

Hierarchical Vision-Language Planning for Multi-Step Humanoid Manipulation

André Schakkal^{1,2} Ben Zandonati¹ Zhutian Yang¹ Navid Azizan¹

¹Massachusetts Institute of Technology

²École Polytechnique Fédérale de Lausanne



Fig. 1: Our hierarchical humanoid manipulation system autonomously executes a multi-step rearrangement task. The robot first pushes aside an obstacle, then picks up a bag and places it onto another surface.

Abstract—Enabling humanoid robots to reliably execute complex multi-step manipulation tasks is crucial for their effective deployment in industrial and household environments. This paper presents a hierarchical planning and control framework designed to achieve reliable multi-step humanoid manipulation. The proposed system comprises three layers: (1) a low-level reinforcement-learning-based controller responsible for tracking whole-body motion targets, (2) a mid-level set of skill policies trained via imitation learning that produce motion targets for different steps of a task, and (3) a high-level vision-language planning module that determines which skills should be executed and also monitors their completion in real time using pre-trained vision-language models (VLMs). Experimental validation is performed on a Unitree G1 humanoid robot executing a non-prehensile pick-and-place task. Over 40 real-world trials, the hierarchical system achieved a 73% success rate in completing the full manipulation sequence. These experiments confirm the feasibility of the proposed hierarchical system, highlighting the benefits of VLM-based skill planning and monitoring for multi-step manipulation scenarios. See <https://vlp-humanoid.github.io/> for a video demonstration of the policy rollout.

I. INTRODUCTION

Humanoid robots offer a promising path towards general-purpose autonomy, integrated seamlessly into human environments. The humanoid morphology, while advantageous for tool use and ergonomic compatibility, presents challenging control and planning problems [11]. Recent reinforcement-learning (RL) and imitation-learning (IL) breakthroughs have produced striking single-skill demonstrations such as walking, dancing, or boxing [6, 10, 13, 12, 14]. Yet, practical applications require that the humanoid carry out multi-step tasks, i.e., interacting with multiple objects and regions over time. The sequencing and integration of these learned skills

remains challenging, due to the high-dimensional nature of humanoid control, the requirement for continuous real-time visual feedback, and the complexity of determining precise transitions between consecutive skills.

To bridge the gap between isolated skill execution and long-horizon autonomy, robots must combine low-level motor dexterity with higher-level reasoning and task monitoring. Vision-language models (VLMs) present a unique opportunity to integrate planning and monitoring with robust learned control policies, enabling embodied humanoid agents to execute extended, multi-step manipulation tasks in unstructured, human environments [8, 29, 28].

In this paper, we present a hierarchical framework for humanoid manipulation planning and control, spanning three layers. The lowest level employs a whole-body tracking controller trained via proximal policy optimization (PPO) [25], which tracks motion targets provided by a mid-level imitation learning (IL) policy. This mid-level policy is trained on human teleoperation data, where human poses are retargeted onto the robot’s morphology. Each IL-based skill policy maps egocentric visual observations and proprioceptive states to full-body motion commands, enabling the execution of complex loco-manipulation tasks. At the highest level, we propose a hierarchical planning module consisting of two VLMs: a low-frequency VLM planner is used to generate sequences of IL skills from visual and textual task inputs, and a higher-frequency VLM-based skill-execution monitor that continuously verifies skill completion, orchestrating transitions between skills.

We empirically validate our proposed hierarchical system on the 29-DoF Unitree G1 humanoid robot, performing a representative multi-step pick-and-place task. In real-world

testing, the integrated system successfully completed the full manipulation sequence in 73% of 40 trials. Our hierarchical approach demonstrates effective coordination of manipulation skills and visual feedback, highlighting the benefit of structured skill decomposition and VLM-based execution monitoring for reliable, multi-step humanoid autonomy.

The primary contributions of our work are: (1) a novel hierarchical vision-language planning and monitoring framework that dynamically sequences and verifies humanoid manipulation skills; and (2) an integrated autonomous system validated on a real humanoid robot performing complex multi-step tasks in realistic environments, demonstrating practical feasibility and effectiveness.

II. RELATED WORK

A. Hierarchical Control in Humanoids

Recent approaches to humanoid control commonly split the control task into two separate layers: a low-level reinforcement learning (RL) controller that tracks reference motion targets, and a higher-level module that generates these motion targets. ExBody [6] first showed that a PPO tracker trained in Isaac Gym [21] can reproduce upper-body MoCap clips on the Unitree H1 while leaving the legs free for balance. ExBody2 [16] extends the idea to whole-body loco-manipulation via teacher-student distillation and domain randomization, achieving high-fidelity whole-body tracking on both H1 and G1 platforms. HumanPlus [10] replaces the multilayer perceptron (MLP) backbone of the low-level policy with a transformer backbone. Additionally, it expands the training data to use the AMASS dataset [20], significantly increasing the diversity of available human motion examples.

While low-level policies reliably track motion targets, generating these references is equally critical for accomplishing manipulation tasks. H2O (Human-to-Humanoid) [15] utilized single-camera human pose estimation to enable zero-shot teleoperation. OmniH2O [12] extended H2O’s concept, offering a unified interface for teleoperation using various modalities, including RGB cameras, VR, motion capture suits, and exoskeleton systems. OmniH2O further incorporated diffusion-based imitation learning [7], enabling the autonomous acquisition of manipulation skills directly from human demonstrations. Similarly, HumanPlus [10] adopted this hierarchical structure, leveraging human-teleoperated demonstrations recorded along with proprioceptive and egocentric visual data. These demonstrations were used to train the Humanoid Imitation Transformer (HIT)—a transformer-based imitation learning policy inspired by the Action Chunking Transformer (ACT) [31]. Using only 20–40 demonstrations per skill, HumanPlus reported 60–100% success rates across diverse whole-body manipulation tasks.

Across these studies, a common hierarchical structure appears, featuring a robust RL-based low-level policy for tracking motion references and a high-level module generating these references through teleoperation or imitation learning. Although this approach has shown success in single-skill tasks (e.g., dancing, lifting, and boxing), existing systems generally

require human intervention when transitioning between skills, thus limiting their ability to operate autonomously over extended task sequences.

B. VLMs for Long-Horizon Manipulation

Executing multi-step tasks demands semantic understanding, robust memory, and resilience to errors. Recent approaches leveraging large vision-language-action (VLA) models, such as RT-2 [4], OpenVLA [17], and $\pi_{0.5}$ [3], demonstrate promise in end-to-end planning but typically require large-scale paired datasets and lack interpretability. To address these limitations, hierarchical approaches integrate intermediate semantic representations, bridging abstract instructions and concrete actions (e.g., RT-H [2], NaVILA [5]). Another direction employs explicit hierarchies, where high-level vision-language models predict coarse geometric plans refined by lower-level controllers, effectively separating semantic reasoning from robot dynamics [19, 27]. Additionally, methods like KALM [9], RoboPoint [30], and ManipGen [8] use VLMs to predict spatial affordances or parameterized skill libraries, facilitating efficient zero-shot generalization.

Despite the advances in humanoid control, current humanoid systems lack integrated high-level modules capable of autonomously selecting and sequencing skills, as well as verifying successful execution. Addressing this gap, our work proposes an additional third hierarchical layer, combining a VLM-based planner and skill monitor on top of the established two-layer humanoid control stacks, enabling fully autonomous execution of extended loco-manipulation tasks.

III. METHOD

We propose a three-layer hierarchical control and planning framework for multi-step humanoid manipulation, illustrated in Fig. 2. The system consists of: (1) a low-level whole-body tracking controller trained via reinforcement learning (Sec. III-A), (2) a mid-level motion-generation module composed of multiple imitation-learned action expert policies, each trained from teleoperated demonstrations (Sec. III-B), and (3) a high-level vision-language-based planning and execution monitoring module (Sec. III-C). The following sections detail each component.

A. Low-Level Tracking Policy

Our low-level module is a reinforcement learning-based tracking controller designed to track generated motion goals. Building upon the ExBody architecture [6], we extended the implementation to the full 29-DoF Unitree G1 humanoid robot, incorporating whole-body tracking.

1) Problem Formulation: The tracking policy, denoted as

$$\pi_T : G \times S \rightarrow A,$$

maps a motion goal G and the robot’s current state S to feasible joint commands A . The goal space G consists of two components: a Root Motion Goal G_m defined by a desired base linear velocity v_{ref} and orientation ω_{ref} , and an Expression Goal G_e defined by target joint angles q_{ref} and

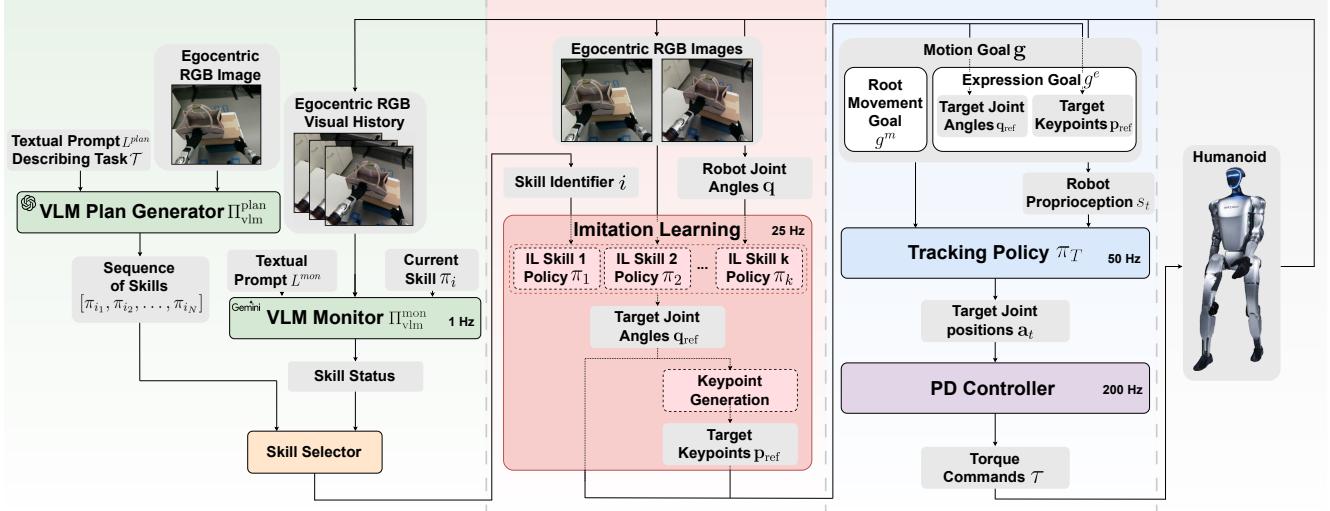


Fig. 2: Overview of the proposed hierarchical framework for autonomous multi-step humanoid manipulation. The system consists of: (1) a high-level planning module using vision-language models (VLMs) to generate task-specific skill sequences and monitor their execution via visual feedback, (2) a mid-level motion-generation module composed of imitation-learned action expert policies, mapping egocentric visual inputs and proprioceptive robot states into target joint configurations, and (3) a low-level reinforcement-learning-based tracking policy generating feasible joint-angle trajectories, executed by a high-frequency PD controller.

a set of 3D keypoint positions p_{ref} corresponding to specific body landmarks, the expression goal specifies the desired motion or gesture that the robot should track. The state S includes proprioceptive observations such as joint positions and velocities, IMU data, and a short history of previous states. The policy outputs target joint angles at 50 Hz, enforced using a high-frequency PD controller operating at 200 Hz.

2) Policy Architecture and Training: We employ a standard actor-critic architecture trained via proximal policy optimization (PPO) [25]. Both the actor and critic networks are implemented as three-layer MLPs (256 hidden units, ReLU activations). Training occurs within the Isaac Gym simulator using 4096 parallel environments. To ensure robustness in real-world conditions, we extensively randomize environment parameters, including gravity, friction, mass distribution, motor strength, external perturbations, and terrain irregularities.

The reward function combines tracking accuracy terms, adapted from ExBody [6] and ExBody2 [16], along with additional stability-promoting regularization terms specifically tuned for the G1 robot. Detailed reward components are provided in Appendix A.

3) Training Data: Training data was sourced from a curated subset of the AMASS dataset [20], selected to cover the robot's operational range. The data was supplemented by additional teleoperation demonstrations that we collected, tailored to the specific downstream skills and tasks outlined in the project. This combination ensures both diversity and task-relevance of the training data. Quantitative and qualitative evaluations of the tracking performance of the low-level policy outlined in Appendix A.

B. Mid-Level Imitation Learned Skills

Building upon the low-level tracking policy's capability to execute motion goals precisely, the mid-level module generates higher-level skill behaviors autonomously. Specifically, it translates sensory inputs into coherent joint-motion targets that serve as reference trajectories for the tracking policy. This module integrates two complementary capabilities: an RGB-based teleoperation pipeline for demonstration collection, and imitation-learned (IL) expert policies trained from these demonstrations to execute individual skills autonomously.

1) Teleoperation via Human Pose Retargeting: To facilitate the collection of demonstration data, we developed an RGB-based teleoperation pipeline allowing human operators to control the humanoid robot directly. This system consists of three primary stages: human pose estimation from RGB images, retargeting of human poses onto the robot, and generation of robot-specific keypoints.

For human pose estimation, we employ HybrIK (Hybrid Inverse Kinematics) [18], a hybrid analytical-neural inverse kinematics model. HybrIK initially predicts 3D joint positions from single-view RGB inputs using a convolutional neural network (CNN). Subsequently, these predictions are refined through a physics-informed inverse kinematics optimization step, yielding temporally smooth and anatomically plausible human joint configurations. This pipeline eliminates the need for specialized motion-capture equipment, requiring only a standard RGB camera setup.

Due to morphological differences between human operators and the Unitree G1 robot—including limb proportions, joint limits, and degrees of freedom—a dedicated retargeting proce-

dure translates the human pose estimates into robot-compatible joint configurations. This retargeting procedure is outlined in Appendix A.

As a final step, the low-level tracking policy expects as input both joint-angle references and corresponding 3D keypoints representing the center-of-mass (CoM) positions of selected robot body links. To generate these keypoints, we apply forward kinematics in simulation: we first set the robot to the retargeted joint configuration, and then compute the spatial coordinates of the predefined body positions. The resulting joint angles and keypoints form the complete motion reference that can be directly executed by the tracking controller.

2) *Imitation Learning Module*: Building upon the established teleoperation pipeline, we next develop the imitation learning (IL) module, the mid-level component responsible for generating autonomous motion targets from onboard sensory inputs.

While the low-level tracking policy executes externally provided motion references, the IL module produces these references directly from sensory inputs, enabling robot autonomy. Formally, the IL policy is defined as a function

$$\pi_{IL} : \mathcal{X} \rightarrow \mathcal{A},$$

where the observation state at each timestep t is

$$x_t = (\mathbf{q}_t, I_t^{(1)}, I_t^{(2)}),$$

comprising proprioceptive joint angles \mathbf{q}_t and binocular egocentric RGB images $I_t^{(1)}$ and $I_t^{(2)}$. The action space \mathcal{A} consists of future joint configurations \mathbf{q} , directly interfacing with the low-level tracking policy.

For the IL policies, we adopt the Humanoid Imitation Transformer (HIT) [10], designed specifically for high-DoF humanoids with binocular vision. It is based on the Action Chunking Transformer (ACT) [31] model, which has demonstrated improved robotic control by predicting multi-step future actions in one forward pass. At inference, HIT outputs action chunks of 50 joint-angle targets at 25 Hz. During training, an auxiliary L_2 loss between predicted and actual future visual embeddings improves visual grounding and generalization. The predicted joint angles are subsequently transformed into corresponding 3D keypoints via forward kinematics, matching the inputs required by the low-level tracking controller.

Instead of using the robot’s onboard Intel RealSense camera, we employ two externally mounted ELP high-speed RGB cameras with wide-angle lenses. This binocular configuration provides consistent frame rates and explicit depth information, enhancing spatial reasoning capabilities essential for manipulation tasks.

Demonstration data for IL training is collected through teleoperation, forming a dataset

$$\mathcal{D} = \{(x_1, a_1), (x_2, a_2), \dots, (x_N, a_N)\},$$

containing synchronized state-action pairs recorded at 25 Hz. We collect data for the representative task rearrangement

involving picking a bag from a table and placing it onto another. Only successful sequences (30 in total) are retained, segmented into separate skill-specific datasets, i.e., picking and placing, for training two policies. IL policy training employs the demonstration data with hyperparameters specified in Table V. After training, we evaluated independently each skill policy over 30 autonomous trials, which showed success rates of 90% for picking and 83% for placing.

C. Multi-Step Skill Planning and Execution Monitoring

While imitation-learned skills enable reliable execution of short-term actions, real-world humanoid tasks frequently require combining multiple skills into longer sequences to achieve complex goals. We address this by introducing a high-level planning and execution monitoring module, capable of dynamically selecting and verifying the execution of skill sequences. Our approach leverages pretrained vision-language models (VLMs) embedded within a closed-loop *planner-monitor* architecture.

To clearly distinguish conceptual terminology, we define:

- A **skill** as an individual, short-duration manipulation capability (e.g., picking or placing an object).
- A **task** as a higher-level objective involving at least two skills executed sequentially that change the state of objects in the world.

Our hierarchical system thus plans at the task-level while monitoring and executing at the skill-level.

1) *Problem Formulation*: Formally, a task is defined by a natural-language goal g (e.g., “Pick up the bag and place it on the table”) and an initial visual observation o_0 . The goal is to autonomously generate and reliably execute a sequence of parameterized skills $\sigma = [\pi^{(1)}, \pi^{(2)}, \dots, \pi^{(N)}]$ to satisfy the conditions described by g .

2) *System Architecture*: Our high-level planning and monitoring module consists of two complementary components:

- (a) **VLM Planner (\mathcal{P})**: A GPT-4o model [22] generates structured, interpretable skill sequences from visual and textual task inputs.
- (b) **VLM Skill Monitor (\mathcal{M})**: A lightweight Gemini-2.0-Flash-Lite model [28] continuously verifies the completion of each executed skill at approximately 1 Hz.

Together, these two components form an iterative planning-monitoring loop, ensuring coherent multi-step task execution.

3) *Skill Library and PDDL-like Representation*: We represent each skill available to the robot using structured, human-readable descriptions that include the ‘preconditions’ and ‘effects’ fields of an operator definition in Planning Domain Definition Language (PDDL) [1]. Each skill description includes a one-sentence summary, explicit natural-language preconditions, effects, and example verification queries, as illustrated for the `Pick` skill in Fig. 3. Unlike rigid symbolic representations, this PDDL-like form balances logical rigor and flexibility, enabling the planner to perform common-sense reasoning without exhaustive state enumeration.

```

Pick:
Description: 'Pick up an object and hold it up'
Preconditions: [
    'hand is empty',
    'object is graspable',
    'object is on a surface'
]
Effects: [
    'robot is holding object up',
    'object is no longer on the surface'
]
Examples: 'Has the robot successfully picked up and
is now holding the bag?'

```

Fig. 3: Example structured skill description provided to the VLM Planner and Skill Monitor.

4) **VLM Planner:** Given an initial image o_0 , task instruction g , and the structured skill library, the VLM Planner \mathcal{P} outputs a grounded sequence of executable skills σ . GPT-4o generates the skill sequence by visually grounding skill preconditions and effects, translating them into binary visual-question-answering queries for logical coherence. For instance, the planner implicitly understands causal ordering, such as the necessity of picking up an object before placing it elsewhere, from the provided natural-language annotations, without explicit symbolic planning constraints.

5) **VLM Skill Monitor:** The VLM Skill Monitor \mathcal{M} continuously evaluates the execution state through real-time visual feedback. Operating at approximately 1 Hz, it assesses short video snippets (10–15 frames extracted from 1.5-second segments) captured during skill execution. For each skill, the monitor uses the provided natural-language verification queries to output a binary decision: either *completed* or *in progress*. This approach ensures timely and reliable skill transitions while maintaining real-time responsiveness.

A walkthrough example of the functioning of this module is presented in Appendix C while Appendix D shows example VLM prompts.

D. Full Hierarchical System

The complete integrated system implements the hierarchical pipeline presented in Fig. 2. The pipeline comprises: (1) a **high-level VLM planner** (GPT-4o) that generates a sequence of skills from initial visual inputs and natural-language task instructions, (2) a real-time **VLM skill monitor** (Gemini-2.0-Flash-Lite) that verifies skill execution status based on visual feedback, (3) a **skill selector** that dynamically chooses the current skill to execute based on the planned sequence and monitor feedback, (4) a **mid-level imitation learning Module** that activates the corresponding imitation-learned policy according to the selected skill, producing joint-motion targets from egocentric images and proprioceptive data, and (5) a **low-level tracking policy** combined with a high-frequency **PD controller**, converting these joint targets into executable motor torque commands.

A key advantage of this hierarchical design is its inherent modularity: individual IL skills are trained independently and added directly to the skill library, enabling effortless

expansion of the robot’s behavioral repertoire. Newly acquired skills become immediately available for integration by the high-level planning module, facilitating seamless composition into increasingly complex, multi-step tasks without extensive system reconfiguration.

IV. EXPERIMENTS

We conducted experiments to evaluate the effectiveness and robustness of our proposed hierarchical control framework for multi-step humanoid manipulation tasks on a real-world robot platform.

A. Experimental Setup

Experiments were conducted using the 29-degree-of-freedom (DoF) Unitree G1 humanoid robot. The robot was equipped with two externally mounted ELP RGB cameras, providing binocular egocentric visual inputs for the mid-level imitation learning (IL) policies. The robot was tested within a controlled environment consisting of two tables and objects arranged to simulate representative manipulation tasks common in household scenarios.

B. Evaluation Task

To validate the hierarchical framework, we defined a representative multi-step manipulation task involving:

- 1) **Picking:** The robot must grasp and lift the bag from the initial table.
- 2) **Placing:** The robot needs to accurately place the bag onto a second table.

C. Experimental Procedure

A total of 40 independent trials were conducted. The robot operated fully autonomously, executing plans generated by the high-level VLM planner, with skills dynamically selected and monitored for completion in real-time by the VLM execution monitor. Success was defined by the robot correctly picking up the bag and successfully placing it on the designated target surface.

D. Experimental Results

Over 40 trials, the integrated hierarchical system successfully completed the entire manipulation sequence with an overall success rate of 73%. During each trial, we recorded failures at three distinct levels of the hierarchical framework. The observed failures were categorized as follows, in order of frequency:

- **Skill Policy Failure:** Occurred when a mid-level imitation policy failed to execute the desired skill. Most frequently occurred during the grasping phase, primarily due to variations in object positioning beyond training data distributions.
- **VLM Execution Monitor Failure:** Instances where the visual verification prematurely indicated task completion due to subtle positioning inaccuracies or ambiguous visual cues.
- **VLM Planner Failure:** Occasional mis-grounding issues occurred, resulting in the generation of incorrect skill sequences.

Qualitatively, the robot exhibited stable and coherent manipulation behaviors across successful trials. Skills transitioned smoothly under guidance from the VLM execution monitor, demonstrating effective real-time adaptation and corrective feedback. Minor irregularities in the skills arose due to executing multiple consecutive 50-step action chunks, causing subtle positional resets between chunks. This suggests that introducing a smoothing or blending function between consecutive chunks could further enhance motion fluidity.

V. DISCUSSION

The experiments demonstrate that augmenting the canonical two-layer humanoid stack with a vision-language planning and monitoring layer enables reliable execution of multi-step non-prehensile manipulation routines in the physical world. Deployed on a 29-DoF Unitree G1, our proposed three-level architecture successfully completed the obstructed bag-transfer task in 73% of forty independent trials. The execution monitor’s one-second mean verification latency at 1Hz fits comfortably within the robot’s motion cycle, and coexists with high-frequency skill policies and RL controller, illustrating that the large-model reasoning and perception need not preclude high-rate control.

Compared to prior humanoid manipulation systems, which often rely on open-loop sequences of predefined skills, our approach introduces continuous closed-loop reasoning. By integrating interpretable VLM-based planning with real-time visual verification, our framework dynamically selects appropriate skills and explicitly determines when transitions between skills should occur.

Furthermore, unlike purely end-to-end vision-language-action (VLA) approaches, which depend heavily on large-scale, paired demonstration datasets and typically lack interpretability, our explicit planner-monitor design provides clear introspection into decision-making processes. This transparency facilitates targeted debugging, allowing developers to easily diagnose and address failures. Similarly, compared to strictly symbolic Task and Motion Planning (TAMP) frameworks, our system’s reliance on soft, natural-language-based skill descriptions—akin to a flexible version of PDDL—greatly simplifies the integration and extension of new behaviors without necessitating detailed domain-specific expertise.

A. Failures and Limitations

A qualitative failure analysis reveals three dominant modes. First, skill-policy drift occurs when partial occlusions or out-of-distribution object poses cause the imitation policy to stall short of a grasp or placement. Second, planner misgrounding emerges when the deliberative VLM hallucinates objects or fails to sequence skills correctly. This typically involves adding additional irrelevant plan steps or formulating redundant transition questions. Third, optimistic monitor judgments occasionally trigger a premature transition when subtle geometric conditions—such as a bag resting on the edge of a table—are misclassified as success.

Several limitations temper the present findings. Our chosen evaluation task family, while non-trivial, is confined to tabletop pick-and-place scenes and therefore under-samples the combinatorial diversity of domestic manipulation. The 1Hz verification rate, although adequate for deliberative motions, would be insufficient for fast skills, such as rapid recovery motions. This implicitly defines the timescale of skill abstraction.

B. Future Work

Our modular architecture provides clear pathways for future enhancements:

Error detection and adaptive re-planning: A critical future enhancement includes adapting the VLM execution monitor system to explicitly detect execution failures or anomalous states [23, 26]. Through failure detection, we can trigger real-time adaptive replanning. This promises to improve the robustness of the system.

Enhanced semantic and contextual conditioning: Future work involves incorporating richer conditioning beyond visual and proprioceptive states from the high-level execution monitor. For example, replacing the ResNet backbone with that of CLIP [24] would allow the VLM to have finer-grained semantic control of the skills via text-conditioning.

Expanding the skill library: Finally, the current skill library is focused primarily on pick-and-place. An immediate next step involves expanding the robot’s skill repertoire. Examples include non-prehensile bi-manipulation and tool use. This broader skill set would enable more complex, realistic long-horizon tasks, requiring the planning and monitoring components to handle increasingly diverse and dynamic environments.

VI. CONCLUSION

In this work, we presented a hierarchical planning and control framework for multi-step humanoid manipulation tasks, combining robust RL-based whole-body tracking, imitation-learned skill policies, and a high-level vision-language planning and monitoring module. Evaluated on the Unitree G1 humanoid robot, our integrated system demonstrated successful autonomous execution of complex manipulation sequences, achieving a 73% success rate across real-world trials. Crucially, our approach offers inherent modularity, interpretability, and adaptability—enabling straightforward extension of the robot’s skill repertoire. This work highlights the effectiveness of integrating vision-language models with hierarchical control, paving the way toward increasingly capable autonomous humanoid systems operating in dynamic human environments.

REFERENCES

- [1] Constructions Aeronautiques, Adele Howe, Craig Knoblock, ISI Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins Sri, Anthony Barrett, Dave Christianson, et al. Pddl the planning domain definition language. *Technical Report, Tech. Rep.*, 1998.

- [2] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debiddatta Dwibedi, and Dorsa Sadigh. RT-H: Action Hierarchies Using Language. arXiv, June 2024. doi: 10.48550/arXiv.2403.01823. arXiv:2403.01823 [cs].
- [3] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. Pi0.5: a Vision-Language-Action Model with Open-World Generalization. arXiv, April 2025. doi: 10.48550/arXiv.2504.16054. arXiv:2504.16054 [cs].
- [4] 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, Alexander 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 Soricu, 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. arXiv, July 2023. doi: 10.48550/arXiv.2307.15818. arXiv:2307.15818 [cs].
- [5] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged Robot Vision-Language-Action Model for Navigation. arXiv, February 2025. doi: 10.48550/arXiv.2412.04453. arXiv:2412.04453 [cs].
- [6] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive Whole-Body Control for Humanoid Robots. arXiv, March 2024. doi: 10.48550/arXiv.2402.16796. arXiv:2402.16796 [cs].
- [7] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. page 02783649241273668. SAGE Publications Sage UK: London, England, 2023.
- [8] Murtaza Dalal, Min Liu, Walter Talbott, Chen Chen, Deepak Pathak, Jian Zhang, and Ruslan Salakhutdinov. Local Policies Enable Zero-shot Long-horizon Manipulation. arXiv, March 2025. doi: 10.48550/arXiv.2410.22332. arXiv:2410.22332 [cs].
- [9] Xiaolin Fang, Bo-Ruei Huang, Jiayuan Mao, Jasmine Shone, Joshua B. Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Keypoint Abstraction using Large Models for Object-Relative Imitation Learning. arXiv, October 2024. doi: 10.48550/arXiv.2410.23254. arXiv:2410.23254 [cs].
- [10] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. HumanPlus: Humanoid Shadowing and Imitation from Humans. arXiv, June 2024. doi: 10.48550/arXiv.2406.10454. arXiv:2406.10454 [cs].
- [11] Zhaoyuan Gu, Junheng Li, Wenlan Shen, Wenhao Yu, Zhaoming Xie, Stephen McCrary, Xianyi Cheng, Abdulaziz Shamsah, Robert Griffin, C. Karen Liu, Abderrahmane Kheddar, Xue Bin Peng, Yuke Zhu, Guanya Shi, Quan Nguyen, Gordon Cheng, Huijun Gao, and Ye Zhao. Humanoid Locomotion and Manipulation: Current Progress and Challenges in Control, Planning, and Learning. arXiv, April 2025. doi: 10.48550/arXiv.2501.02116. arXiv:2501.02116 [cs].
- [12] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. OmniH2O: Universal and Dexterous Human-to-Humanoid Whole-Body Teleoperation and Learning. arXiv, June 2024. doi: 10.48550/arXiv.2406.08858. arXiv:2406.08858 [cs].
- [13] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8944–8951. IEEE, 2024.
- [14] Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, Zeji Yi, Guannan Qu, Kris Kitani, Jessica Hodgins, Linxi "Jim" Fan, Yuke Zhu, Changliu Liu, and Guanya Shi. ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. arXiv, April 2025. doi: 10.48550/arXiv.2502.01143. arXiv:2502.01143 [cs].
- [15] Zhengmao He, Kun Lei, Yanjie Ze, Koushil Sreenath, Zhongyu Li, and Huazhe Xu. Learning visual quadrupedal loco-manipulation from demonstrations. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9102–9109. IEEE, 2024.
- [16] Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. ExBody2: Advanced Expressive Humanoid Whole-Body Control. arXiv, March 2025. doi: 10.48550/arXiv.2412.13196. arXiv:2412.13196 [cs].
- [17] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action

- Model. arXiv, September 2024. doi: 10.48550/arXiv.2406.09246. arXiv:2406.09246 [cs].
- [18] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3383–3393, 2021.
- [19] Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Raymond Yu, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, and Ankit Goyal. HAMSTER: Hierarchical Action Models For Open-World Robot Manipulation. arXiv, February 2025. doi: 10.48550/arXiv.2502.05485. arXiv:2502.05485 [cs].
- [20] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.
- [21] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. arXiv, August 2021. doi: 10.48550/arXiv.2108.10470. arXiv:2108.10470 [cs].
- [22] OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, A. J. Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, et al. GPT-4o System Card. arXiv, October 2024. doi: 10.48550/arXiv.2410.21276. arXiv:2410.21276 [cs].
- [23] Young-Jin Park, Hao Wang, Shervin Ardestir, and Navid Azizan. Quantifying representation reliability in self-supervised learning models. In *Uncertainty in Artificial Intelligence*, pages 2835–2860. PMLR, 2024.
- [24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [25] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. arXiv, August 2017. doi: 10.48550/arXiv.1707.06347. arXiv:1707.06347 [cs].
- [26] Apoorva Sharma, Navid Azizan, and Marco Pavone. Sketching curvature for efficient out-of-distribution detection for deep neural networks. In *Uncertainty in artificial intelligence*, pages 1958–1967. PMLR, 2021.
- [27] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolò Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi Robot: Open-Ended Instruction Following with Hierarchical Vision-Language-Action Models. arXiv, February 2025. doi: 10.48550/arXiv.2502.19417. arXiv:2502.19417 [cs].
- [28] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.
- [29] Zhutian Yang, Caelan Garrett, Dieter Fox, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Guiding Long-Horizon Task and Motion Planning with Vision Language Models. In *ICRA*. IEEE, 2025.
- [30] 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 for Robotics. arXiv, June 2024. doi: 10.48550/arXiv.2406.10721. arXiv:2406.10721 [cs].
- [31] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. arXiv, April 2023. doi: 10.48550/arXiv.2304.13705. arXiv:2304.13705 [cs].

APPENDIX

A. Low-Level Tracking Policy

In this section, we describe additional details on the low-level RL tracking policy.

Recall that the output of the low-level tracking policy consists of target joint positions that are fed into a PD controller. Given gains k_p^i, k_d^i , the torques τ_i on each joint i are computed as:

$$\tau_i = k_p^i \left(a_t^i - q_t^i \right) - k_d^i \dot{q}_t^i.$$

The PD controller gains for each joint are shown in Tab. I

TABLE I: PD controller gains for each joint.

Joint	k_p (Nm/rad)	k_d (Nms/rad)
Hip Yaw	100	2.5
Hip Roll	100	2.5
Hip Pitch	100	2.5
Knee	200	5.0
Ankle Pitch	20	0.2
Ankle Roll	20	0.1
Shoulder Pitch	90	2.0
Shoulder Roll	60	1.0
Shoulder Yaw	20	0.4
Elbow	60	1.0
Waist	400	5.0

1) *Policy design and training procedure*: The policy follows an actor-critic paradigm with a three-layer perception (MLP) backbone. Each hidden layer contains 256 units and ReLU activations. The system is trained by PPO in Isaac gym using a 0.02 integration timestep and 6144 parallel environments. The PPO hyperparameters used in training are shown in Tab. II.

TABLE II: PPO hyperparameters used in training.

Hyperparameter	Value
Discount Factor (γ)	0.99
GAE Parameter (λ)	0.95
Timesteps per Rollout	21
Epochs per Rollout	5
Minibatches per Epoch	4
Entropy Bonus (α_2)	0.01
Value Loss Coefficient (α_1)	1.0
Clip Range	0.2
Reward Normalization	Yes
Learning Rate	1×10^{-3}
# Environments	4096
Optimizer	Adam

To achieve robustness under real-world conditions, we randomize diverse physical parameters in training. Specifically, in each simulated episode, we include:

- **Gravity Variation**: We sample gravity vectors near $(0, 0, -9.81)$, shifting each component by up to ± 0.1 . This accounts for inertial or slight calibration errors.
- **Friction Coefficients**: Each environment has friction coefficients drawn from $[0.6, 2.0]$, modeling differences in ground contact or footwear slip.

- **Base Mass and Center of Mass**: We add up to 5kg of virtual mass to the robot’s torso and shift its center of mass by up to 7cm. This simulates payloads or hardware variations.

- **Push Disturbances**: At random intervals, a lateral velocity impulse (up to 0.3m/s) disturbs the robot, prompting the policy to recover balance.

- **Motor Strength Range**: Each joint’s torque capacity is scaled by a factor in $[0.8, 1.2]$. This reflects actuator performance variability.

- **Terrain Complexity**: Instead of a flat plane, we use a bumpy *trimesh* terrain. The robot must learn to adapt foot placement and maintain stability on uneven ground.

At each time step, the instantaneous reward is computed as a weighted sum of individual terms. Specifically, this is a combination of tracking terms and regularization terms. These are shown in Tab. III and Tab. IV.

TABLE III: Final tracking reward terms and weights.

Term	Expression	Weight
<i>Expression Goal \mathcal{G}^e</i>		
DoF Position	$\exp(-0.7 \ \mathbf{q}_{\text{ref}} - \mathbf{q}\)$	3.0
Keypoint Position	$\exp(-\ \mathbf{p}_{\text{ref}} - \mathbf{p}\)$	2.0
<i>Root Movement Goal \mathcal{G}^m</i>		
Linear Velocity	$\exp(-4.0 \ \mathbf{v}_{\text{ref}} - \mathbf{v}\)$	6.0
Velocity Direction	$\exp(-4.0 \cos(\mathbf{v}_{\text{ref}}, \mathbf{v}))$	6.0
Roll & Pitch	$\exp(- \Omega_{\text{ref}}^{\phi\theta} - \Omega^{\phi\theta})$	1.0
Yaw	$\exp(- \Delta y)$	1.0

TABLE IV: Regularization reward terms and weights.

Term	Expression	Weight
<i>Feet Related</i>		
Height	$\max(h_{\text{feet}} - 0.2, 0)$	2.0
Feet Air Time	T_{air}	10
Drag	$\ \mathbf{v}_{\text{foot}}\ \sim \mathbf{1}_{\text{new_contact}}$	-0.1
Feet Contact Force	$\ \mathbf{F}_{\text{feet}}^z\ \geq 5 \times \ \mathbf{F}_{\text{feet}}^{x,y}\ $	-3×10^{-3}
Stumble	$\mathbf{1}\{\ \mathbf{F}_{\text{feet}}\ > 4 \times F_{\text{feet}}^z\}$	-2
<i>Other Items</i>		
DoF Acceleration	$\ \dot{\mathbf{q}}_t\ ^2$	-3×10^{-7}
Action Rate	$\ \mathbf{a}_t - \mathbf{a}_{t-1}\ ^2$	-0.1
Energy	$\ \mathbf{q}\ ^2$	-1×10^{-3}
Collision	$\mathbf{1}_{\text{collision}}$	-1
DoF Limit Violation	$\mathbf{1}\{q_t \notin [q_{\min}, q_{\max}]\}$	-10
DoF Deviation	$\ \mathbf{q}_t - \mathbf{q}_{\text{default}}\ ^2$	-1.0
Vertical Lin. Velocity	v_z^2	-1.0
Horiz. Ang. Velocity	$\ \boldsymbol{\omega}_{xy}\ ^2$	-2.0
Projected Gravity	$\ \mathbf{g}_{xy}\ ^2$	-2.0

2) *Deployment evaluation*: Following successful validation of the tracking policy in simulation, we deploy the policy on the physical Unitree G1 robot. Example real-time teleoperation is shown in Fig. 4.

Quantitatively, we evaluate the tracking performance of our trained policy in simulation by computing the mean absolute error (MAE) for joint angles and keypoint positions over representative test motions. Specifically, we obtain an average

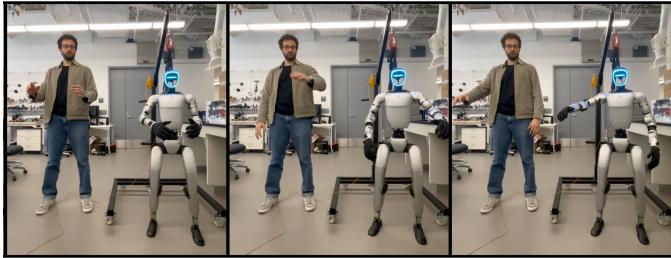


Fig. 4: Real-time teleoperation on the physical Unitree G1 robot. A Human operator is providing real-time motion commands captured via an RGB camera, and the robot is executing the corresponding commands accurately, demonstrating robust and responsive real-world tracking performance.

joint-angle error of 0.0593 rad (3.4°) and an average keypoint position error of 13.88 cm. These metrics indicate good tracking capabilities, validating our trained policy as suitable and reliable for real-world deployment on the humanoid robot.

B. Mid-Level Imitation Learning

In this section, we outline further implementation details for the mid-level imitation learning skill policies.

1) *Teleoperation Architecture*: An overall diagram for the RGB teleoperation architecture is shown in Fig. 5.

The joint retargeting procedure is outlined in Algorithm 1 while an example of the human pose estimation used for teleoperation is shown in Fig. 6.

Algorithm 1 Retargeting procedure: human pose to robot configuration

Input: Source skeleton tree S , source T-pose S_0 , target skeleton tree T , joint mapping M

Output: Retargeted skeleton state S_T

- 1: **Joint filtering:** Remove joints from S and S_0 that do not appear in the mapping M ;
 - 2: **Skeleton orientation alignment:** Compute alignment rotation R_{align} to align the human coordinate frame with the robot reference frame;
 - 3: Apply R_{align} to S and S_0 ;
 - 4: **Root translation normalization:** Extract global root translation t from S and scale to match robot proportions;
 - 5: **Global rotation computation:** Compute relative rotation R_{rel} from the human T-pose S_0 to the current pose S ;
 - 6: Apply R_{rel} to the robot T-pose to obtain R_{global} ;
 - 7: **Retargeted pose assembly:** Combine R_{global} and translation t to construct the retargeted pose S_T ;
 - 8: **Ground plane adjustment:** Adjust the z-axis of the translation in S_T to ensure feet contact the ground;
 - return** Retargeted skeleton state S_T
-

2) *Imitation Learning Policy Details*: Hyperparameters for the Humanoid Imitation Transformer (HIT) [10] implementation are outlined in Tab. V.

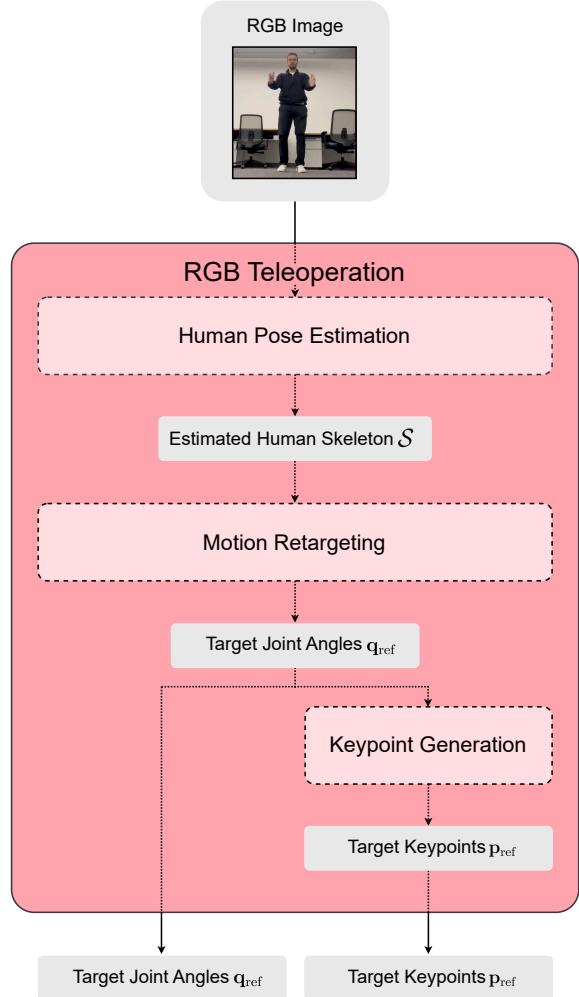


Fig. 5: Overview of the Teleoperation Module pipeline. Human poses captured by an RGB camera are first estimated and then retargeted to the robot’s kinematic structure. The resulting robot joint configurations are finally used to generate specific 3D keypoints required by the low-level tracking policy.

C. Walk-through Example of our System Pipeline

Example Pick-and-Place Task. Consider the illustrative task of *relocating an object from location A to location B* with the absence of obstacles. The hierarchical module operates as follows:

- 1) The VLM Planner $\Pi_{\text{vlm}}^{\text{Plan}}$ generates a sequence of skills $\{\pi_1, \pi_2, \dots, \pi_k\}$ based on visual inputs and task prompts. In this example, the sequence would be *pick*, followed by *place*.
- 2) The robot begins executing the first skill in the sequence, here the *pick* skill, governed by the trained imitation learning policy π_{pick} .
- 3) Concurrently, the VLM Skill Monitor actively observes the robot’s real-time visual feedback to assess whether the current skill (i.e., picking the object) has been completed. In this example, the monitor verifies whether the object



(a) Human operator



(b) HybrIK pose detection result

Fig. 6: Example of human pose detection by HybrIK, illustrating the original operator image (a) and corresponding detected skeletal configuration (b).

TABLE V: Training hyperparameters for HIT model.

Hyperparameter	Value
Chunk size	50
Hidden dimension	512
Feedforward dimension	512
Transformer decoder layers	6
Batch size	34
Learning rate	1×10^{-5}
Training steps	100,000
Backbone network	ResNet-18 (pretrained weights)
Positional embeddings	Enabled
Feature loss weight	0.005
Encoder module	Disabled (decoder-only architecture)
Random seed	0

is correctly grasped and above the target location.

- 4) If the Skill Monitor indicates the skill remains *in-progress*, the robot continues executing the current pick skill. Once the monitor confidently determines the completion of this skill, it signals the transition to the next planned skill.
- 5) Upon receiving a *completed* indication from the Monitor, the robot initiates the subsequent skill in the planned sequence—in this case, the place skill, governed by the corresponding policy π_{place} .

D. Example VLM prompts

Fig. 7 shows the system prompt used in OpenAI API, Fig. 8 shows the prompt for the planning task, while Fig. 9 shows an example GPT-4o response.

You are a helpful planning assistant for a robot. Your task is to create a sequence of actions (a plan) to achieve a given goal, based on the current environment shown in the image(s) and a set of available skills.

To achieve this, you will analyze the provided context, ground the skills, and MOST IMPORTANTLY, formulate the task of detecting skill success as a video question answering task. Each skill may have example questions to help ground the skills to the specific objects and actions in the context.

Output the plan as a JSON list, where each item in the list is an object representing a single step (a grounded skill). Each step object must have the following keys:

- "skill_name": The name of the skill used (from the available skills list).
- "description": A concise natural language description of *this specific action* being taken (e.g., "pick up the red block from the table").
- "preconditions": A natural language description of the state *required* just before executing *this specific action*, grounded to the objects involved (e.g., "the robot hand is empty and the red block is clear on the table").
- "effects": A natural language description of the state *resulting* from executing *this specific action*, grounded to the objects involved (e.g., "the robot hand is holding the red block").
- "question": A natural language question that can be used to verify if the action was successful – TRY TO FOLLOW SIMILAR QUESTION FORMATTING AND LANGUAGE AS THE EXAMPLE QUESTIONS PROVIDED IN THE SKILLS DESCRIPTION – the question should be specific and long enough to capture the essence of the action and its effects!

If one of the preconditions or effects is IMPORTANT, it must be formulated as part of the question, so that the question can be used to verify the success of the action.

Carefully consider the preconditions, effects and example question wording of the available skills to ensure the plan is logical and feasible, and the question makes sense given the context.

Fig. 7: Example system prompt for VLM skill planning.

```

Task:
  Pick up the bag and place it down on the white
  table.

Available Skills:
- name: "pick"
  description: "Pick up an object."
  preconditions:
    - "hand is empty"
    - "object is on a surface"
  effects:
    - "hand is holding object"
    - "object is no longer on its previous surface"
    - "(IMPORTANTLY) object is held up as far to the
      left as possible"

- name: "place"
  description: "Place a held object onto a surface
  ."
  preconditions:
    - "hand is holding object"
    - "target surface is clear"
    - "target surface is reachable"
  effects:
    - "hand is empty"
    - "object is on target surface"

- name: "push"
  description: "Push an object from one location to
  another."
  preconditions:
    - "hand is empty"
    - "object is pushable"
    - "object is reachable"
    - "target location is clear"
    - "object is at its start location"
  effects:
    - "object is at target location"

Generate the plan as a JSON list:

```

Fig. 8: Example generated prompt for VLM skill planning.

```

[
SkillDescription(
  name='pick',
  description='Pick up the bag from the box.',
  preconditions="The robot's hand is empty and the
  bag is on the box.",
  effects="The robot's hand is holding the bag up
  as far to the left as possible, and the bag
  is no longer on the box.", question='Has the
  robot finished picking up the bag and is
  holding the bag up to the left as far as
  possible?'
),
SkillDescription(
  name='place',
  description='Place the bag onto the white table
  .',
  preconditions="The robot's hand is holding the
  bag, the white table is clear, and the table
  is reachable.",
  effects="The robot's hand is empty and the bag is
  on the white table.",
  question="Is the bag now placed on the white
  table and the robot's hand empty?"
)
]
```

Fig. 9: Example VLM skill planning results.