Hierarchical Vision-Language-Action Policies for Global Reasoning in Assembly Tasks

Moritz Hesche[1,2] **Moritz Reuss**[1] **Marc Forstenhäusler**[2] **Rudolf Lioutikov**[1]

[1]Intuitive Robots Lab, Karlsruhe Institute of Technology, Germany
[2]Carl Zeiss AG, Germany

Abstract: The rapid development of robotic control policies holds great potential for automating industrial assembly, especially for small-batch production, where traditional methods are not cost-effective. To achieve the necessary flexibility, robots must interpret the same multimodal assembly instructions used by human workers, which typically combine text with visual elements like technical drawings to convey spatial and semantic information. While existing policies primarily focus on a single instruction modality at a time, we specifically address the challenge of combining information from complementary text *and* sketch instructions for robot action prediction. To this end, we propose a hierarchical framework where a fine-tuned vision-language model (VLM) acts as a high-level planner, converting multimodal instructions into a sequence of symbolic subtasks in the context of the environment observation. Crucially, each symbolic subtask is spatially grounded by a corresponding mask predicted over the robot's visual observation. The modular design improves interpretability and adaptability, and by offloading instruction understanding to the VLM planner, it reduces the data annotation burden on the manipulation policy. We validate our approach on a novel benchmark designed for quantifying multimodal instruction execution capabilities and conduct careful ablation studies to assess the impact of key design decisions.

Keywords: Planning, Hierarchical Models, Vision-Language-Models, Imitation Learning

1 Introduction

Learned robot policies are poised to enable the autonomous execution of complex industrial assembly processes, a capability particularly valuable for small-batch production and scenarios with frequently changing configurations where traditional automation is economically unviable. A key obstacle, however, to transition with minimal effort from human-centered assembly lines to fully automated ones lies in enabling robots to understand existing human-tailored assembly instructions. These instructions are often multimodal, combining textual descriptions with visual elements like sketches or diagrams to convey both high-level semantics and crucial spatial information that is difficult to express in language alone. While the integration of pretrained vision and language foundation models into robot policies [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] has driven significant progress, they are generally still not capable of interpreting the unique, combined format of industrial assembly instructions.

Current goal-conditioned policies often operate on a single input modality, such as a language command [11, 12, 13, 14, 15, 16] or a goal image [17, 18, 6, 19, 20], making them ill-suited for the inherently multimodal nature of real-world assembly documents. Some architectures support multiple modalities [21, 2, 9], but typically process only one at a time, failing to jointly reason over complementary information from different modalities. This gap is compounded by the lack of large-scale robotic datasets annotated with such multimodal instructions, which are necessary to train

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

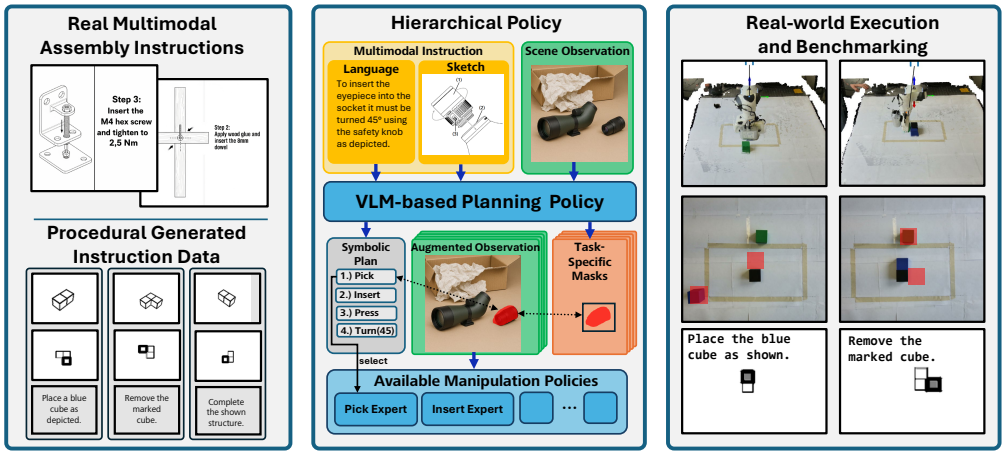


Figure 1: An overview of our work, from motivation to method and validation. **(Left)** We are motivated by real-world multimodal assembly instructions that combine sketches and text. **(Middle)** Our hierarchical policy uses a VLM planner to translate these instructions into a symbolic plan, grounded by spatial masks, for a downstream manipulation policy. **(Right)** The system is then validated on a custom benchmark in a real-world block rearrangement task.

robust end-to-end policies. To bridge this gap, we propose a hierarchical framework that explicitly decouples high-level, multimodal instruction understanding from low-level, physical robot control. Our core contribution lies in the specific interface between these stages: we use a pretrained VLM to predict a sequence of symbolic subtask representations which are each parameterized by a predicted spatial mask. The generated subtasks are executed sequentially by a downstream manipulation policy whose 3D point cloud observations are augmented with the subtask-specific projected masks, effectively creating a task-conditioned visual input. This method of guiding a manipulation policy through direct observation augmentation allows for integrating the capabilities of pretrained foundation models for complex instruction following without requiring architectural changes to the manipulating policy. The proposed way of decoupling also improves data efficiency: the manipulation policy no longer needs to be trained on robot trajectories paired with complex, varied multimodal instructions, but only on a more structured and compact set of subtasks conditioned on mask-augmented observations. A high-level overview of our proposed framework is illustrated in Figure 1.

To summarize, our primary contributions are:

- A hierarchical framework where a VLM-based planner translates multimodal (text and sketch) instructions into an intermediate representation that is used for conditioning inputs for a secondary manipulation policy.
- An interpretable intermediate representation consisting of a sequence of symbolic subtasks (e.g., pick, place) spatially grounded in the robot's current observation via predictive masks, effectively translating abstract instructions into an actionable format.
- A benchmark and data generation pipeline for evaluating the ability of robotic systems to follow multimodal assembly instructions, addressing a key gap in existing evaluation resources.

2 Related Work

Our work is related to three key areas of robotic learning: hierarchical policies that use foundation models for planning and a separate policy for robot action execution, methods that allow multimodal goal specification, and visuomotor control from 3D point cloud observations.

Hierarchical Policies with Foundation Models A common paradigm in robot learning is to use a hierarchical structure that decouples high-level reasoning from low-level control [22, 23, 24, 25, 26, 8, 27, 28]. While many works use foundation models as powerful feature encoders for a downstream policy [24, 1, 19], we instead focus on a class of models where a foundation model acts as an explicit high-level planner. This planner stage translates a complex, human-friendly instruction into a more beneficial representation for the manipulation policy. A common strategy is to have a Vision-Language Model (VLM) generate symbolic or textual plans, such as sequences of skill IDs [29, 28, 30, 8, 10, 28], executable programming code [26, 25] or natural language motion descriptions [22, 31, 13]. More related to our method are approaches where the intermediate representation is inherently visual. For instance, prior works have used VLMs to predict or classify points of interest in the observation space [23, 27, 32] or generate 2D visual traces of an intended trajectory [33, 34]. Our approach builds on this paradigm by using a VLM to generate a plan that is both symbolic and visually grounded via spatial masks.

Visual and Multimodal Goal Specification. While language is a common modality for goal description [11, 12, 35, 19, 36, 37], its ambiguity can be limiting for tasks requiring precise spatial instructions. Conversely, policies can be conditioned on visual goals, such as images of a target state [18, 6, 38] or user-drawn sketches [17, 39], to provide the necessary spatial detail in the conditioning input. Despite this advantage, these visual-only goals lack the expressiveness to convey more abstract task information. By using modality-specific encoders [21, 2, 19, 40, 10] for a set of different goal modalities, some policies allow a user to choose the most suitable format for a given task. However, these systems are typically restricted to processing only one of their supported modalities at a time. This design prevents these models from processing multimodal instructions where different modalities are complementary and must be understood jointly. The regime of robot policies that can jointly process such complementary instructions is less explored, though notable exceptions like VIMA [9] can handle interleaved text and image prompts. Our work contributes to this area by using a VLM as a high-level planner that simultaneously processes both visual instruction sketches and textual descriptions together with the environment observation to output a sequence of structured, spatially-grounded subtask representations for a downstream manipulation policy.

3D Visuomotor Policies A recent trend in visuomotor learning is the shift from 2D images to 3D representations to provide policies with explicit geometric information, reducing the burden of inferring spatial relationships from 2D data. We adopt this paradigm and use a 3D policy for our manipulation stage. State-of-the-art models manage the high dimensionality of 3D data with various strategies, such as using feature point clouds [14, 12], sparse voxel sampling [15] or multi-view 2D-to-3D lifting [16, 37]. More recently, the direct integration of 3D data into large language models [41] has enabled powerful end-to-end 3D-VLAs [6].

3 Method

We propose a hierarchical framework that decouples high-level instruction understanding from low-level action execution to enable robots to interpret and execute multimodal assembly instructions combining text and sketches. Our system consists of two main components: (1) a VLM-based planning policy that translates multimodal instructions into structured symbolic plans with spatial grounding, and (2) a manipulation policy that executes these plans utilizing mask-augmented observations.

3.1 Intermediate Representation

A key design choice for hierarchical policies is the interface between the high-level planning stage and the lower-level manipulation policy. The intermediate representation connecting these stages acts as an information bottleneck and must carefully balance the trade-off between preserving all relevant information from the original instruction and simplifying the control problem for the manipu-

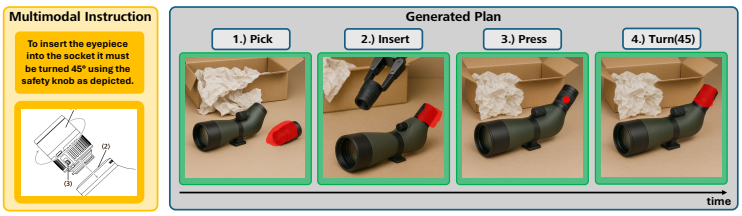


Figure 2: The planner generates a plan of symbolic subtask IDs, each associated with a mask (red overlay) on the current observation to guide the manipulation policy.

lation policy. To this end, our representation consists of a sequence of subtasks $\tau = [\tau_1, \tau_2, \ldots, \tau_n]$, where each subtask τ_i is a tuple containing:

- A symbolic subtask identifier (e.g., pick, place, insert) from a predefined finite set $\mathcal{T}$. This abstracts away the high variability of natural language, simplifying the learning problem for the manipulation policy.
- A mask $m_i \in \{0,1\}^{H \times W}$ overlaid on the robot's observation image. These masks provide explicit spatial grounding, localizing the object to be manipulated or target location, thereby parameterizing the symbolic subtask.

Figure 2 visualizes a generated four-step plan for an example assembly task, showing the sequence of symbolic subtasks and their corresponding visual groundings. By combining a symbolic ID with a visual mask, the proposed representation converts the rich spatial information from multimodal instructions into a direct, actionable signal for the manipulation policy. The decomposition into a finite set of primitives is particularly well-suited for industrial settings, where complex assembly processes often consist of recurring subtasks (e.g., pick-and-place, insertion, screwing) for which specialized and reliable expert policies can be developed. Furthermore, because the representation is fully human-interpretable, it offers a clear advantage for debugging and human oversight compared to opaque, latent vector-based interfaces.

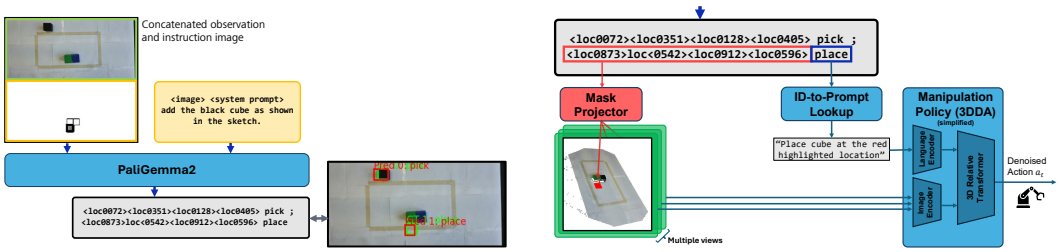
3.2 Planning Policy

The planning policy translates the high-level, multimodal instruction and current scene observation into the intermediate representation described above.

Model We use PaliGemma2 [42], a versatile open-source VLM, as the backbone for our planner. It features a SigLIP vision encoder [43] and a Gemma language model [44], and critically, its vocabulary contains special tokens to represent location-aware predictions like bounding boxes or masks that refer to regions in the input image.

Input and Output The planner takes three inputs: the current RGB scene observation, the instruction sketch, and the textual instruction. Figure 3a provides a visual overview of the planner's inputs and its generated output. To handle the two visual inputs with a standard single-image VLM, we concatenate the observation and sketch images vertically before tokenization. We experimented with other concatenation schemes at the token level but empirically found that image-space concatenation performed better. The model is fine-tuned to autoregressively generate a structured text string containing a sequence of subtask IDs, each followed by four location tokens representing a bounding box mask.

Fine-tuning While PaliGemma2 provides strong vision-language priors, initial experiments showed its zero-shot performance was insufficient for our specific task, particularly for predicting masks based on spatial relations expressed with text and sketches. We therefore fine-tuned the model using Low-Rank Adaptation (LoRA) to adapt the pre-trained model to our task domain. Further details on the dataset used for this process are provided in Section 4.1



(a) Planning Stage (b) Manipulation Stage

Figure 3: An overview of our two-stage method. **(a)** The planning stage generates a plan from a multimodal instruction. **(b)** The manipulation stage executes the plan using direct observation augmentation and a mapping of the task ID to a predefined prompt.

3.3 Manipulation Policy

The manipulation policy translates the symbolic plan from the high-level planner into robot actions. Our modular framework supports two approaches: using specialized, hand-crafted expert policies selected by the subtask ID, or a single, general-purpose learned policy conditioned on the subtask representation. While using task-expert policies offers significant potential for improving reliability and efficiency in structured industrial settings, we focus on a unified, learned policy in this work to evaluate the system's overall generality.

Model We adopt the 3D Diffuser Actor (3DDA) [12] as our manipulation policy, a state-of-the-art model that predicts 6-DoF end-effector keyposes. The model operates on a 3D feature point cloud, which is constructed by projecting the CLIP embeddings of multi-view RGB-D images into a shared world space. Operating directly in 3D provides the policy with explicit geometric information, reducing the burden of inferring spatial relationships from 2D images. Instead of producing a single deterministic output, 3DDA learns a distribution over possible actions through a conditional diffusion process. It iteratively denoises an initial random pose, conditioned on the feature point cloud and a language instruction processed by a CLIP language encoder, to generate the final action. This entire process is handled by a transformer architecture that effectively fuses the different inputs to guide the action generation. Our work proposes a novel and effective method for conditioning such powerful 3D policies to follow complex multimodal instructions.

Conditioning A key advantage of our approach is the flexibility of the intermediate representation, which serves as a general-purpose interface for conditioning a wide range of manipulation policies without altering their architecture. The symbolic subtask ID can be used to select a specialized, hand-crafted expert policy for that primitive. Alternatively, for a general-purpose, multi-task manipulation policy, the ID is used to generate a consistent conditioning signal, shielding the policy from the complexity of the original multimodal instruction. As illustrated in Figure 3b, our implementation with the 3DDA policy maps each subtask ID to a pre-defined text prompt. This mapping, however, could be adapted for policies that use other conditioning schemes, such as one-hot vectors.

The spatial mask provides the second, visual component of the conditioning. We incorporate it by re-coloring the corresponding pixels in the RGB-D images before they are projected into the 3D feature point cloud. This method of direct observation augmentation allows us to inject spatial information from the planning stage into any visuomotor policy that operates on RGB, RGB-D or colored point cloud data.

Execution The planning and manipulation policies run asynchronously, decoupling the computationally expensive VLM planner from the robot's high-frequency control loop. To handle scene changes between updates of the planner, the 2D masks predicted by the planner are projected into a 3D volume. For subsequent manipulation policy steps between planner updates, the points that fall within this latest 3D volume are still highlighted, ensuring the spatial cue remains valid even as

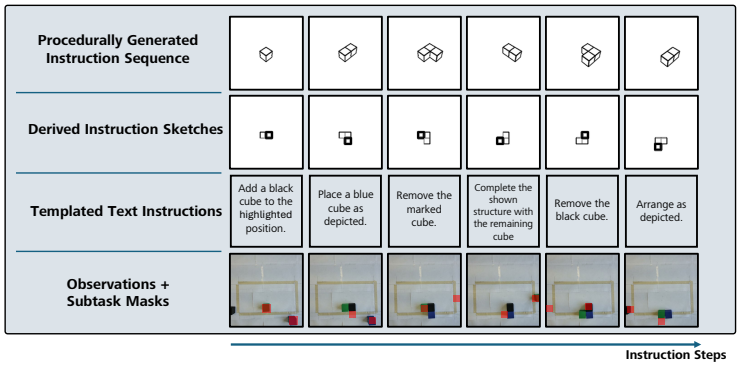


Figure 4: An example of a generated instruction sequence and its corresponding training data. We derive multimodal instructions (sketches and text) from a high-level plan (top row). These are then paired with real-world observations and ground-truth subtask masks (bottom row) to form a complete data tuple for training and evaluation.

the robot and objects move. This design allows the system to benefit from the powerful reasoning of large foundation models for planning, without compromising the high control frequency required for smooth and reactive manipulation.

4 Experiments

In our experiments, we aim to answer the following research questions: **(I)** How does the quality of the VLM planner's mask predictions impact overall system performance compared to oracle (human) masks? **(II)** How crucial is the visual mask for conditioning the manipulation policy compared to using only a detailed textual description? **(III)** Does our hierarchical framework outperform a flat, end-to-end policy that is directly conditioned on the raw multimodal instruction?

Therefore, we conduct a series of real-world experiments to evaluate the performance of our proposed method and analyze the contributions of its key components through systematic ablations.

4.1 Experimental Setup and Benchmark

Platform Our setup consists of a Franka Emika Panda 7-DoF robot arm. The vision system uses two Orbbec Femto Bolt RGB-D cameras–one static and one mounted on the robot's gripper. Both cameras are extrinsically calibrated to the robot's base frame.

Task and Benchmark To address the lack of standardized benchmarks for multimodal instructions, we developed a custom evaluation protocol centered on a block rearrangement task. The benchmark is designed to test the joint understanding of complementary instruction modalities, where neither the text nor the sketch is sufficient on its own. Typically, the textual description specifies abstract information like the operation to perform or the color of the relevant cube (see example instructions in Figure 4). The accompanying schematic sketch provides the crucial spatial context, such as the target location for a placement. This setup requires the system to fuse information from both sources to correctly interpret the goal and ground it in the physical scene.

Data Collection We collected two distinct datasets for our hierarchical system. For the planning policy, we created a dataset of 800 instruction-observation tuples. The instructions, each comprising a text and sketch pair, were created using a procedural generation pipeline. Figure 4 provides an example of such a generated instruction sequence and its corresponding real-world, mask-augmented observations. We then gathered the training data by manually arranging cubes in the real world to match these instructions and capturing the corresponding scene observations. Since this process does not require robot trajectories, it was highly efficient. Ground-truth masks for each step were

Table 1: System performance across different configurations. We report the percentage of rollouts that successfully complete a given number of consecutive steps, as well as the final average number of successful steps (out of 5), averaged over 5 rollouts.

	Success Rate at Step N					Average
Method	**1**	**2**	**3**	**4**	**5**	**Steps**
Ours (VLM + Mask cond.)	100%	100%	100%	60%	20%	**3.8**
Oracle Planner	100%	100%	100%	60%	40%	4.0
Text-Only Baseline	60%	20%	0%	0%	0%	0.6
End-to-End Baseline	60%	40%	20%	0%	0%	1.2

generated via a semi-automated process and manually verified. For the manipulation policy, we collected a smaller dataset of 200 subtask demonstrations via joystick teleoperation, manually marking keyframes to ensure high-quality action labels.

Metric Inspired by the popular CALVIN benchmark [45], we measure performance by the number of consecutively successful steps completed within an episode of five instruction steps. A step is considered successful if the resulting cube placement is within a strict geometric tolerance of the target pose. For each setting, we evaluate performance over 5 rollout episodes.

Baselines We compare our proposed method against several key baselines to analyze the contribution of each component:

- **Oracle Planner:** To evaluate the manipulation policy's performance independent of planning errors, this variant uses human-annotated "oracle" masks and plans as direct input to the manipulation policy, rather than using the output of the VLM. This serves as a practical upper bound for the performance of the planning stage.
- **Text-Only:** To isolate the effect of the visual mask, this baseline removes it from the intermediate representation. The manipulation policy is instead conditioned on a detailed, procedurally generated textual description of the subtask (e.g., "place the red cube to the left of the blue cube").
- **End-to-End:** This baseline removes the hierarchical structure entirely. The manipulation policy is modified to directly take the raw multimodal instruction (text and sketch) as input, without any intermediate planning or mask generation.

4.2 Results and Ablation Studies

Our main results are summarized in Table 1. Our full hierarchical system successfully executes an average of **3.8 out of 5 steps**. The system is robust in early steps, but performance can degrade in later steps due to the accumulation of small placement errors (e.g., minor gaps or rotational offsets) that make subsequent placements more challenging. The following ablation studies discuss these results in detail. Figure 5 shows a qualitative example of a successful multi-step rollout. A more detailed visualization of a rollout and common failure modes can be found in the appendix.

Is the planner the bottleneck? (VLM vs. Human Masks) To isolate the impact of planning errors, we compare our system against an Oracle Planner that uses human-annotated masks. The oracle serves as a practical upper bound for the planning stage. When using these perfect masks, the system's performance only increases slightly from 3.8 to 4.0 average successful steps (see Table 1). This small gap indicates that our VLM planner is highly effective and that the manipulation policy is the primary performance bottleneck. This conclusion is further supported by the planner's strong standalone performance, achieving a mean Intersection-over-Union (IoU) of 0.79 on the test set.

How important is visual hinting? (Masks vs. Text-Only) To evaluate the importance of our visual conditioning method, we compare our system to a Text-Only baseline. For this baseline, we

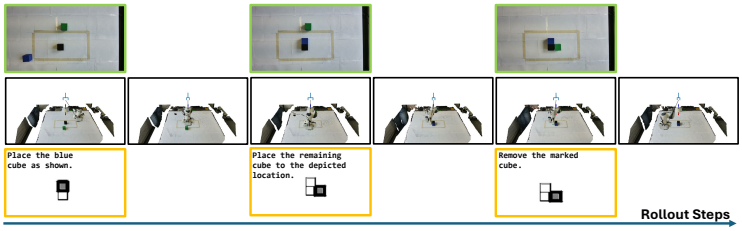


Figure 5: A qualitative example of a successful multi-step rollout from our real-world experiments. The robot correctly interprets a sequence of three complementary multimodal instructions (bottom row) to successfully rearrange the colored cubes into the target configurations (top row).

use our procedural generation metadata to create precise, template-based textual descriptions that contain all the necessary spatial and semantic information. For example, the policy might receive only a detailed command like ”place the blue cube to the right of the green cube so that the cubes form an L-shape”. Despite providing these complete textual instructions, the performance dropped dramatically from 3.8 to 0.6 average successful steps. The policy struggled to ground the complex spatial relations from text alone, highlighting that our method of explicit visual hinting via mask augmentation is crucial for reliable execution.

Is hierarchy necessary? (Hierarchical vs. End-to-End) To justify our hierarchical design, we compare it to a flat, end-to-end baseline. For this baseline, we modified the 3DDA policy to directly accept the raw multimodal instruction by adding a second image encoder for the instruction sketch and concatenating its features with the scene observation tokens. This end-to-end model performed poorly, achieving only 1.2 average successful steps. The poor performance confirms that simply providing the raw instruction is insufficient, as the policy struggles to learn the complex mapping from abstract, multimodal goals to precise actions from a limited dataset. In contrast, our hierarchical approach makes the problem significantly more tractable by first translating the complex instruction into an explicit, spatially-grounded intermediate representation.

5 Conclusion

We introduced a hierarchical robotic policy framework capable of interpreting and executing multimodal assembly instructions that combine language and visual sketches. The key to our approach is the decoupling of high-level reasoning from low-level control, which is achieved through a structured intermediate representation. This representation consists of a sequence of symbolic subtasks, where each subtask is parameterized by a spatial mask that a VLM planner predicts and projects directly onto the manipulation policy’s visual input. Our experiments demonstrate that this method of grounding symbolic plans with explicit visual hints is a highly effective strategy. The ablation studies underscore that this form of direct observation augmentation is critical, significantly outperforming baselines that rely on text-only instructions or end-to-end conditioning on raw multimodal inputs.

6 Limitations

While our results are promising, this work has limitations that suggest clear directions for future research. The current framework was evaluated in a controlled tabletop setting and should be validated on more diverse, real-world assembly tasks. Furthermore, fine-tuning on datasets of human-authored industrial manuals, rather than our procedurally generated instructions, would be crucial for improving generalization to real-world ambiguity. Finally, incorporating mechanisms for error detection and recovery would significantly enhance the system’s autonomy and robustness in dynamic environments.

7 Acknowledgments

The work was funded by the German Research Foundation (DFG) – 448648559. The authors also acknowledge support by the state of Baden-Württemberg through HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the German Federal Ministry of Education and Research. The work was also supported by the Carl Zeiss AG.

References

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

[2] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[3] S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, S. Bohez, K. Bousmalis, A. Brohan, T. Buschmann, A. Byravan, S. Cabi, K. Caluwaerts, F. Casarini, O. Chang, J. E. Chen, X. Chen, H.-T. L. Chiang, K. Choromanski, D. D'Ambrosio, S. Dasari, T. Davchev, C. Devin, N. D. Palo, T. Ding, A. Dostmohamed, D. Driess, Y. Du, D. Dwibedi, M. Elabd, C. Fantacci, C. Fong, E. Frey, C. Fu, M. Giustina, K. Gopalakrishnan, L. Graesser, L. Hasenclever, N. Heess, B. Hernaez, A. Herzog, R. A. Hofer, J. Humplik, A. Iscen, M. G. Jacob, D. Jain, R. Julian, D. Kalashnikov, M. E. Karagozler, S. Karp, C. Kew, J. Kirkland, S. Kirmani, Y. Kuang, T. Lampe, A. Laurens, I. Leal, A. X. Lee, T.-W. E. Lee, J. Liang, Y. Lin, S. Maddineni, A. Majumdar, A. H. Michaely, R. Moreno, M. Neunert, F. Nori, C. Parada, E. Parisotto, P. Pastor, A. Pooley, K. Rao, K. Reymann, D. Sadigh, S. Saliceti, P. Sanketi, P. Sermanet, D. Shah, M. Sharma, K. Shea, C. Shu, V. Sindhwani, S. Singh, R. Soricut, J. T. Springenberg, R. Sterneck, R. Surdulescu, J. Tan, J. Tompson, V. Vanhoucke, J. Varley, G. Vesom, G. Vezzani, O. Vinyals, A. Wahid, S. Welker, P. Wohlhart, F. Xia, T. Xiao, A. Xie, J. Xie, P. Xu, S. Xu, Y. Xu, Z. Xu, Y. Yang, R. Yao, S. Yaroshenko, W. Yu, W. Yuan, J. Zhang, T. Zhang, A. Zhou, and Y. Zhou. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[4] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, et al. Openvla: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024.

[5] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[6] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan. 3d-vla: A 3d vision-language-action generative world model. In *International Conference on Machine Learning*, pages 61229–61245. PMLR, 2024.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[8] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[9] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. Vima: Robot manipulation with multimodal prompts. In *International Conference on Machine Learning*, pages 14975–15022. PMLR, 2023.

[10] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang. An embodied generalist agent in 3d world. In *International Conference on Machine Learning*, pages 20413–20451. PMLR, 2024.

[11] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[12] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *8th Annual Conference on Robot Learning*, 2024.

[13] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn. Yell at your robot: Improving on-the-fly from language corrections. *arXiv preprint arXiv:2403.12910*, 2024.

[14] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. In *Conference on Robot Learning*, pages 3949–3965. PMLR, 2023.

[15] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. In *Proceedings of The 6th Conference on Robot Learning*, pages 785–799. PMLR, Mar. 2023.

[16] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. RVT: Robotic View Transformer for 3D Object Manipulation. In *Proceedings of The 7th Conference on Robot Learning*, pages 694–710. PMLR, Dec. 2023.

[17] P. Sundaresan, Q. Vuong, J. Gu, P. Xu, T. Xiao, S. Kirmani, T. Yu, M. Stark, A. Jain, K. Hausman, et al. Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches. In *8th Annual Conference on Robot Learning*, 2024.

[18] I. Kapelyukh, V. Vosylius, and E. Johns. DALL-E-Bot: Introducing Web-Scale Diffusion Models to Robotics. *IEEE Robotics and Automation Letters*, 8(7):3956–3963, July 2023. ISSN 2377-3766, 2377-3774. doi:10.1109/LRA.2023.3272516.

[19] M. Reuss, Ö. E. Yağmurlu, F. Wenzel, and R. Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. *Robotics: Science and Systems*, 2024.

[20] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977*, 2023.

[21] R. Shah, R. Martín-Martín, and Y. Zhu. Mutex: Learning unified policies from multimodal task specifications. In *Conference on Robot Learning*, pages 2663–2682. PMLR, 2023.

[22] P. Sharma, A. Torralba, and J. Andreas. Skill induction and planning with latent language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1713–1726, 2022.

[23] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox. Robopoint: A vision-language model for spatial affordance prediction in robotics. In *8th Annual Conference on Robot Learning*, 2024.

[24] Y. Shentu, P. Wu, A. Rajeswaran, and P. Abbeel. From llms to actions: latent codes as bridges in hierarchical robot control. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8539–8546. IEEE, 2024.

[25] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[26] S. H. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor. ChatGPT for Robotics: Design Principles and Model Abilities. *IEEE Access*, 12:55682–55696, 2024. ISSN 2169-3536. doi: 10.1109/ACCESS.2024.3387941.

[27] F. Liu, K. Fang, P. Abbeel, and S. Levine. MOKA: Open-Vocabulary Robotic Manipulation through Mark-Based Visual Prompting. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, Apr. 2024.

[28] R. Garcia, S. Chen, and C. Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. *arXiv preprint arXiv:2410.01345*, 2024.

[29] P. Sundaresan, S. Belkhale, D. Sadigh, and J. Bohg. Kite: Keypoint-conditioned policies for semantic manipulation. In *Conference on Robot Learning*, pages 1006–1021. PMLR, 2023.

[30] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg. Text2Motion: From Natural Language Instructions to Feasible Plans. *Autonomous Robots*, 47(8):1345–1365, Dec. 2023. ISSN 0929-5593, 1573-7527. doi:10.1007/s10514-023-10131-7.

[31] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.

[32] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. In *International Conference on Machine Learning*, pages 37321–37341. PMLR, 2024.

[33] Y. Li, Y. Deng, J. Zhang, J. Jang, M. Memmel, C. R. Garrett, F. Ramos, D. Fox, A. Li, A. Gupta, and A. Goyal. HAMSTER: Hierarchical Action Models for Open-World Robot Manipulation. In *The Thirteenth International Conference on Learning Representations*, Oct. 2024.

[34] D. Niu, Y. Sharma, G. Biamby, J. Quenum, Y. Bai, B. Shi, T. Darrell, and R. Herzig. Llarva: Vision-action instruction tuning enhances robot learning. In *8th Annual Conference on Robot Learning*, 2024.

[35] M. Reuss, M. Li, X. Jia, and R. Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. In *Robotics: Science and Systems*, 2023.

[36] O. Mees, L. Hermann, and W. Burgard. What Matters in Language Conditioned Robotic Imitation Learning over Unstructured Data. https://arxiv.org/abs/2204.06252v2, Apr. 2022.

[37] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox. Rvt-2: Learning precise manipulation from few demonstrations. In *RSS 2024 Workshop: Data Generation for Robotics*, 2024.

[38] Y. Wang, M. Zhang, Z. Li, K. R. Driggs-Campbell, J. Wu, L. Fei-Fei, and Y. Li. D3 fields: Dynamic 3d descriptor fields for zero-shot generalizable robotic manipulation. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

[39] Y. Xu, L. Li, C. Yu, and D. Hsu. " stack it up!": 3d stable structure generation from 2d hand-drawn sketch. *arXiv preprint arXiv:2508.02093*, 2025.

[40] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[41] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan. 3D-LLM: Injecting the 3D World into Large Language Models. *Advances in Neural Information Processing Systems*, 36: 20482–20494, Dec. 2023.

[42] A. Steiner, A. S. Pinto, M. Tschannen, D. Keysers, X. Wang, Y. Bitton, A. Gritsenko, M. Minderer, A. Sherbondy, S. Long, et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

[43] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

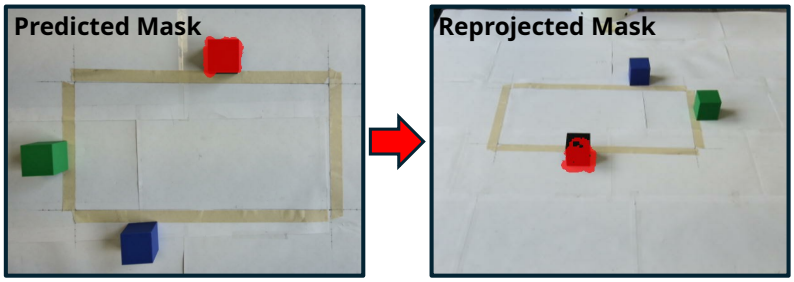
[44] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[45] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

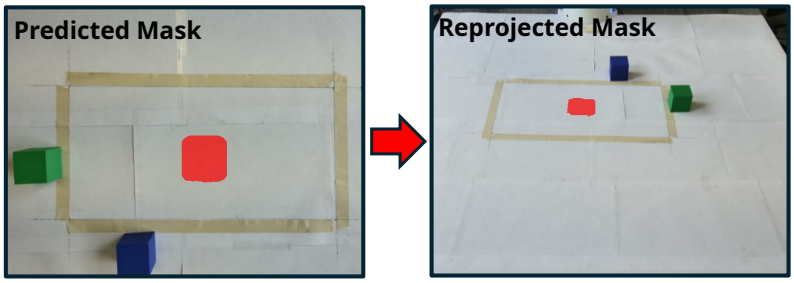
A Appendix

A.1 Mask Projection

Mask reprojection allows the spatial guidance generated by the V-LM planner for a single camera view to be applied across all views used by the manipulation policy. The process first associates each masked pixel in the source view with a 3D spherical volume using the available depth information from the calibrated cameras. Subsequently, any pixel in a target view whose corresponding 3D point falls within one of these volumes is also masked. This technique enhances efficiency by limiting the number of images the planner must process, without restricting the manipulation policy from leveraging multi-view, mask-augmented observations for control.



(a) A predicted mask (left) for a target object is reprojected (right).



(b) A predicted mask (left) for a target location is reprojected (right).

Figure 6: Visualization of our mask reprojection method for two common cases. In both examples, a mask predicted in one camera view is successfully reprojected onto a different view, allowing for multi-view guidance of the manipulation policy.

A.2 Extended Rollout Example

We conduct a series of rollouts per policy variation to evaluate their performance. Figure 7 provides a detailed visualization of a complete and successful five-step rollout, illustrating the interplay between the multimodal instructions, the planner's mask predictions, and the robot's physical execution.

A.3 Examples of Failure Modes

Due to the high precision required by the block rearrangement benchmark, small errors in the predicted pose can compound and lead to task failure. While the specific causes vary, we observed several recurring failure modes, which are depicted in Figure 8.

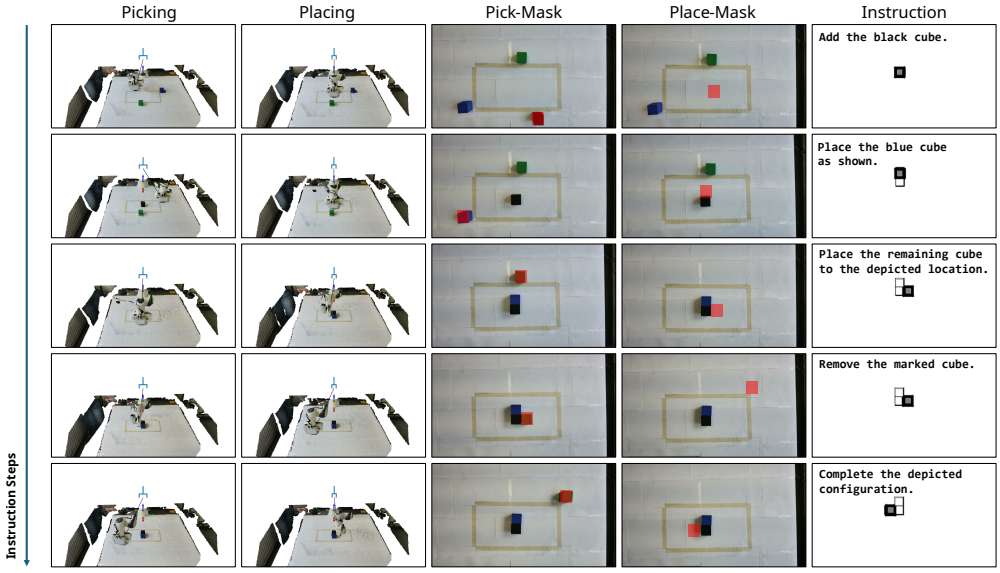
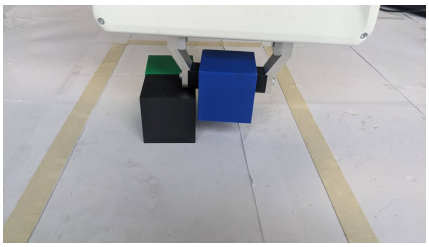
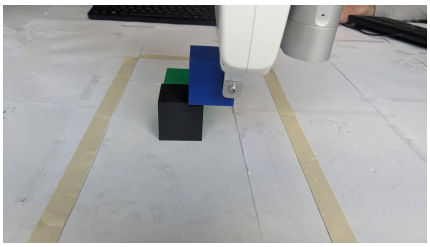


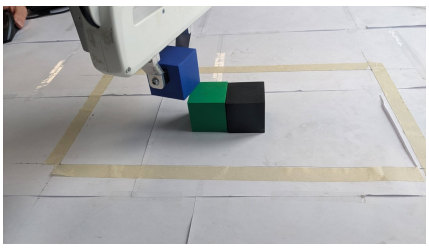
Figure 7: Visualization of a successful five-step rollout sequence. The right-most column depicts the multimodal instruction given to the planning stage at each step. The third and fourth columns show the corresponding mask predictions from the VLM planner. The first two columns show the successful physical execution of the rearrangement task by the robot.



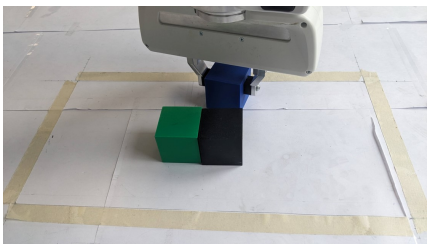
(a) Incorrect gripper orientation causes a collision with an adjacent object.



(b) An off-centered grasp of the objects leads to a collision during placement.



(c) An incorrect object orientation causes a collision during placement.



(d) The policy fails to place the object at the precisely instructed target location.

Figure 8: Visualization of different failure modes of the policy.