Error analysis

Previous models trained on a variety of pick-and-place tasks generalize better to novel tasks. Our model's failures stem primarily from being trained on a single task. Increasing the variety of training tasks would likely improve performance, but since training on just one task already required 15 hours, we chose to keep the project scope narrow. As a future improvement, we can introduce a curriculum of related pick-and-place tasks with varying object shapes, sizes, materials, and receptacle types to teach the model to handle novel dynamics and geometries. Second, adding synthetic domain randomization (random textures, lighting, and camera poses) and procedural scene generation would expose the network to a wider range of visual contexts without dramatically increasing training time. Third, multi-task learning, where the agent jointly optimizes for several objectives such as grasp stability, object recognition, and goal grounding, could produce more robust internal representations. We could also leverage pretrained vision-language models (e.g. CLIP) to bootstrap semantic understanding of object attributes and more complex language instructions. To improve precision, integrating affordance-based grasp planning and fine-grained reward shaping for placement accuracy may help the gripper learn subtler manipulations. Together, these steps would build on our initial insights and help close the gap toward truly flexible, language-conditioned robotic pick-and-place. Despite having all these difficulties, this work helped us to gain valuable insight into common failure and success modes that can be addressed in future research.

Task	Train ep.	Test ep.	RVT-2	% Success
Place groceries into cupboard	100	25	21/25	**84%**

Table 1: Evaluation results

# of epochs	15
# of RGBD cameras	4 (front, top, right, gripper-mounted)
Batch size	4
Optimizer	LAMB
Learning rate	1.25 x $10^{-}5$

Table 2: Training hyperparameters.

Sample Language Instructions for Task: Place the object in the cupboard

Success Cases:

- **Instruction 1:** "Pick the object that has same color with ocean and place it into the cupboard"
- **Instruction 2:** "Pick the cracker and put it into the cupboard."

Failure Cases:

- **Instruction 1:** "Pick item whose name starts with "c" and place it into cubpoard".
- **Instruction 2:** "Pick item in pyramid shape and place it into the middle shelf"

The reason for the above failure cases can be due to the architecture of the model. The agent learns to map language embeddings to object classes, which means that it has not been trained to get notion of first letter as in the case of Instruction-1. In addition, it doesn't have the ability to detect the shapes or textures as well. As a result, when asked for a "pyramid," it has no learned representation to tell it which object in the scene matches that shape descriptor. That is why it simply defaults to whatever pick-point heatmap it can produce, which ends up at the wrong object.

9 Replicability

All our code, simulations, modifications, along with training scripts, configuration files, and preprocessing steps, are documented on our GitHub repository. The results are reproducible both in simulation and on any platform with a similar sensor setup. To make things easier, we have hosted

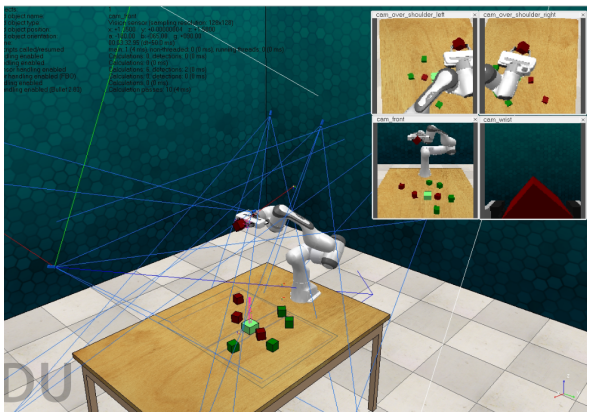

Figure 3: Figure 5: Task: Stack red block on green block

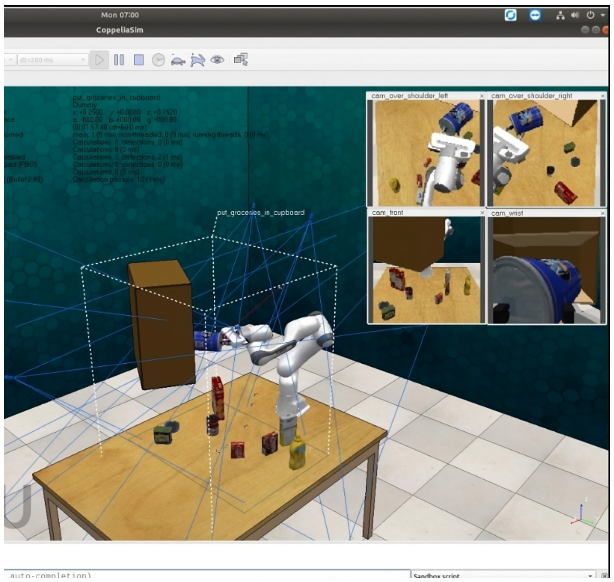

Figure 4: Task: Place blue box in the cupboard

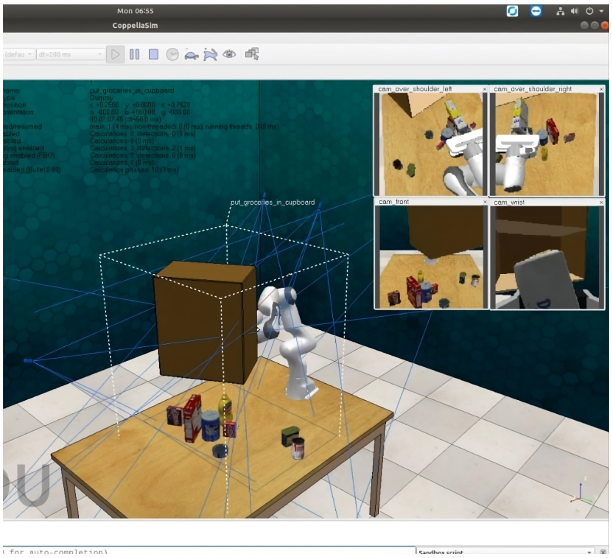

Figure 5: Task: Place cracker box in the cupboard

all relevant details - including setup instructions, model checkpoints, and visual demos - on our public project website.

10 Ethics

The deployment of our work (and VLMs in general) in robotic manipulation introduces ethical and societal concerns that merit careful attention. While these systems offer strong generalization and flexibility across tasks, their dependence on large-scale pre-trained vision-language models brings certain safety, fairness, and accountability risks. Key concerns include:

- **Bias Propagation**: VLMs are trained on internet-scale datasets that may carry implicit societal biases - such as associating specific tools or roles disproportionately with certain genders or contexts. When a robot interprets vague commands like "*pick a suitable object*", it might prioritize choices influenced by those patterns rather than the actual task needs. To address this, it's important to regularly evaluate model outputs in manipulation tasks for bias and incorporate filtering mechanisms that enforce fairness and task-specific constraints.
- **Robustness and Safety**: These models can fail when faced with ambiguous, adversarial, or out-of-distribution prompts. In the real world, this can lead to unsafe or unintended physical actions. Safety can be improved by introducing action validation layers that enforce environmental or mechanical constraints and designing human-in-the-loop systems to supervise execution during critical moments.
- **Privacy Implications**: Since our VLM-based robotic system often relies on constant visual data streams, deploying it in homes or workplaces raises privacy and surveillance concerns. With proper precautions and measures, these issues can be mitigated by ensuring on-device, local processing of sensor inputs rather than sending them to external servers.
- **Labor Market Impact**: As VLM-driven systems like ours grow more capable, they may automate a wider range of manipulation tasks, potentially affecting manual labor jobs across industries. While this is a broader automation concern, responsible design, such as keep-

ing humans in supervisory or decision-making roles, can help retain human agency and reduce displacement.

By proactively integrating such safeguards, VLM-based robotic systems can be developed and deployed in a way that is both effective and ethically aligned.

11 Discussions

11.1 Limitations

Although our model surprisingly works very well across various in-task generalization, it fails to generalize to tasks on which it was not trained on. For example, the model trained only on object placing in the cupboard task, does not perform well on the bottle opening task. This is attributed to the large domain shift between the two task categories. Moreover, our model also fails to generalize to the large variations in language commands. This could be attributed to the fact that CLIP/SigLIP/SigLIP2 often struggle with compositional understanding of complex instructions. In addition, our model also fails to accommodate a large domain shift in language style. This is because CLIP and its variants are trained on web-scraped captions that differ significantly from the concise, imperative commands used in robotics. As a result, synonyms or unusual phrasings (“pop the top” vs “open the bottle”) may lie far apart in embedding space, causing misinterpretations.

11.2 Future research

Bimanual Language-conditioned Robotic Manipulation: Most existing works employ a *single* robotic manipulator for training and evaluation. Only a few papers explore a bimanual setup, with *PerACT2* being considered the primary benchmark for dual-arm robotic manipulation (Grotz et al., 2024). One of our future research goal will be to incorporate bimanual tasks and analyze the performance of our model in comparison to PerACT2. Evaluating how well existing robust models adapt to the presence of a second manipulator and manage seamless cooperation while avoiding any collisions would be interesting.

Long-Horizon Tasks: Another direction of research could be, to extend the scope of our project to include long-horizon manipulation tasks. For example, tasks such as "Pick up the spoon, go to the bowl, then pick up the food present in a bowl using the spoon, now take another scoop.", Chaining such long-horizon tasks in still an active area of research in embodied AI and robot learning. Successfully implementing the above objectives will provide the necessary insights and capabilities to further contribute to this area.

12 Appendix

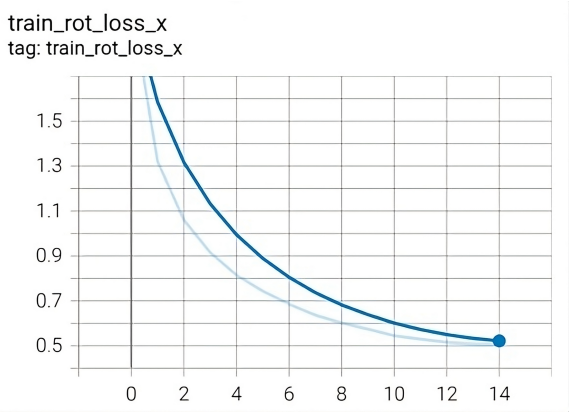

Figure 6: Rotation Loss - X

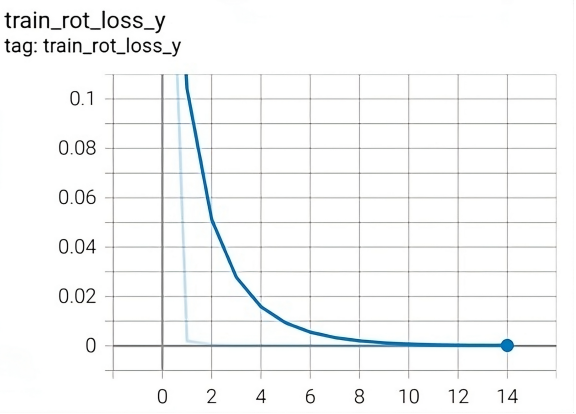

Figure 7: Rotation Loss - Y

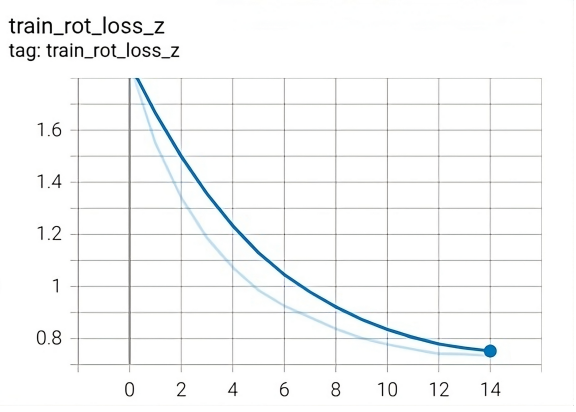

Figure 8: Rotation Loss - Z

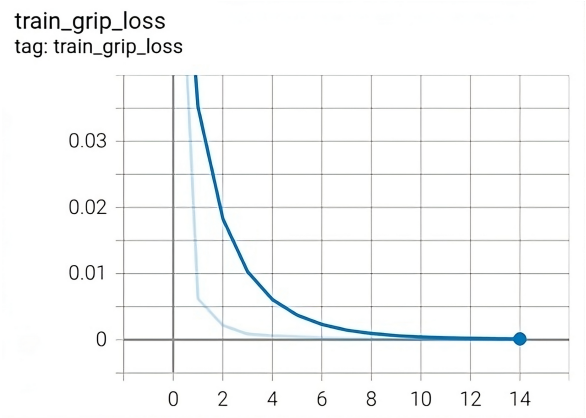

Figure 9: Gripper Loss

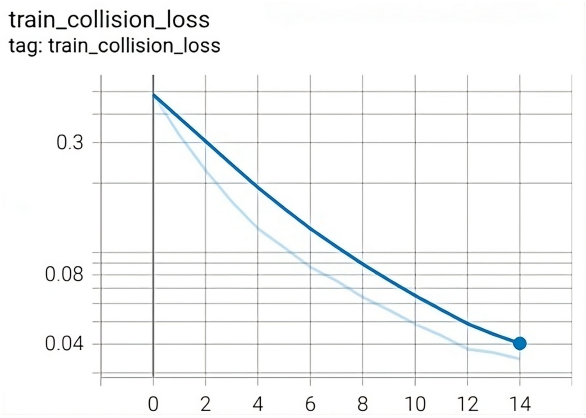

Figure 10: Collision Loss

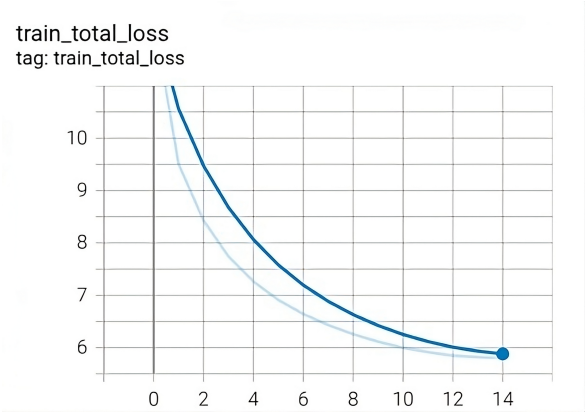

Figure 11: Total Loss

References

Yuhong Deng, Kai Mo, Chongkun Xia, and Xueqian Wang. 2024. Learning language-conditioned deformable object manipulation with graph dynamics. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7508–7514. IEEE.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. 2025. Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation. *arXiv preprint arXiv:2501.18564*.

Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. 2025. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. 2024. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*.

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

Markus Grotz, Mohit Shridhar, Yu-Wei Chao, Tamim Asfour, and Dieter Fox. 2024. Peract2: Benchmarking and learning for robotic bimanual manipulation tasks. In *CoRL 2024 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*.

Markus Hafner, Maria Katsantoni, Tino Köster, James Marks, Joyita Mukherjee, Dorothee Staiger, Jernej Ule, and Mihaela Zavolan. 2021. Clip and complementary methods. *Nature Reviews Methods Primers*, 1(1):20.

Stephen James, Marc Freese, and Andrew J. Davison. 2019. Pyrep: Bringing v-rep to deep robot learning. *arXiv preprint arXiv:1906.11176*.

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

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 2024. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memme, Raymond Yu, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, et al. 2025. Hamster: Hierarchical action models for open-world robot manipulation. *arXiv preprint arXiv:2502.05485*.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer.

2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2021. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2022. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*.

Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. 2024. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. 2021. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR.

Xinyu Zhang, Yuhan Liu, Haonan Chang, Liam Schramm, and Abdeslam Boularias. 2024. Autoregressive action sequence learning for robotic manipulation. *arXiv preprint arXiv:2410.03132*.