ONEVLA: A UNIFIED FRAMEWORK FOR EMBODIED TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Navigation and manipulation are fundamental capabilities of embodied intelligence, enabling robots to interpret natural language commands and interact physically with their surroundings. However, current Vision-Language-Action (VLA) models remain constrained by task-specific architectures, specializing in either navigation or manipulation, which hinders the development of general-purpose robotic agents. To bridge this gap, we introduce ***OneVLA***, a unified architecture that integrates these distinct tasks into a single, cohesive framework. Specifically, we design a unified action head capable of generating both navigation and manipulation actions without requiring task-specific variants. Furthermore, we propose a multi-stage progressive training strategy—incorporating curated data construction and Chain-of-Thought (CoT) fine-tuning—that facilitates strong positive transfer and mutual reinforcement between the two domains. Extensive experiments in both simulated and real-world environments demonstrate that ***OneVLA*** achieves state-of-the-art performance, significantly outperforming both specialized single-task and existing cross-task models. By unifying these core capabilities, ***OneVLA*** paves the way for truly general-purpose robotic systems. The source code will be publicly released.

1 INTRODUCTION

Embodied intelligence, which endows intelligent agents with the ability to physically interact with the real-world environment and humans, is a key pathway to achieving artificial general intelligence (AGI) (Wu et al., 2025b; Liu et al., 2025a; Tan et al., 2025). In recent years, advancements in vision-language-action (VLA) (Liu et al., 2025c; Bu et al., 2025b; Wang et al., 2025; Hirose et al., 2025) models have significantly enhanced the performance of embodied intelligence tasks such as navigation and manipulation. VLA models combine the semantic understanding capabilities of vision-language models (VLMs) (Liu et al., 2025b; Goyal et al., 2025; Bai et al., 2025a) with perception and action generation capabilities. These models excel at translating natural language instructions into physical actions and have demonstrated impressive performance in various robotic tasks (Black et al., 2025; Bai et al., 2025b).

However, current VLA models suffer from a fundamental architectural limitation that prevents them from becoming truly general-purpose robotic foundation models (Zheng et al., 2025; Bu et al., 2025b). Existing methods can be broadly categorized into two types: single-task VLA models focusing on either navigation or manipulation, and cross-task VLA models that attempt to handle multiple embodiments. While cross-task models like UniVLA (Bu et al., 2025b) have made some progress in transferring knowledge across different robotic embodiments for navigation and manipulation tasks, they still require ***task-specific model variants***, and cannot seamlessly adapt to different embodied tasks. This limitation raises two key questions: (1) *Can we design a unified VLA architecture that can effectively perform navigation and manipulation tasks without task-specific model variants?* (2) *Can these two fundamental embodied intelligence tasks mutually benefit each other during training, thereby improving overall performance?*

To address this challenge, we propose ***OneVLA***, a unified VLA architecture capable of handling both navigation and manipulation tasks within a single framework while overcoming the limitations of task-specific approaches. As shown in Fig. 1, unlike previous methods that required ***separate action heads*** or ***specific model variants*** for different tasks, ***OneVLA*** utilizes an innovative unified action output head as its core architectural innovation: concatenating the action dimensions of navigation and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

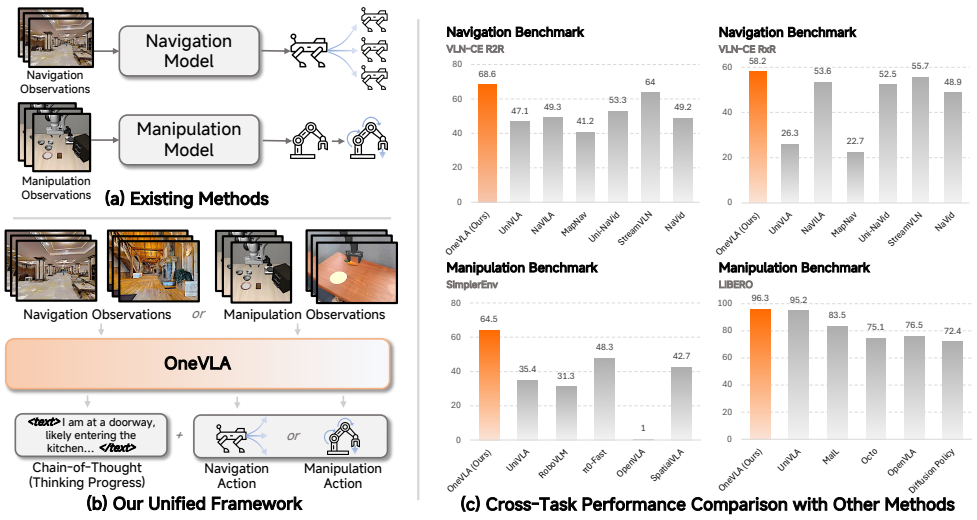

Figure 1: **Overview of *OneVLA*.** (a) Previous methods require building separate models for navigation and manipulation tasks, limiting the transfer of cross-task knowledge. (b) Our ***OneVLA*** proposes a **unified framework** that can handle navigation and manipulation tasks simultaneously in a single model, generating task-specific actions and general question-answering capabilities without task-specific model variants. (c) ***OneVLA*** achieves state-of-the-art (SOTA) performance on multiple navigation and manipulation benchmarks, significantly outperforming single-task-specific models and existing cross-task methods.

manipulation, thereby enabling the output of cross-task actions while maintaining independent loss calculations during training. To equip the model with cross-task capabilities, we adopt a three-stage progressive training strategy. The first stage learns fundamental manipulation skills. The second stage incorporates navigation data to improve cross-task generalization and navigation performance. The third stage introduces Chain-of-Thought (CoT) data for final refinement. In addition, we jointly train the backbone with a general vision–language objective, enabling shared vision–language representations and task-adaptive outputs. Through extensive experiments in both simulated and real-world environments, we demonstrated that ***OneVLA*** achieves state-of-the-art performance in both navigation and manipulation benchmarks, significantly outperforming single-task specialized models and cross-task models. More importantly, we provide empirical evidence that joint training on these two fundamental embodied tasks leads to mutual performance improvements, suggesting that a unified approach is not only more elegant architecturally but also superior in performance. Extensive experiments yielded two surprising findings: (1) *A unified VLA model can efficiently complete both navigation and manipulation tasks without task-specific model variants,* and (2) *Joint training of different embodied tasks can significantly improve the performance of each other.*

Our main contributions are summarized as follows:

- We propose ***OneVLA***, a unified VLA architecture that performs both navigation and manipulation tasks without task-specific model variants.
- We introduce an innovative unified action head and a multi-stage training strategy, enabling it to perform diverse embodied tasks and facilitating effective mutual enhancement between different task domains.
- Through comprehensive experiments, we demonstrate that ***OneVLA*** achieves state-of-the-art performance on multiple benchmarks and provide evidence that navigation and manipulation tasks can mutually benefit from each other during mixed training.
- We break down previous architectural barriers between independent task domains, establishing a new paradigm for embodied intelligence and paving the way for general-purpose robotic systems capable of holistic environmental interaction.

2 Related Work

VLA Models for Embodied Tasks. Vision-Language-Action (VLA) models (Team et al., 2024; Li et al., 2024; Hao et al., 2025; Zhang et al., 2025a; Liu et al., 2024b; Yang et al., 2025; Song et al., 2025) improve performance on embodied tasks by combining multimodal perception with motor control. However, recent methods generally focus on a single domain: in navigation, NaVid and Uni-NaVid (Zhang et al., 2024; 2025b) uses historical frame encoding to achieve temporally consistent navigation; MapNav (Zhang et al., 2025c) constructs explicit spatial maps to enhance path planning; and OmniVLA (Hirose et al., 2025) achieves cross-embodiment navigation by using goal-conditioned control. In manipulation, OpenVLA, RT-1-X, UniAct, etc. (Kim et al., 2025b; O'Neill et al., 2024; Zheng et al., 2025) achieve strong generalization capabilities through large-scale pre-training on large scale dataset (O'Neill et al., 2024; Wu et al., 2025a; Bu et al., 2025a); π_0 model (Black et al., 2024) employs a flow matching action head for precise control; and $\pi_{0.5}$ (Black et al., 2025) further improves open-world performance through co-training on heterogeneous long-horizon data. Despite these advances, all these VLA methods can only handle one task and lack cross-task capabilities. We propose ***OneVLA***, a unified VLA framework that can generate either navigation or manipulation actions within a single model, thus accomplishing both tasks.

Cross-Embodiment Learning. Cross-embodiment learning aims to transfer knowledge to different robotic platforms (Yang et al., 2024; Zheng et al., 2025; Bu et al., 2025b). Previous work has primarily focused on cross-embodiment generalization for single tasks: RT-1-X (O'Neill et al., 2024), OpenVLA (Kim et al., 2025b), $\pi_{0.5}$ (Black et al., 2025), etc., use a shared motion space to implement operations by different robotic arms, while OmniVLA (Hirose et al., 2025) provides a general navigation strategy for heterogeneous robots. Octo (Team et al., 2024), UniAct (Zheng et al., 2025), etc., can adapt to different view and motion spaces, but are still limited to manipulation tasks. General strategies such as π series (Black et al., 2024; 2025) can handle robotic manipulation or mobile manipulation tasks with different degrees of freedom, but mainly focus on manipulation tasks and lack fine-grained embodied navigation capabilities. Recent cross-task studies, such as ECE (Yang et al., 2024), CrossFormer (Doshi et al., 2025) and UniVLA (Bu et al., 2025b), while possessing cross-task capabilities, still require ***separate action heads*** or ***specific model variants*** for different tasks. These methods face significant limitations across embodiments and tasks. Our ***OneVLA*** framework overcomes these challenges through a unified architecture and action head, enabling navigation or manipulation tasks to be performed effectively without different action heads or specific model variants.

3 Methodology

3.1 Overview

We propose ***OneVLA***, a unified VLA framework capable of simultaneously generating both language responses and robotic actions for multiple embodied tasks. As shown in Fig. 2, at each time step t, given multimodal inputs including visual observations o_t (multi-view RGB images), natural language instructions l_t, and optional robot state r_t, our model generates a dual output: a text response y_t explaining the robot's understanding of the instruction; and a sequence of future actions $a_{t:t+T}$ for execution, in the following form: OneVLA : $(o_t, l_t, r_t) \rightarrow (y_t, a_{t:t+T})$. The action output $a_{t:t+T}$ can be adapted to different task domains without modifying the architecture. For embodied navigation tasks, the actions are processed into discrete movement commands represented as a probability distribution $a_{navi} = [p_{forward}, p_{turn_left}, p_{turn_right}, p_{stop}]$. For embodied manipulation tasks, the actions are continuous 7-DOF end-effector controls $a_{mani} = [\Delta x, \Delta y, \Delta z, \Delta\text{roll}, \Delta\text{pitch}, \Delta\text{yaw}, g]$. To achieve this dual output generation, we introduce special tokens <text>...</text><action>...</action>, which explicitly distinguish the language generation phase and the action generation phase within a single forward pass.

3.2 OneVLA Architecture

Unified Vision-Language Encoder. ***OneVLA*** employs a unified vision-language backbone to encode multimodal inputs from both navigation and manipulation tasks. Given multi-view visual observations $o_t = \{I_1, I_2, \ldots, I_M\}$ and a language instruction l_t, visual encoder first extracts image

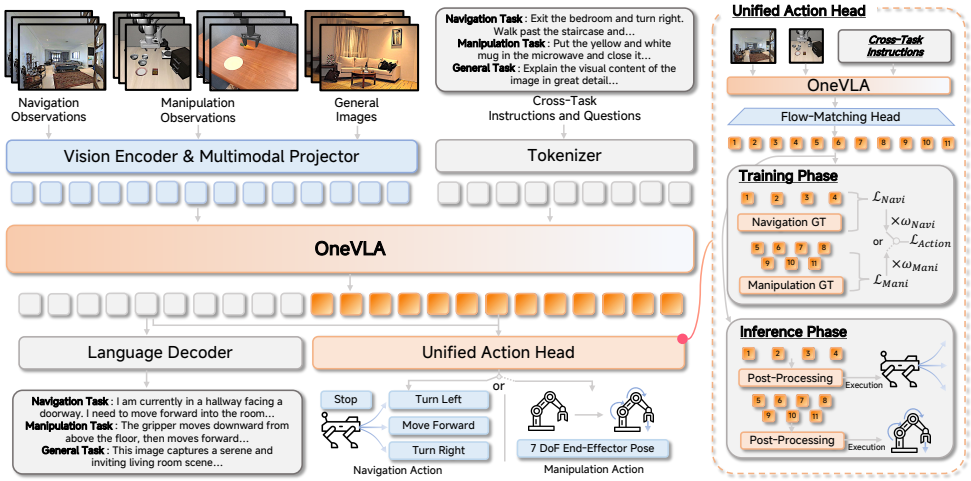

Figure 2: **Overall Framework of *OneVLA*.** For multimodal inputs in navigation or manipulation tasks (multi-view RGB images and natural language instructions), ***OneVLA*** utilizes a unified vision encoder and tokenizer to extract cross-task vision-language features. A unified action head with a flow matching mechanism generates task-specific actions within an 11-dimensional unified action space: discrete navigation commands or continuous 7-DOF manipulation control. During training, we use task-specific weight masks to compute independent losses for different tasks, allowing the model to learn both tasks simultaneously without changing the architecture.

features from all views, which are then projected into language-model embedding space:

$$V = \mathrm{Proj}(\mathrm{VisionEncoder}(o_t)). \tag{1}$$

The projected visual tokens are concatenated with the embedded language tokens and processed by the vision-language backbone:

$$H = \mathrm{OneVLA}([V; T]), \tag{2}$$

where T denotes the tokenized language instruction and H is the resulting multimodal representation shared by subsequent language generation and action prediction modules.

This unified encoder allows navigation, manipulation, and general vision-language samples to be represented in the same latent space, enabling cross-task feature sharing without introducing task-specific visual or language encoders. We provide the detailed backbone configuration in Appendix B.

Output Generation. To achieve simultaneous generation of language responses and robot actions, we introduce a token-based decoding mechanism that explicitly separates the two output modalities during a single forward propagation. This design allows the model to first reason about the task through language generation, and then generate the corresponding action sequence based on visual observation and the generated reasoning results. We expand the vocabulary with four special tokens: <text>...</text><action>...</action>. During inference, these tokens structure the autoregressive generation process:

$$\text{sequence} = \langle\text{text}\rangle\ y_t\ \langle/\text{text}\rangle\ \langle\text{action}\rangle\ \hat{a}_{t:t+T}\ \langle/\text{action}\rangle, \tag{3}$$

where y_t represents the text response notation explaining the robot's understanding, and $\hat{a}_{t:t+T}$ represents the discretized action notation. The complete hidden state $\mathbf{H} = \mathrm{OneVLA}([\mathbf{V}; \mathbf{T}]) \in \mathbb{R}^{(N_v+N_t)\times d_h}$ provided by the vision-language encoder serves as the conditional vector for the action decoder, effectively capturing joint reasoning from both vision and language context.

Unified Action Head. The core innovation of ***OneVLA*** lies in its unified action head, which seamlessly handles heterogeneous action spaces across different embodied tasks without requiring architectural modifications for specific tasks. Unlike previous works that required separate policy heads for navigation and manipulation, our method concatenates the action dimensions of these two tasks into a unified action space, while maintaining independence of loss calculations during training:

$$\mathbf{a}_{\text{unified}} = [\mathbf{a}_{\text{navi}} \mid \mathbf{a}_{\text{mani}}] \in \mathbb{R}^{11}, \tag{4}$$

where $\mathbf{a}_{\text{navi}} \in \mathbb{R}^4$ represents the navigation action, which is the probability distribution of discrete commands $[p_{\text{forward}}, p_{\text{turn left}}, p_{\text{turn right}}, p_{\text{stop}}]$; and $\mathbf{a}_{\text{mani}} \in \mathbb{R}^7$ represents the manipulation action, which is the continuous 7-DOF end-effector control $[\Delta x, \Delta y, \Delta z, \Delta\text{roll}, \Delta\text{pitch}, \Delta\text{yaw}, g]$.

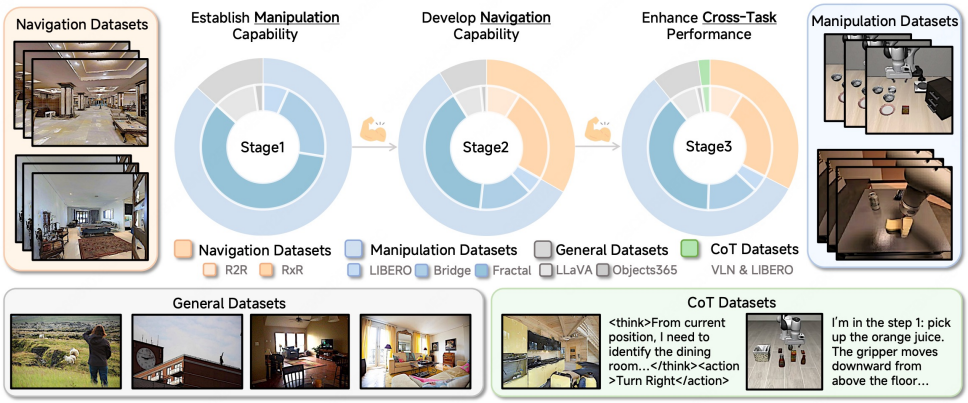

Figure 3: **Multi-Stage Training Strategy of *OneVLA*.** We propose a three-stage progressive training method: the first stage uses manipulation and general visual question answering (VQA) datasets to establish basic manipulation capabilities; the second stage introduces navigation data to develop navigation capabilities and achieve cross-task knowledge transfer through joint training; the third stage integrates Chain-of-Thought (CoT) datasets to enhance reasoning abilities and cross-task performance, ultimately yielding a unified model that performs well in both navigation and manipulation without requiring task-specific model variants.

Given the complete hidden state $\mathbf{H}$ provided by the visual language encoder, the action head employs a flow-matching diffusion process to predict future action sequences. Based on the flow-matching formula, we sample noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and interpolation time steps $t \sim \text{Beta}(\alpha, \beta)$, then construct a noisy trajectory $\mathbf{a}_t = (1-t)\boldsymbol{\epsilon} + t\mathbf{a}_{\text{gt}}$, where $\mathbf{a}_{\text{gt}}$ is the true action sequence. The model learns to predict the velocity field $\mathbf{v}_\theta = \mathbf{a}_{\text{gt}} - \boldsymbol{\epsilon}$ by minimizing the flow-matching objective. The architecture comprises an action encoder that conditions the noisy action trajectory on time-steps using sinusoidal position encoding; a DiT backbone network with cross-attention layers that fuses the encoded action with the visual language features $\mathbf{H}$; and an action decoder that projects the diffused output into an 11-dimensional unified action space. During inference, the model uses Euler integrals to iteratively denoise N time steps: $\mathbf{a}_{t+\Delta t} = \mathbf{a}_t + \Delta t \cdot \mathbf{v}_\theta(\mathbf{a}_t, t, \mathbf{H})$, progressively refining the action prediction from random noise to the final output.

To address the semantic differences between discrete navigation actions and continuous manipulation control, we introduce dimension-specific loss weighting. For a training sample i with task type $\tau_i \in \{\text{navi}, \text{mani}\}$, we define a task-specific weight mask to determine which dimensions contribute to the loss. The action loss $\mathcal{L}^i_{\text{action}}$ for each sample is calculated as follows:

$$\mathcal{L}^i_{\text{action}} = \text{mean}(\mathbf{w}_{\tau_i} \odot (\hat{\mathbf{v}}^i_\theta - \mathbf{v}^i)^2), \tag{5}$$

where $\odot$ denotes element-wise multiplication, and $\mathbf{w}_{\tau_i} \in \{\mathbf{w}_{\text{navi}}, \mathbf{w}_{\text{mani}}\}$ is the task-specific weight mask. Key dimensions (such as the stopping action) are assigned higher weights to address class imbalance, while padding dimensions have near-zero weights to ensure they don't contribute to gradients during training. This design allows the model to learn two tasks simultaneously while respecting their unique action space features. During each forward propagation, model generates actions for a single task based on input instructions, with task-specific weight masks automatically selecting relevant action dimensions and applying appropriate supervision. This unified architecture eliminates the need for task-specific model variants or switching—task differentiation naturally arises from the learned cross-modal representations and weighted training objectives.

Loss Function Design. The model is trained end-to-end using a composite loss function balancing the three objectives:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{vlm}}\mathcal{L}_{\text{vlm}} + \lambda_{\text{text}}\mathcal{L}_{\text{text}} + \lambda_{\text{action}}\mathcal{L}_{\text{action}}, \tag{6}$$

where λ_{vlm}, λ_{text}, and λ_{action} are weight coefficients used to control contribution of each component.

To preserve the pre-trained vision language understanding capabilities and enhance cross-modal reasoning, we incorporated general vision language data (image descriptions, visual question answering) into the training of the model. For these samples, we only computed the VLM loss as:

$$\mathcal{L}_{\text{vlm}} = -\sum_{i=1}^{N_{\text{vlm}}} \log p_\theta(y^i_{\text{vlm}} \mid \mathbf{o}_i, \mathbf{l}_i), \tag{7}$$

where N_{vlm} is the number of general VQA samples in the batch, and y^i_{vlm} is the target text response. This loss function applies standard cross-entropy to the generated tokens, enabling the backbone network to maintain strong vision-language alignment.

For embodied task samples (navigation and manipulation), we supervise the language reasoning phase to encourage explicit understanding of the task. We employ a standard next-token prediction objective that covers the entire reasoning sequence, including the boundary tokens (<text>, </text>) as well as the internal content:

$$\mathcal{L}_{\text{text}} = -\sum_{i=1}^{N_{\text{task}}} \sum_{j=1}^{|y_t^i|} \log p_\theta(y^i_{t,j} \mid \mathbf{H}^i, y^i_{t,<j}), \tag{8}$$

where N_{task} is the number of embodied task samples, and $|y_t^i|$ is the length of the inference text for sample i. $\mathcal{H}^i$ represents the multimodal hidden state of the encoder. This encourages the model to understand and express the action before generating it, thereby improving interpretability and reasoning capabilities.

For action generation, we compute the flow-matching loss using task-specific weight masks to accommodate the heterogeneous output spaces. The loss is defined as:

$$\mathcal{L}_{\text{action}} = \frac{1}{N_{\text{task}}} \sum_{i=1}^{N_{\text{task}}} \text{mean}\left(\mathbf{w}_{\tau_i} \odot (\hat{\mathbf{v}}^i_\theta - \mathbf{v}^i)^2\right), \tag{9}$$

where $\mathbf{w}_{\tau_i} \in \{\mathbf{w}_{\text{navi}}, \mathbf{w}_{\text{mani}}\}$ is the task-specific weight mask, $\mathbf{v}^i = \mathbf{a}^i_{\text{gt}} - \boldsymbol{\epsilon}^i$ is the ground truth velocity field, and $\hat{\mathbf{v}}^i_\theta$ is the model's prediction. The weight mask ensures that only task-relevant action dimensions contribute gradients, enabling joint training on heterogeneous action spaces.

3.3 Multi-Stage Training Strategy

As shown in Fig. 3, to enable ***OneVLA*** to effectively learn navigation and manipulation capabilities within a unified framework, we designed a three-stage progressive training strategy, gradually introducing task complexity and cross-task knowledge transfer.

Establishing Manipulation Capabilities. In the first stage, we focus on building robust manipulation skills, training solely using manipulation datasets and general vision-language datasets. The training data mixes manipulation task samples from an open-source cross-embodiment dataset with general visual question answer (VQA) samples. In this stage, the model learns language instructions through actions, translating them into continuous 7-DOF end-effector control while maintaining a strong understanding of visual language. The loss function for the first stage is:

$$\mathcal{L}_{\text{stage1}} = \lambda_{\text{vlm}}\mathcal{L}_{\text{vlm}} + \lambda_{\text{action}}\mathcal{L}^{\text{mani}}_{\text{action}}, \tag{10}$$

where the action loss $\mathcal{L}^{\text{mani}}_{\text{action}}$ uses a manipulation weight mask $\mathbf{w}_{\text{mani}}$ to supervise only the 7 degrees of freedom of the manipulation dimension.

Developing Navigation Capabilities. In Stage 2, we introduce navigation data into the joint training while retaining manipulation and general VQA data. The key innovation of this phase is that the unified action head can now learn navigation and manipulation tasks simultaneously through task-specific weight masking, thereby enabling knowledge transfer between the two domains. The joint dataset consists of three parts: (1) navigation task samples from VLN dataset; (2) manipulation task samples from Stage 1; and (3) general VQA samples. The loss function becomes:

$$\mathcal{L}_{\text{stage2}} = \lambda_{\text{vlm}}\mathcal{L}_{\text{vlm}} + \lambda_{\text{action}}(\mathcal{L}^{\text{navi}}_{\text{action}} + \mathcal{L}^{\text{mani}}_{\text{action}}), \tag{11}$$

where $\mathcal{L}^{\text{navi}}_{\text{action}}$ and $\mathcal{L}^{\text{mani}}_{\text{action}}$ use $\mathbf{w}_{\text{navi}}$ and $\mathbf{w}_{\text{mani}}$ respectively. Critically, joint training allows the vision-language encoder to learn shared representations, thus benefiting both tasks.

Enhancing Cross-Task Performance. In the final stage, we introduce Chain-of-Thought (CoT) data to strengthen the model's reasoning capabilities and improve its decision-making quality across two tasks. The training mixed dataset now contains all four data types: navigation, manipulation, general VQA, and CoT. The loss function is as follows:

$$\mathcal{L}_{\text{stage3}} = \lambda_{\text{vlm}}\mathcal{L}_{\text{vlm}} + \lambda_{\text{text}}\mathcal{L}_{\text{text}} + \lambda_{\text{action}}(\mathcal{L}^{\text{navi}}_{\text{action}} + \mathcal{L}^{\text{mani}}_{\text{action}}), \tag{12}$$

Table 1: **Performance Comparison of Models on Navigation and Manipulation Tasks.** The ***Unified*** column indicates whether a model supports both navigation and manipulation tasks without task-specific model variants. UniVLA-Navi.* and UniVLA-Mani.* represent UniVLA's (Bu et al., 2025b) performance using **task-specific fine-tuned action heads** on the same training data for navigation and manipulation respectively. **" - - "** indicates the model does not possess the capability to perform the specified task.

Method	Params	Unified	Navigation (VLN)		Manipulation (SimplerEnv)				
			R2R↑	RxR↑	Put Sp. on Tow.↑	Put Ca. on Pl.↑	Stack Gr. on Ye.↑	Put Egg. in Bask.↑	Avg.↑
Navigation VLA Models									
NaVid (Zhang et al., 2024)	7B	✗	49.2	48.9	- -	- -	- -	- -	- -
Uni-NaVid (Zhang et al., 2025b)	7B	✗	53.3	52.5	- -	- -	- -	- -	- -
NaVILA (Cheng et al., 2025)	7B	✗	62.5	53.6	- -	- -	- -	- -	- -
MapNav (Zhang et al., 2025c)	7B	✗	41.2	22.7	- -	- -	- -	- -	- -
StreamVLN-S.RGB (Wei et al., 2025)	7B	✗	64.0	55.7	- -	- -	- -	- -	- -
Manipulation VLA Models									
RT-1-X (O'Neill et al., 2024)	35M	✗	- -	- -	0.0	4.2	0.0	0.0	1.1
Octo-Base (Team et al., 2024)	93M	✗	- -	- -	12.5	8.3	0.0	43.1	16.0
π_0 (Black et al., 2024)	3B	✗	- -	- -	29.1	0.0	16.6	62.5	27.1
π_0-Fast (Pertsch et al., 2025)	3B	✗	- -	- -	29.1	21.9	10.8	66.6	32.1
OpenVLA (Kim et al., 2025b)	7B	✗	- -	- -	0.0	0.0	0.0	4.1	1.0
OpenVLA-OFT (Kim et al., 2025a)	7B	✗	- -	- -	34.2	30.0	30.0	72.5	41.8
RoboVLM (Liu et al., 2025b)	2B	✗	- -	- -	29.2	25.0	12.5	58.3	31.3
SpatialVLA (Qu et al., 2025)	7B	✗	- -	- -	16.7	25.0	29.2	100.0	42.7
Magma (Yang et al., 2025)	7B	✗	- -	- -	37.5	29.2	20.8	91.7	44.8
Cross-Task VLA Models									
UniVLA-Navi.* (Bu et al., 2025b)	7B	✗	47.1	26.3	- -	- -	- -	- -	- -
UniVLA-Mani.* (Bu et al., 2025b)	7B	✗	- -	- -	37.5	33.3	4.2	66.7	35.4
OneVLA (Our Unified VLA Model)	**3B**	✓	**68.6**	**58.2**	**87.5**	**37.5**	**50.0**	**83.3**	**64.5**
vs. UniVLA (Bu et al., 2025b)			***(+21.5)***	***(+31.9)***	***(+50.0)***	***(+4.2)***	***(+45.8)***	***(+16.6)***	***(+29.1)***

where $\mathcal{L}_{\text{text}}$ supervises text generation on CoT samples.

The multi-stage training strategy enables efficient development and knowledge transfer between embodied capabilities, ultimately generating a unified model that performs well in both navigation and manipulation ***without requiring model variants for specific tasks***.

4 Experiments

4.1 Simulated Benchmarks

Navigation Benchmarks. For navigation evaluation, we employed the VLN-CE (Krantz et al., 2020) benchmark, which extends room-to-room (R2R) (Krantz et al., 2020) and room-to-room (RxR) (Ku et al., 2020) datasets to the continuous action space within the Habitat simulator (Savva et al., 2019), requiring the agent to navigate to a target location following natural language instructions. We evaluated on the $\text{val}_{\text{unseen}}$ dataset, which contains previously unseen environments to test generalization capabilities. We used oracle success rate (OSR) as the primary metric, measuring the percentage of rounds in which the agent reached the target location within a specified distance threshold.

Manipulation Benchmark. To evaluate manipulation performance, we used the SimplerEnv-Bridge (Li et al., 2025) benchmark, which provides a realistic simulation environment capable of highly simulating the physical dynamics and visual appearance of the real world. We used a WidowX robotic arm equipped with the Bridge platform and focused on testing four representative tasks: (1) "Put Spoon on Towel", (2) "Put Carrot on Plate", (3) "Stack Green Block on Yellow Block", and (4) "Place Eggplant in Yellow Basket". Each task was run 24 times independently, and we report the success rate of each task as well as the average success rate of all four tasks.

4.2 Implementation Details

OneVLA uses Qwen2.5-VL-3B-Instruct (Bai et al., 2025a) as its unified vision-language backbone, with a hidden layer dimension of 2048. The unified action head adopts a DiT-B (Peebles & Xie, 2023) architecture with an 11-dimensional output space, which is composed of 4-dimensional navigation actions and 7-dimensional manipulation actions. We train the model using the AdamW (Loshchilov & Hutter, 2017) optimizer with learning rates of 4×10^{-5} (base model), 1×10^{-5} (vision-language

(a) **Ablation Study on Multi-Stage Training.** UniVLA-Navi.* and UniVLA-Mani.* represent UniVLA's (Bu et al., 2025b) performance with **task-specific fine-tuned action heads** on the same training data for navigation and manipulation respectively.

Method	Params	VLN R2R↑	VLN RxR↑	SimplerEnv Avg. SR↑
UniVLA-Navi.* (Bu et al., 2025b)	7B	47.1	26.3	- -
UniVLA-Mani.* (Bu et al., 2025b)	7B	- -	- -	35.4
OneVLA (Single-Stage Training)	3B	56.1	42.9	46.2
OneVLA (Multi-Stage Training)	3B	**68.6**	**58.2**	**64.5**
vs. Single-Stage Training		*(+12.5)*	*(+15.3)*	*(+18.3)*

(b) **Ablation Study on Cross-Task Joint Training.** UniVLA-Navi.* and UniVLA-Mani.* represent UniVLA's (Bu et al., 2025b) performance with **task-specific fine-tuned action heads** on the same training data for navigation and manipulation respectively.

Method	Params	VLN R2R↑	VLN RxR↑	SimplerEnv Avg. SR↑
UniVLA-Navi.* (Bu et al., 2025b)	7B	47.1	26.3	- -
UniVLA-Mani.* (Bu et al., 2025b)	7B	- -	- -	35.4
OneVLA (Navi. Only)	3B	48.8	33.6	- -
OneVLA (Mani. Only)	3B	- -	- -	40.7
OneVLA (Joint Training)	3B	**56.1**	**42.9**	**46.2**
vs. Single-Task Training		*(+7.3)*	*(+9.3)*	*(+5.5)*

interface), and $1{\times}10^{-4}$ (action model). Task sampling follows a 6:4 ratio of navigation data to manipulation data. The loss weights are set to $\lambda_{\text{action}} = 1, \lambda_{\text{text}} = \lambda_{\text{vlm}} = 0.1$. See App. B for more implementation details.

4.3 Comparison with SOTA Methods

As shown in Tab. 1, ***OneVLA*** achieves SOTA performance in both navigation and manipulation tasks benchmarks while maintaining a unified architecture without the need for task-specific model variants. Compared to UniVLA (7B) (Bu et al., 2025b), the current most competitive cross-task baseline which requires task-specific action head fine-tuning, our model demonstrates significant performance improvements: ***21.5***% increase on the VLN-CE R2R dataset (68.6% vs. 47.1%), ***31.9***% increase on the RxR dataset (58.2% vs. 26.3%), and ***29.1***% increase on the SimplerEnv dataset (64.5% vs. 35.4%). In addition to outperforming cross-task models, compared to single-task models, ***OneVLA*** achieves ***4.6***% and ***2.5***% improvements over the latest SOTA navigation method StreamVLN (7B) (Wei et al., 2025) on the R2R and RxR tasks, respectively; and ***19.7***% improvement in average success rate on the SimplerEnv manipulation task compared to the latest SOTA method Magma (Yang et al., 2025). Notably, our ***3B*** unified model outperforms all ***7B*** task-specific models, demonstrating that effective architecture design and training strategies can achieve superior performance with significantly fewer parameters and without the need for task-specific model variants.

4.4 Ablation Study

Effect of Multi-Stage Training. To validate the effectiveness of our proposed multi-stage training strategy, we conducted an ablation study comparing single-stage training with our three-stage progressive training method, where both approaches were trained on complete dataset for same steps. As shown in Tab. 2a, the multi-stage training strategy achieved significant improvements across all benchmarks: 12.5% improvement on the VLN-CE R2R task (68.6% vs. 56.1%), a 15.3% improvement on the RxR task (58.2% vs. 42.9%), and an 18.3% improvement on the SimplerEnv manipulation task (64.5% vs. 46.2%). These results demonstrate that gradually introducing task complexity leads to more effective knowledge transfer and learning compared to training with all data simultaneously from the beginning. Specifically, our strategy first establishes manipulation capabilities, then develops navigation skills, and finally leverages chain-of-thought data to enhance cross-task performance.

Effect of Cross-Task Joint Training. We compare single-task training (single dataset) with joint training (both datasets in single stage) under the same training steps. As shown in Tab. 2b, joint training significantly improved performance compared to training each task separately: R2R performance improved by 7.3% (56.1% vs. 48.8%), RxR performance improved by 9.3% (42.9% vs. 33.6%), and SimplerEnv task performance improved by 5.5% (46.2% vs. 40.7%). These results provide empirical evidence that navigation and manipulation tasks can mutually benefit from joint training, suggesting that shared vision-language representations learned from different embodied tasks contribute to better generalization and cross-task knowledge transfer.

Effect of Action Horizon. We conducted an ablation study to investigate the impact of action horizon on model performance in navigation and manipulation tasks. As shown in Fig. 4, we evaluated action horizon ranging from 2 to 8 time steps while keeping all other hyperparameters constant. The results show that action horizon=5 achieves the best balance, yielding optimal performance on both R2R (68.6%) and SimplerEnv (64.5%) tasks. Shorter horizons (2-4 steps) lead to suboptimal performance due to insufficient temporal modeling. Conversely, excessively long horizons (6-8 steps) result in

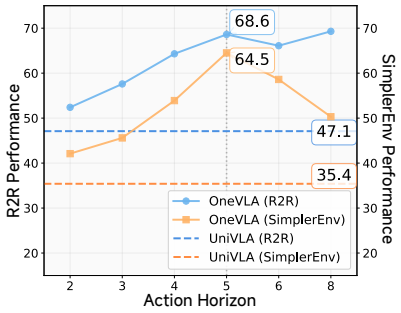

Figure 4: Ablation Study on Action Horizon.

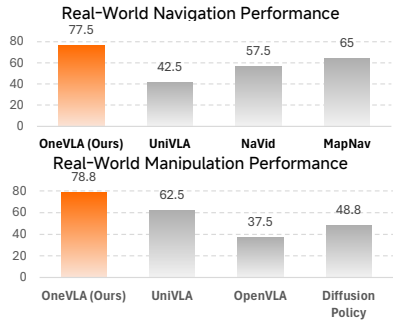

Figure 5: Real-World Experimental Results.

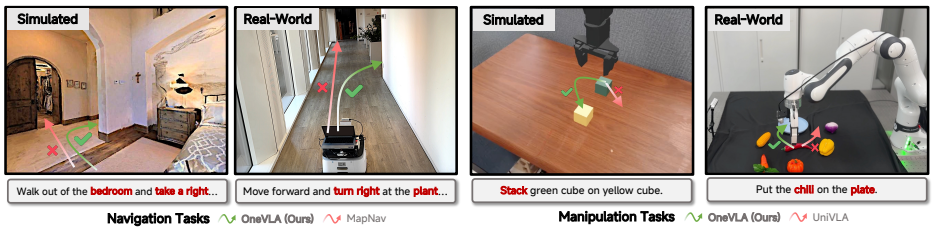

Figure 6: Qualitative Results in both Simulator and Real-World.

performance degradation, particularly in manipulation tasks, where SimplerEnv performance drops from 64.5% (horizon=5) to 50.3% (horizon=8). This is likely due to increased prediction difficulty and accumulated errors over longer time sequences.

4.5 Real-World Experiments

We conducted extensive real-world experiments for both navigation and manipulation tasks. For manipulation tasks, we designed four representative tasks and evaluated each task 20 times using a Franka Research 3 robotic arm, measuring the average success rate across all trials. For navigation tasks, we set up four different scenarios and tested each task 10 times using a mobile robot platform. As shown in Fig. 5, ***OneVLA*** performed exceptionally well in real-world environments, significantly outperforming existing methods on both types of tasks. In manipulation tasks, our model achieved a success rate of 78.8%, 16.3% improvement over UniVLA (Bu et al., 2025b). In navigation tasks, ***OneVLA*** achieved a success rate of 77.5%, 35.0% improvement over UniVLA (Bu et al., 2025b). These results demonstrate that ***OneVLA*** not only performs well in simulated benchmarks but also effectively generalizes to real-world scenarios.

4.6 Qualitative Analysis

Fig. 6 shows the qualitative results of ***OneVLA*** in both simulated and real-world environments. In the simulated navigation task, our model successfully understands the instructions and generated corresponding movement commands. In the manipulation task, ***OneVLA*** demonstrates precise object manipulation capabilities, accurately completing tasks such as stacking blocks and placing objects. More importantly, these capabilities effectively transferred to real-world scenarios. In real-world experiments, ***OneVLA*** maintains robust performance in both navigation and manipulation tasks, successfully executing complex instructions using a real robotic platform. These qualitative results demonstrate that our unified framework can handle various embodied tasks in different environments.

5 Conclusion

In this paper, we introduce ***OneVLA***, the first unified vision-language-action framework designed to handle both navigation and manipulation tasks simultaneously with a single model, eliminating the need for task-specific variants. Using an innovative unified action head and a progressive multi-stage training strategy, ***OneVLA*** achieves state-of-the-art performance across multiple benchmarks, showcasing the mutual benefits of joint training for navigation and manipulation. Extensive experiments in both simulated and real-world environments demonstrate that ***OneVLA*** significantly outperforms existing single-task and cross-task methods, paving the way for general-purpose robotic systems capable of comprehensive environmental interaction.

AI USE STATEMENT

We used vision-language models to generate chain-of-thought (CoT) annotations for navigation and manipulation training, as described in Appendix A.4. These annotations provide intermediate reasoning that connects visual observations and task instructions to actions, and are used for CoT fine-tuning. The authors take responsibility for the use of the generated annotations and for the accuracy and integrity of the final manuscript.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025a.

Shuanghao Bai, Wenxuan Song, Jiayi Chen, Yuheng Ji, Zhide Zhong, Jin Yang, Han Zhao, Wanqi Zhou, Zhe Li, Pengxiang Ding, et al. Embodied robot manipulation in the era of foundation models: Planning and learning perspectives. *arXiv preprint arXiv:2512.22983*, 2025b.

Johan Bjorck, Valts Blukis, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. GR00T N1.5: An improved open foundation model for generalist humanoid robots. NVIDIA Research, June 2025. URL `https://research.nvidia.com/labs/gear/gr00t-n1_5/`.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Annual Conference on Robot Learning*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025a.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *Proceedings of Robotics: Science and Systems*, 2025b.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *International Conference on 3D Vision*, pp. 667–676, 2017.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *Robotics: Science and Systems*, 2025.

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. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing, 2025. URL `https://github.com/starVLA/starVLA`.

Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Conference on Robot Learning*, pp. 496–512, 2025.

Ankit Goyal, Hugo Hadfield, Xuning Yang, Valts Blukis, and Fabio Ramos. Vla-0: Building state-of-the-art vlas with zero modification. *arXiv preprint arXiv:2510.13054*, 2025.

Peng Hao, Chaofan Zhang, Dingzhe Li, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Tla: Tactile-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2503.08548*, 2025.

Noriaki Hirose, Catherine Glossop, Dhruv Shah, and Sergey Levine. Omnivla: An omni-modal vision-language-action model for robot navigation. *arXiv preprint arXiv:2509.19480*, 2025.

Xiaogang Jia, Qian Wang, Atalay Donat, Bowen Xing, Ge Li, Hongyi Zhou, Onur Celik, Denis Blessing, Rudolf Lioutikov, and Gerhard Neumann. Mail: Improving imitation learning with selective state space models. In *Conference on Robot Learning*, pp. 3888–3907, 2025.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pp. 2679–2713, 2025b.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120, 2020.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4392–4412, 2020.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning*, pp. 3705–3728, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024a.

Huaping Liu, Di Guo, and Angelo Cangelosi. Embodied intelligence: A synergy of morphology, action, perception and learning. *ACM Computing Surveys*, 57(7), 2025a.

Huaping Liu, Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, and Hanbo Zhang. Towards generalist robot policies: What matters in building vision-language-action models. 2025b.

Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Pengju An, Xiaoqi Li, Kaichen Zhou, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation. *Advances in Neural Information Processing Systems*, 37:40085–40110, 2024b.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025c.

Qingxiang Liu, Ting Huang, Zeyu Zhang, and Hao Tang. Nav-r1: Reasoning and navigation in embodied scenes. *arXiv preprint arXiv:2509.10884*, 2025d.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2603.04356.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration0. In *IEEE International Conference on Robotics and Automation*, pp. 6892–6903, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems*, 2025.

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. In *Robotics: Science and Systems*, 2024.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9339–9347, 2019.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. *arXiv preprint arXiv:2508.10333*, 2025.

Huajie Tan, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Yaoxu Lyu, Mingyu Cao, Zhongyuan Wang, and Shanghang Zhang. Roboos: A hierarchical embodied framework for cross-embodiment and multi-agent collaboration. *arXiv preprint arXiv:2505.03673*, 2025.

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

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736, 2023.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems*, 2025a.

Yujie Wu, Huaihai Lyu, Yingbo Tang, Lingfeng Zhang, Zhihui Zhang, Wei Zhou, and Siqi Hao. Evaluating gpt-4o's embodied intelligence: A comprehensive empirical study. *TechRxiv preprint techrxiv.174495686.69962588/v1*, 2025b.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14203–14214, 2025.

Jonathan Yang, Catherine Glossop, Arjun Bhorkar, Dhruv Shah, Quan Vuong, Chelsea Finn, Dorsa Sadigh, and Sergey Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation. *arXiv preprint arXiv:2402.19432*, 2024.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *The Thirteenth International Conference on Learning Representations*, 2024.

Chaofan Zhang, Peng Hao, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Vtla: Vision-tactile-language-action model with preference learning for insertion manipulation. *arXiv preprint arXiv:2505.09577*, 2025a.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. In *Robotics: Science and Systems*, 2024.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. In *Robotics: Science and Systems*, 2025b.

Lingfeng Zhang et al. Mapnav: A novel memory representation via annotated semantic maps for vlm-based vision-and-language navigation. In *The Annual Meeting of the Association for Computational Linguistics*, pp. 13032–13056, 2025c.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22508–22519, 2025.

APPENDIX

This technical appendix provides additional details of dataset construction, implementation details, as well as experimental results that are omitted from the main body of this paper due to the page limit.

Specifically, this appendix is organized as follows:

- Sec. A provides details on the Dataset Construction.
- Sec. B outlines the Implementation Details.
- Sec. C performs More Experimental Results.
- Sec. D presents Real-World Experimental Results.
- Sec. E showcases further Visualization Results.
- Sec. F discusses Limitations and Broader Impacts.

A DATASET DETAILS

In this section, we will describe in detail the datasets used to train ***OneVLA***. As shown in Tab. 3, our training data consists of four main components: navigation datasets for developing navigation capabilities, manipulation datasets for learning fine-grained object interaction skills, general VQA datasets for vision-language understanding and a Chain-of-Thought (CoT) dataset for enhancing the model's reasoning and decision-making abilities. Below, we will describe the composition, processing pipeline, and characteristics of each type of dataset in detail.

A.1 NAVIGATION DATASETS

To conduct navigation training, we collected real trajectory data from two widely used vision-and-language navigation (VLN) benchmark datasets: Room-to-Room (R2R) (Krantz et al., 2020) and Room-across-Room (RxR) (Ku et al., 2020). The R2R dataset contains 10k trajectories in 90 indoor environments of the Matterport3D (Chang et al., 2017) scene dataset, each accompanied by natural language instructions averaging 29 words in English. The RxR dataset extends this to a multilingual setting, containing approximately 20k English instruction trajectories. Each original trajectory includes a natural language instruction, a sequence of discrete action IDs (0=stop, 1=move forward, 2=turn left, 3=turn right), and corresponding egocentric RGB observation images captured at each step.

Our data processing pipeline first converts the discrete action IDs into continuous 4-dimensional one-hot encoded action vectors, with the following mapping: stop $\rightarrow [0.0, 0.0, 0.0, 1.0]$, move forward $\rightarrow [1.0, 0.0, 0.0, 0.0]$, turn left $\rightarrow [0.0, 1.0, 0.0, 0.0]$, turn right $\rightarrow [0.0, 0.0, 1.0, 0.0]$. Each action vector is paired with its corresponding observation image. This preprocessing stage generates an indexed episode structure, where each episode contains the natural language instruction, a sequence of 4-dimensional action vectors, and the corresponding image filenames.

During training, we construct a single sample by selecting a step index from the trajectory and building a temporal window of observation data around that step. For a trajectory containing T action-observation pairs, we generate samples at steps $t \in \{0, 1, \ldots, T-1\}$ with a step size s (default $s = 1$ for dense sampling). At each step t, we construct the observation history by uniformly sampling H frames from all original frames before the current step. Specifically, assuming the current frame's original index is t_{orig}, we sample historical frames from the range $[0, t_{orig} - 1]$. The principle is to always include the first frame (index 0) to provide initial viewpoint context, while the remaining $H - 1$ frames are uniformly sampled within the range $[1, t_{orig} - 1]$ using linear interpolation, with indices $\lfloor \frac{i \cdot (t_{orig} - 1)}{H - 1} \rfloor$, where $i \in \{1, \ldots, H - 1\}$. This uniform sampling strategy ensures consistent temporal coverage regardless of trajectory length, while preserving the crucial starting viewpoint. The final observation sequence concatenates these historical frames with the current frame, forming an input containing $H + 1$ frames.

For action supervision, each sample at step t predicts a sequence of future actions $\{a_t, a_{t+1}, \ldots, a_{t+C-1}\}$, where C is the action block size. If $t + C$ exceeds the trajectory length,

Table 3: Training dataset composition showing the distribution of samples across different data sources and task categories.

Dataset	Category	Proportion (%)
R2R (Krantz et al., 2020)	Navigation	8.0
RxR (Ku et al., 2020)	Navigation	44.2
LIBERO (Liu et al., 2023)	Manipulation	4.4
Bridge (Walke et al., 2023)	Manipulation	5.3
Fractal (Brohan et al., 2022)	Manipulation	8.9
R2R/RxR CoT	Navigation CoT	5.3
LIBERO CoT	Manipulation CoT	4.4
LLaVA-665K (Liu et al., 2024a)	General VQA	12.0
Objects365COCO (Community, 2025)	General VQA	7.5
Total	-	**100.0**

the sequence is padded with the last action (usually a stop action). To enable multi-dataset training with heterogeneous action spaces, we further pad the 4-dimensional navigation actions to an 11-dimensional unified action space by appending 7 zeros: $a_{nav} \in \mathbb{R}^4 \rightarrow a_{unified} \in \mathbb{R}^{11}$, where the first 4 dimensions correspond to navigation, and the remaining 7 dimensions are reserved for manipulation tasks. After preprocessing, the R2R training set contains approximately 600K step-level samples, while RxR contains approximately 1.2M samples.

A.2 Manipulation Datasets

To train our manipulation model, we utilized three large-scale robotic manipulation datasets: LIBERO (containing approximately 50,000 demonstrations using the Franka Panda robot), Bridge (containing approximately 60,000 trajectories using the WidowX robotic arm), and Fractal (containing approximately 100,000 demonstrations covering various manipulation scenarios). Each dataset provides RGB image observations, proprioceptive states, continuous action commands, and natural language task descriptions following the LeRobot standard format.

The actions in the manipulation tasks are represented as 7-DOF continuous control commands: $a_{mani} = [\Delta x, \Delta y, \Delta z, \Delta roll, \Delta pitch, \Delta yaw, gripper]$, specifying the end-effector pose change and gripper state. Our processing pipeline uses robot-platform-specific configurations to handle the heterogeneity of different robot platforms: LIBERO uses the `libero_franka` configuration, applying min-max normalization to the 8-dimensional proprioceptive states and 7-DOF actions; Bridge applies the `oxe_bridge` configuration, using quantile-based (q99) normalization to handle action outliers; Fractal follows the `oxe_rt1` configuration, which is tailored to its robot platform. During sampling, we extract the current observation at timestep t and predict $C = 5$ future action sequences. These 7-dimensional manipulation actions are extended to a unified 11-dimensional space by zero-padding dimensions 0-3 (reserved for navigation): $a_{mani} \in \mathbb{R}^7 \rightarrow a_{unified} \in \mathbb{R}^{11}$, to ensure consistent tensor shapes across different datasets.

We constructed a mixed dataset where samples from different sources are weighted and interleaved according to specified mixing configurations, which specify `(dataset_name, weight, robot_type)` tuples. Image observations are resized to 224×224 pixels and optionally augmented with random cropping (scale=0.95) and color jittering (brightness ±0.3, contrast ±0.4) to improve robustness. After preprocessing and constructing the mixed dataset, the final manipulation training set contains approximately 4 million step-level samples, covering different manipulation skills and embodiments.

A.3 General VQA Datasets

To preserve and enhance the vision-language understanding capabilities of the backbone, we introduce general VQA datasets during training. Specifically, we used LLaVA-665K (Liu et al., 2024a) and Objects365COCO (Community, 2025). They provide object-centric visual understanding data

covering a wide range of object categories. These general VQA samples are supervised only through the vision-language modeling loss $\mathcal{L}_{vlm}$, without action prediction, allowing the model to maintain strong cross-modal alignment and semantic understanding capabilities while learning embodied control.

Table 4: Key Training Hyperparameters of ***OneVLA***.

Hyperparameter	**Value**
Model Architecture	
Vision-Language Backbone	Qwen2.5-VL-3B (Bai et al., 2025a)
VL Hidden Dimension	2048
VL Number of Layers	36
Vision Encoder Depth	32
Vision Hidden Dimension	1280
Action Model Type	DiT-B
Action Hidden Dimension	2560
Action Model Layers	16
Unified Action Dimension	11
Action Horizon	5
Optimization	
Optimizer	AdamW
Base Learning Rate	4×10^{-5}
VL Interface Learning Rate	1×10^{-5}
Action Model Learning Rate	1×10^{-4}
LR Scheduler	Cosine with Min LR
Minimum Learning Rate	5×10^{-7}
Weight Decay	1×10^{-8}
Warmup Steps	5000
Max Gradient Norm	1.0
Training Configuration	
Mixed Precision	BF16
Gradient Checkpointing	Enabled
VLM Batch Size per Device	4
Navigation Batch Size per Device	2
Manipulation Batch Size per Device	16
Loss Configuration	
VLA Loss Weight (λ_{action})	1.0
VLM Loss Weight (λ_{vlm})	0.1
CoT Text Loss Weight (λ_{text})	0.1
Navigation Task Weight	6.0
Manipulation Task Weight	4.0
Navigation Dim Weights	[1.0, 1.0, 1.0, 15.0, 0.1^7]
Manipulation Dim Weights	[0.1^4, 1.0^7]
Diffusion Configuration	
Training Diffusion Steps	8
Inference Diffusion Steps	4
Noise Beta (α, β)	(1.5, 1.0)
Data Configuration	
Image Resolution	224×224
Model Max Length	2048

A.4 CoT Datasets

To enhance reasoning capabilities, we constructed a CoT dataset for navigation and manipulation tasks, which provides explicit reasoning processes between perception and action.

For navigation CoT data, we built upon NavCoT-110K (Liu et al., 2025d), utilizing a VLM to process egocentric RGB image observations and navigation instructions from R2R/RxR trajectories, generating structured reasoning text that clarifies spatial understanding and path planning decisions (e.g., "<text> I am at a doorway, facing a bedroom. I need to find the kitchen entrance on the left...</text>"). This automated annotation method generated approximately 80,000 navigation CoT samples containing intermediate reasoning processes for spatial decision-making.

For manipulation CoT data, we used a VLM to analyze complete LIBERO video sequences and segmented each trajectory into 3-5 subtask stages (e.g., "approach object", "grasp," "transfer," "place"). For each stage, the VLM generated detailed descriptive reasoning processes in the format <text>...</text>, explaining the perception and decision-making processes (e.g., "The robot arm end-

effector is above the red cube. I need to lower the end-effector while maintaining alignment..."). This process generated approximately 50,000 manipulation CoT samples. During training, CoT samples were mixed with standard data, and the model learned to generate both reasoning text (supervised by $\mathcal{L}_{text}$) and corresponding actions, thereby achieving explicit reasoning while maintaining action prediction performance.

B Implementation Details

We use Qwen2.5-VL-3B-Instruct (Bai et al., 2025a) as the vision-language backbone network and employ a 16-layer DiT-B architecture as the action model to implement ***OneVLA***. Complete configuration details can be found in Tab. 4. Following our progressive training strategy, we train ***OneVLA*** in three stages for a total of 60,000 steps: the first stage trains for 20,000 steps on manipulation and general VQA datasets to establish manipulation capabilities; the second stage jointly trains for 30,000 steps on navigation, manipulation, and VQA data to develop cross-task capabilities; and the third stage trains for 10,000 steps using an additional CoT dataset to enhance reasoning abilities. We use the AdamW optimizer with different learning rates: 4×10^{-5} (base model), 1×10^{-5} (VL interface), and 1×10^{-4} (action model). The learning rate follows a cosine decay strategy with a warm-up of 5,000 steps and a minimum learning rate of 5×10^{-7}. The loss weights are set to $\lambda_{\text{vlm}} = 0.1$ and $\lambda_{\text{action}} = 1.0$, with a task sampling ratio of 6:4 (navigation:manipulation). In the 11-dimensional action space, we apply dimension-specific weights: [1.0, 1.0, 1.0, 15.0, 0.1^7] for navigation tasks, where the stop action has a higher weight to address class imbalance; and [0.1^4, 1.0^7] for manipulation tasks to ensure correct gradient flow while preventing padding dimensions from affecting training. Training is performed on 4 nodes, each equipped with 8 H200 GPUs, using mixed precision (BF16) and gradient checkpointing. The batch sizes per device are 4 (VQA), 2 (navigation), and 16 (manipulation). The navigation samples use a 7-frame historical window, an image resolution of 224×224, and a 5-step action prediction horizon. The diffusion process uses 8 denoising steps during training and 4 steps during inference, employing Beta noise scheduling parameters $\alpha = 1.5$ and $\beta = 1.0$.

During inference, ***OneVLA*** employs task-specific post-processing strategies to convert the unified 11-dimensional action output into execution commands suitable for different robotic platforms. For ***Navigation Tasks***, the model outputs a 4-dimensional probability distribution corresponding to discrete movement commands $\mathbf{a}_{\text{nav}} = [p_{\text{forward}}, p_{\text{left}}, p_{\text{right}}, p_{\text{stop}}] \in \mathbb{R}^4$. The final action is determined by selecting the command with the highest probability:

$$a_{\text{exec}} = \arg\max_i \mathbf{a}_{\text{nav}}[i], \quad i \in \{0, 1, 2, 3\}, \tag{13}$$

where a_{exec} is mapped to the corresponding robot action (0=stop, 1=forward, 2=turn left, 3=turn right). To maintain temporal consistency during navigation, we uniformly sample H historical frames from the trajectory as input: $\{\mathbf{I}_0, \mathbf{I}_{\lfloor \frac{t_{\text{orig}}-1}{H-1} \rfloor}, \ldots, \mathbf{I}_{t_{\text{orig}}}\}$, where t_{orig} is the current frame index, and $\mathbf{I}_0$ is always included to preserve the initial viewpoint context. For ***Manipulation Tasks***, the model predicts a sequence of C continuous 7-DOF actions $\mathbf{A}_{\text{mani}} = [\mathbf{a}_1, \ldots, \mathbf{a}_C] \in \mathbb{R}^{C\times 7}$, where each $\mathbf{a}_t = [\Delta x, \Delta y, \Delta z, \Delta\text{roll}, \Delta\text{pitch}, \Delta\text{yaw}, g]$ represents the end-effector pose change and gripper state. First, the normalized predicted values $\tilde{\mathbf{A}}_{\text{mani}}$ are denormalized using dataset-specific statistics:

$$\mathbf{A}_{\text{mani}} = 0.5 \cdot (\tilde{\mathbf{A}}_{\text{mani}} + 1) \odot (\mathbf{a}_{\text{high}} - \mathbf{a}_{\text{low}}) + \mathbf{a}_{\text{low}}, \tag{14}$$

where $\mathbf{a}_{\text{high}}$ and $\mathbf{a}_{\text{low}}$ are the 99th and 1st percentile values of the training data, respectively, and $\odot$ denotes element-wise multiplication. To improve execution stability, we employ an exponentially weighted temporal action integration method: $\mathbf{a}_{\text{exec}} = \sum_{k=0}^{K-1} \alpha^k \mathbf{a}_{t+k} / \sum_{k=0}^{K-1} \alpha^k$, where K is the integration time window and $\alpha = 0.1$ is the decay factor. This adaptive integration mechanism effectively reduces prediction jitter while maintaining responsiveness to environmental changes.

C More Experiments

LIBERO Results. To comprehensively evaluate ***OneVLA***'s manipulation capabilities on complex, long-horizon tasks, we conducted extensive experiments on the LIBERO benchmark (Liu et al., 2023). LIBERO provides a highly challenging robotic manipulation testbed, comprising four distinct task suites to assess different aspects of embodied intelligence: **Spatial** (spatial reasoning and relational

Table 5: **Performance Comparison on LIBERO Benchmark.** We evaluate on four representative long-horizon manipulation tasks from the LIBERO suite and report the success rate for each task along with the average performance.

Method	Params	Spatial↑	Object↑	Goal↑	Long↑	Avg.↑
LAPA (Ye et al., 2024)	7B	73.8	74.6	58.8	55.4	65.7
Diffusion Policy (Chi et al., 2025)	300M	78.3	92.5	68.3	50.5	72.4
Octo (Team et al., 2024)	93M	78.9	85.7	84.6	51.1	75.1
MDT (Reuss et al., 2024)	7B	78.5	87.5	73.5	64.8	76.1
OpenVLA (Kim et al., 2025b)	7B	84.7	88.4	79.2	53.7	76.5
MaIL (Jia et al., 2025)	25M	74.3	90.1	81.8	78.6	81.2
UniVLA (Bu et al., 2025b)	7B	96.5	96.8	95.6	92.0	95.2
***OneVLA* (Ours)**	3B	**98.2**	**97.8**	**98.1**	**96.5**	**97.7**
vs. UniVLA (Bu et al., 2025b)		***(+1.7)***	***(+1.0)***	***(+2.5)***	***(+4.5)***	***(+2.5)***

understanding), **Object** (object-centric manipulation and recognition), **Goal** (goal-oriented planning and execution), and **Long-Horizon** (long-horizon sequential task completion). Each suite contains multiple complex tasks requiring precise control, temporal reasoning, and strong generalization capabilities. As shown in Table 5, ***OneVLA*** achieved state-of-the-art performance on the LIBERO benchmark, with an average success rate of 97.7%, setting a new benchmark for unified embodied intelligence models. Compared to the most competitive cross-task baseline, UniVLA (Bu et al., 2025b) with 7B parameters, our 3B parameter unified model demonstrated consistent improvements across all four task suites: 1.7 percentage points on spatial tasks (98.2% vs. 96.5%), 1.0 percentage point on object tasks (97.8% vs. 96.8%), 2.5 percentage points on goal tasks (98.1% vs. 95.6%), and 4.5 percentage points on long-horizon tasks (96.5% vs. 92.0%). Overall, despite having less than half the parameters and requiring no task-specific model variants, our model still outperformed UniVLA by 2.5 percentage points in average success rate. Compared to single-task manipulation expert models, ***OneVLA*** demonstrates significant advantages. Our model's average performance is 21.2 percentage points higher than OpenVLA (7B) (Kim et al., 2025b) (97.7% vs. 76.5%), and 21.6 percentage points higher than MDT (7B) (Reuss et al., 2024) (97.7% vs. 76.1%). Compared to the widely used imitation learning baseline Diffusion Policy (Chi et al., 2025), ***OneVLA*** demonstrates a significant performance improvement of 25.3 percentage points (97.7% vs. 72.4%). The results demonstrate that ***OneVLA*** not only achieves architectural unification but also delivers tangible performance improvements through effective cross-task learning.

RoboCasa365 Pretraining Evaluation. Table 6 reports results for multi-task learning on RoboCasa365 (Nasiriany et al., 2026), using human demonstrations from 300 pretraining tasks (Human300). Evaluation covers 50 tasks in pretraining kitchens, including atomic tasks, seen composite tasks, and unseen composite tasks that test zero-shot compositional generalization.

For mobile manipulation, we retain the same training strategy and apply the dimension-specific loss weighting from Section 3.2 according to the current action phase: during base movement, the manipulation action dimensions receive a loss weight of 0.1; during manipulation, the navigation action dimensions receive a loss weight of 0.1. As shown in Table 6, ***OneVLA*** achieves a task-weighted average success rate of 28.4%, exceeding UniVLA (9.9%) by 18.5 percentage points and the strongest baseline reported in the original paper, GR00T N1.5 (20.0%), by 8.4 percentage points. OneVLA outperforms UniVLA across all three task splits, including Composite-Unseen (7.9% vs. 1.1%). These results extend the evidence for cross-task transfer on separate navigation and manipulation benchmarks to mobile manipulation, supporting the applicability of our unified action formulation and training strategy to tasks that combine movement and manipulation.

Effect of the OneVLA Framework To further examine whether the performance gains of OneVLA simply come from the proposed unified action head, we conduct an additional ablation by directly attaching our 11-dimensional flow-matching action head to UniVLA. Specifically, we replace UniVLA's task-specific action heads with our unified action head and train the resulting model on the same mixed navigation-manipulation data. This variant removes the task-specific heads of UniVLA, but does not adopt the full OneVLA design, including our unified text-action output formulation, progressive training strategy, and CoT-based refinement.

Table 6: **RoboCasa365 Pretraining Evaluation.** Task success rates (%) on the Human300 multi-task benchmark, evaluated in pretraining kitchens. The first four rows reproduce results from Table 1 of the original RoboCasa365 paper (Nasiriany et al., 2026). Avg. is the task-weighted average over all 50 tasks.

Method	**Atomic↑** (18 tasks)	**Composite-Seen↑** (16 tasks)	**Composite-Unseen↑** (16 tasks)	**Avg.↑**
Diffusion Policy (Chi et al., 2025)	15.7	0.2	1.25	6.1
π_0 (Black et al., 2024)	36.3	5.2	0.7	15.0
$\pi_{0.5}$ (Black et al., 2025)	39.6	7.1	1.2	16.9
GR00T N1.5 (Bjorck et al., 2025)	43.0	9.6	4.4	20.0
UniVLA (Bu et al., 2025b)	23.2	3.8	1.1	9.9
OneVLA (Ours)	**53.4**	**20.8**	**7.9**	**28.4**

Table 7: **Ablation Study on Action Mask Weights.** We evaluate the impact of different padding weight values and the stop action weight in the 11-dimensional unified action space.

w_{navi}	w_{mani}	**VLN R2R**	**VLN RxR**	**SimplerEnv Avg. SR↑**	**Total Avg.↑**
Effect of padding weight (stop weight fixed at 15.0)					
$[1.0, 1.0, 1.0, 15.0, 1.0^7]$	$[1.0^4, 1.0^7]$	49.2	35.8	38.7	41.2
$[1.0, 1.0, 1.0, 15.0, 0.5^7]$	$[0.5^4, 1.0^7]$	56.3	44.6	51.2	50.7
$[1.0, 1.0, 1.0, 15.0, 0.1^7]$	$[0.1^4, 1.0^7]$	**68.6**	**58.2**	**64.5**	**63.8**
$[1.0, 1.0, 1.0, 15.0, 0.0^7]$	$[0.0^4, 1.0^7]$	50.7	37.4	40.3	42.8
Effect of stop action weight (padding weight fixed at 0.1)					
$[1.0, 1.0, 1.0, 1.0, 0.1^7]$	$[0.1^4, 1.0^7]$	51.3	39.8	48.2	46.4
$[1.0, 1.0, 1.0, 5.0, 0.1^7]$	$[0.1^4, 1.0^7]$	59.7	48.5	56.9	55.0
$[1.0, 1.0, 1.0, 15.0, 0.1^7]$	$[0.1^4, 1.0^7]$	**68.6**	**58.2**	**64.5**	**63.8**
$[1.0, 1.0, 1.0, 20.0, 0.1^7]$	$[0.1^4, 1.0^7]$	65.8	54.6	61.7	60.7

As shown in Tab. 8, directly applying our action head to UniVLA improves over the original task-specific UniVLA baselines, achieving 53.8% on R2R, 39.1% on RxR, and 43.6% average success rate on SimplerEnv. However, this variant still performs worse than OneVLA, especially compared with the full model after multi-stage training and CoT refinement. OneVLA outperforms this grafted UniVLA variant by 14.8 points on R2R, 19.1 points on RxR, and 20.9 points on SimplerEnv. These results indicate that the unified action head alone is insufficient to achieve strong cross-task performance. Instead, the gains of OneVLA come from jointly coupling the unified action space with shared cross-task representations and the progressive training strategy.

Effect of Action Mask Weights. To validate our strategy of masking weights for different dimensions, we conducted ablation studies on the padding weights and stop action weights. As shown in Tab. 7, two extreme padding configurations ($w_{pad} = 1.0$ and $w_{pad} = 0.0$) both lead to a significant performance drop, with average performance of only 41-43%. This indicates that *simple zero-padding or uniform weighting is insufficient* for cross-task learning: $w_{pad} = 1.0$ introduces irrelevant gradient noise from inactive dimensions, while $w_{pad} = 0.0$ leads to numerical instability. Our optimal configuration with $w_{pad} = 0.1$ achieves an average performance of 63.8%, an improvement of 19.4-25.8% compared to simple padding strategies. For the stop action weight, we observed that $w_{stop} = 1.0$ results in poor navigation performance due to class imbalance (average 46.4%), while $w_{stop} = 20.0$ overemphasizes the stop action, leading to a performance drop to 60.7%. Our chosen $w_{stop} = 15.0$ achieves the best performance of 63.8%. These results confirm that using small non-zero padding weights and carefully tuned task-specific weights is crucial for effective unified action space learning.

Performance Analysis Across Training Stages. As shown in Tab. 9, our multi-stage training strategy demonstrates consistent and significant performance improvements at each stage. In the first stage, the model achieved 46.7% performance in the SimplerEnv environment by focusing on manipulation and general visual question answering (VQA) data, laying a solid foundation for manipulation capabilities. This stage is crucial for learning robust vision-language correspondences and basic end-effector control abilities. In the second stage, the introduction of navigation data for joint training

Table 8: Ablation Study on the Effect of the OneVLA Framework Beyond Unified Action-Head Grafting. UniVLA-Navi.* and UniVLA-Mani.* represent UniVLA's (Bu et al., 2025b) performance using **task-specific fine-tuned action heads** on the same training data for navigation and manipulation respectively. We replace UniVLA's task-specific action heads with our 11-dimensional unified action head and train it directly on the same mixed navigation-manipulation data.

Method	Params	Unified Head	VLN R2R↑	VLN RxR↑	SimplerEnv Avg. SR↑
UniVLA-Navi.* (Bu et al., 2025b)	7B	No	47.1	26.3	–
UniVLA-Mani.* (Bu et al., 2025b)	7B	No	–	–	35.4
UniVLA + Ours Action Head	7B	Yes	53.8	39.1	43.6
OneVLA (Single-Stage Training)	3B	Yes	56.1	42.9	46.2
OneVLA (Ours, Full)	3B	Yes	**68.6**	**58.2**	**64.5**
vs. UniVLA + Ours Action Head	–	–	***(+14.8)***	***(+19.1)***	***(+20.9)***

Table 9: **Performance Across Training Stages.** We report the performance after each training stage. Stage 1 focuses on manipulation only. Stage 2 introduces navigation data for joint training. Stage 3 incorporates CoT data for final refinement.

Training Stage	**VLN R2R↑**	**VLN RxR↑**	**SimplerEnv Avg. SR↑**	**Avg.**
Stage 1 (Manipulation Only)	- -	- -	46.7	- -
Stage 2 (Joint Training)	63.7	53.4	59.1	58.7
Stage 3 (CoT Refinement)	**68.6**	**58.2**	**64.5**	**63.8**
Stage 2 vs. Stage 1	- -	- -	**+12.4**	- -
Stage 3 vs. Stage 2	**+4.9**	**+4.8**	**+5.4**	**+5.1**

Table 10: **Ablation on Training Stage Order.** We compare our proposed order (Manipulation First) with the reverse order (Navigation First) after the first two training stages. Starting with manipulation yields better overall performance, especially for manipulation tasks.

Training Order	**VLN R2R↑**	**VLN RxR↑**	**SimplerEnv↑**	**Avg.**
Stage 1 Performance				
Navigation First	56.8	46.5	- -	- -
Manipulation First (Ours)	- -	- -	**46.7**	- -
Stage 2 Performance (After Adding Second Task)				
Navigation First	58.3	48.2	35.2	47.2
Manipulation First (Ours)	**63.7**	**53.4**	**59.1**	**58.7**
Performance Gap	**+5.4**	**+5.2**	**+23.9**	**+11.5**

brought numerous benefits: navigation performance reached 63.7% on R2R and 53.4% on RxR , while manipulation performance improved to 59.1% (+12.4%), indicating positive transfer effects from cross-task learning. The shared vision-language encoder learned more generalizable spatial and semantic representations, benefiting both tasks. In the third stage, integrating CoT data led to the final performance improvements: R2R performance increased by 4.9% to 68.6%, RxR performance increased by 4.8% to 58.2%, and SimplerEnv performance increased by 5.4% to 64.5%, with an average improvement of 5.1%. CoT data enhanced the model's reasoning capabilities, enabling it to make more thoughtful decisions and better handle complex multi-step instructions. These results validate that our progressive training strategy effectively leverages task-specific learning, cross-task transfer, and enhanced reasoning abilities to achieve state-of-the-art unified performance.

Effect of Training Stage Order. To validate our proposed stage order, we conducted an ablation study comparing two sequences: "manipulation → navigation" and "navigation → manipulation". As shown in Tab. 10, starting with navigation first leads to significant performance degradation after adding the second task in Stage 2: the SimplerEnv success rate is only 35.2%, while our manipulation-first approach achieves 59.1% (+23.9%). Moreover, the navigation-first strategy shows only modest improvement from Stage 1 (56.8%/46.5%) to Stage 2 (58.3%/48.2%) on R2R/RxR, with gains of merely +1.5%/+1.7%. In contrast, our manipulation-first approach demonstrates strong positive transfer, with manipulation performance improving from 46.7% to 59.1% (+12.4%) after adding navigation data. The overall average performance gap of 11.5% (58.7% vs. 47.2%) validates our design choice: establishing fine-grained manipulation capabilities first provides a stronger foundation

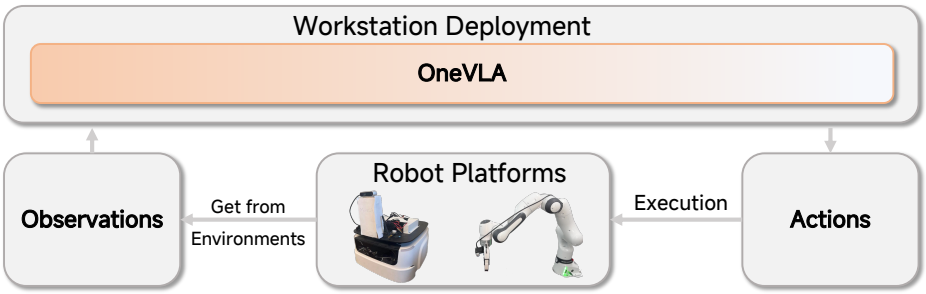

Figure 7: ***OneVLA*** deployment architecture for real-world robot control across navigation and manipulation tasks.

for cross-task learning, while navigation benefits from manipulation pretraining through improved spatial reasoning and vision-language grounding.

D Real-World Experiments

To validate the effectiveness of ***OneVLA*** in real-world scenarios, we conducted extensive experiments on navigation and manipulation tasks using different robotic platforms.

Experimental Setup. We validate ***OneVLA*** on real robotic platforms to assess its real-world performance. For navigation tasks, we deployed the model on a SLAMTEC mobile robot equipped with a RealSense D435i RGB-D camera for egocentric view observation. For manipulation tasks, we used a Franka Research 3 7-DoF robotic arm equipped with two RealSense D435i cameras: one for capturing a third-person view of the workspace and another wrist-mounted camera at the end effector for close-up observation. For navigation tasks, we designed four tasks: two in a conference room (e.g., "walk to the TV") and two in a corridor area (e.g., "navigate to the end of the corridor and turn left"), covering different semantic understanding and action types, including forward movement, turning, and precise stopping. For manipulation tasks, we defined four tasks testing different skills: "put the chili in the plate" (precise grasping and placement), "stack the cups" (contact-rich vertical manipulation), "wipe the whiteboard" (trajectory tracking and tool use), and "open the drawer and put the screwdriver inside" (sequential actions and spatial reasoning). Each task was performed 20 rollouts independently. The success criterion for navigation tasks was reaching the target location with an error of less than 0.5 meters, and the success criterion for manipulation tasks was completing the task objective without collisions. All experiments were conducted in a real office environment or on a tabletop with natural lighting and clutter.

Results Analysis. As shown in Tab. 11 and Tab. 12, ***OneVLA*** achieved significant improvements over all baseline methods, especially when compared to the most competitive cross-task method, UniVLA. In the navigation task, ***OneVLA***'s performance improved by 35.0% (77.5% vs. 42.5%) compared to UniVLA, and by 16.3% (78.8% vs. 62.5%) in the manipulation task. This fully demonstrates the effectiveness of our unified architecture and multi-stage training strategy. ***OneVLA*** demonstrates superior capabilities in complex scenarios: in conference room navigation tasks (Tasks 3-4) requiring long-range planning and multi-step reasoning, ***OneVLA*** achieves success rates of 80.0% and 75.0%, while UniVLA achieved only 45.0% and 40.0%; in manipulation tasks requiring precise control, ***OneVLA*** achieves a success rate of 90.0% in the chili pepper placement task compared to 70.0% for UniVLA, and 80.0% in the cup stacking task requiring delicate contact, compared to 65.0% for UniVLA. These results validate that our proposed unified framework can effectively transfer from simulation environments to real-world applications, enhancing embodied intelligence capabilities beyond other methods.

E More Visualization Results

We present numerous qualitative visualization results in both simulated and real-world environments. For simulated navigation tasks, Fig. 8-Fig. 10 show representative trajectories in the VLN-CE environment, where the agent successfully followed complex natural language instructions involving

Table 11: Real-World Navigation Experiments Results.

Method	Task 1	Task 2	Task 3	Task 4	Avg.
MapNav (Zhang et al., 2025c)	70.0	60.0	65.0	65.0	65.0
NaVid (Zhang et al., 2024)	60.0	55.0	60.0	55.0	57.5
UniVLA (Bu et al., 2025b)	45.0	40.0	45.0	40.0	42.5
OneVLA (Ours)	**80.0**	**75.0**	**80.0**	**75.0**	**77.5**

Table 12: Real-World Manipulation Experiments Results.

Method	Pepper	Stack	Clean	Drawer	Avg.
Diffusion Policy (Chi et al., 2025)	55.0	50.0	45.0	45.0	48.8
OpenVLA (Kim et al., 2025b)	45.0	40.0	30.0	35.0	37.5
UniVLA (Bu et al., 2025b)	70.0	65.0	55.0	60.0	62.5
OneVLA (Ours)	**90.0**	**80.0**	**70.0**	**75.0**	**78.8**

multiple spatial landmarks and directional changes. For simulated manipulation tasks, Fig. 11-Fig. 16 showcase the execution of various skills in the SimplerEnv (Li et al., 2025) and LIBERO (Liu et al., 2023) environments, including pick-and-place operations, stacking, and container insertion, highlighting the model's ability to generate precise 7-DOF end-effector trajectories. In real-world experiments, Fig. 17 shows navigation trials conducted on a SLAMTEC mobile platform navigating through office corridors and meeting rooms, where egocentric RGB-D observations reveal real-world conditions including varying lighting, background clutter, and dynamic elements, demonstrating robust generalization from simulation to reality. Fig. 18 shows manipulation experiments using a Franka Research 3 robotic arm across all four tasks (placing a pepper shaker, stacking cups, wiping a whiteboard, and opening a drawer). The extensive visualization results confirm that our unified framework achieves robust cross-task performance in both controlled simulations and challenging real-world scenarios.

F Limitations and Broader Impacts

This work focuses on building a unified VLA framework for both navigation and manipulation tasks, and on validating the effectiveness of the proposed unified action head and progressive training strategy through simulated benchmarks and real-world robotic experiments. In future work, evaluating OneVLA on more diverse robot platforms and outdoor environments would help provide a more comprehensive understanding of its capabilities. Another promising direction is to explore adaptive action-horizon selection and related temporal abstraction mechanisms, which may further improve the robustness of the system across tasks with different temporal scales. OneVLA may have positive societal impacts by advancing general-purpose robotic systems that can follow natural language instructions across both navigation and manipulation domains, potentially benefiting assistive robotics, service robots, and embodied AI research. When deploying such systems in the real world, attention should also be paid to safety risks.

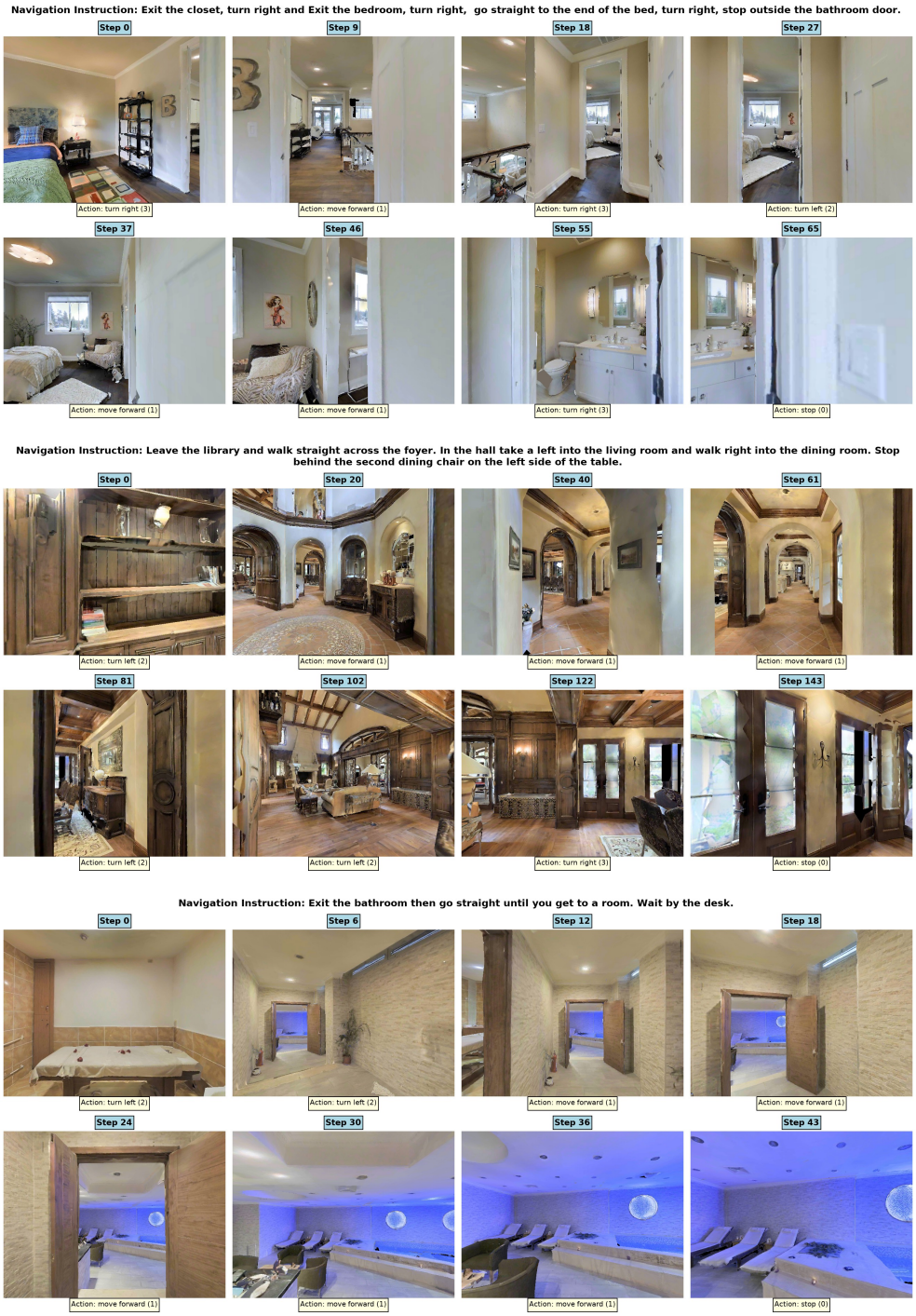

Figure 8: More Visualization Results of Navigation Tasks in Simulator.

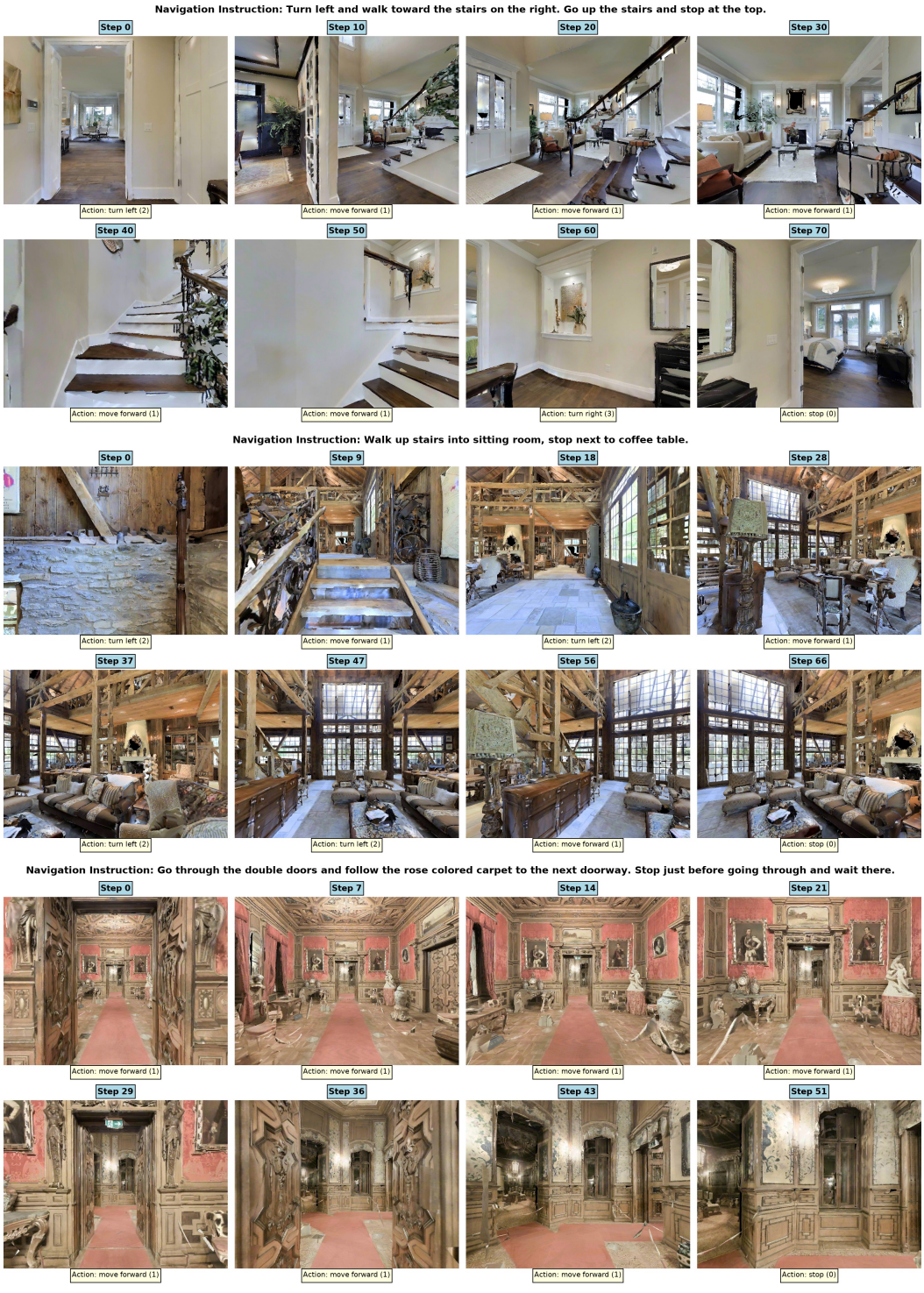

Figure 9: More Visualization Results of Navigation Tasks in Simulator.



Figure 10: More Visualization Results of Navigation Tasks in Simulator.

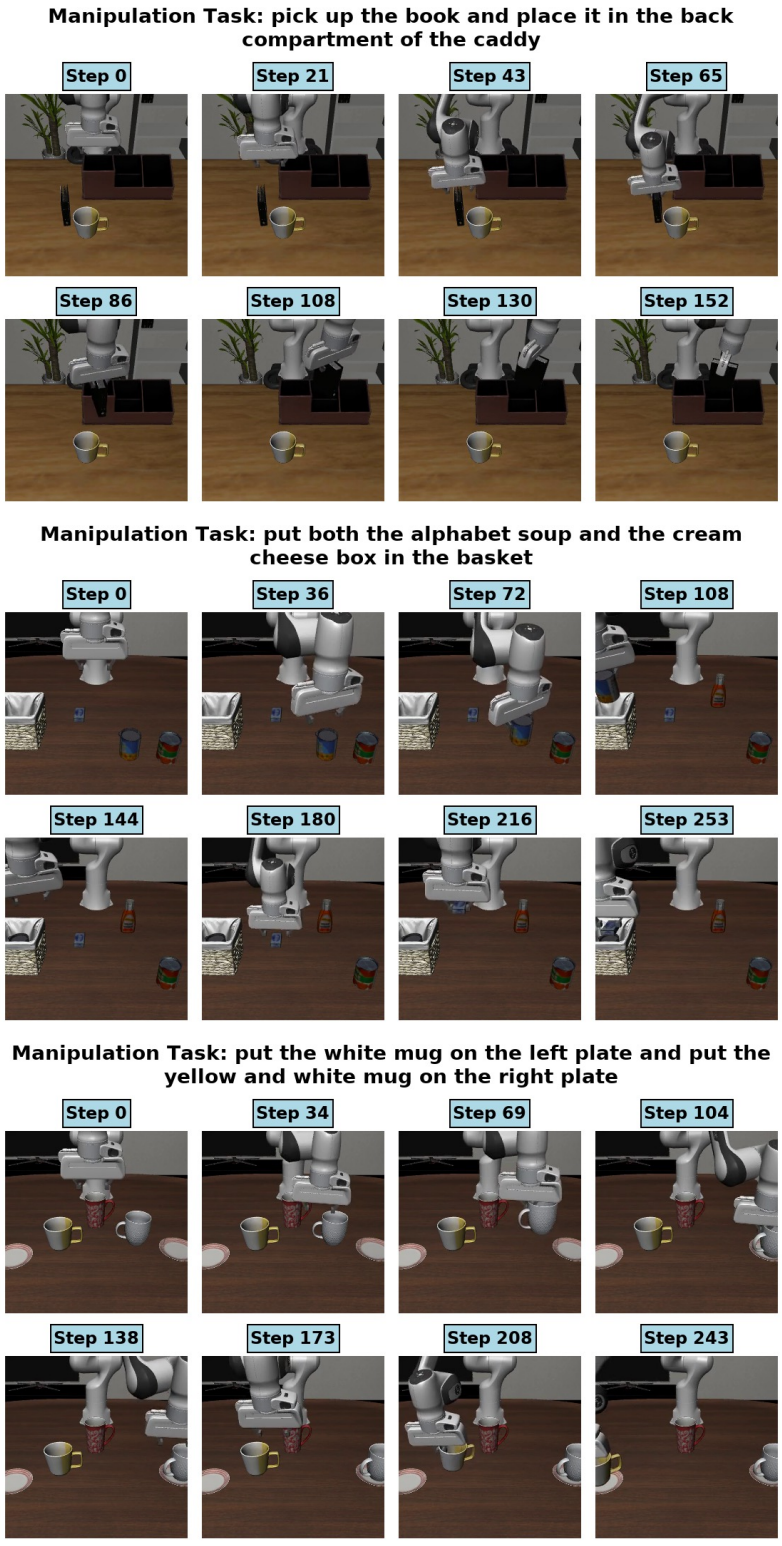

Figure 11: More Visualization Results of LIBERO_10 Tasks in Simulator.

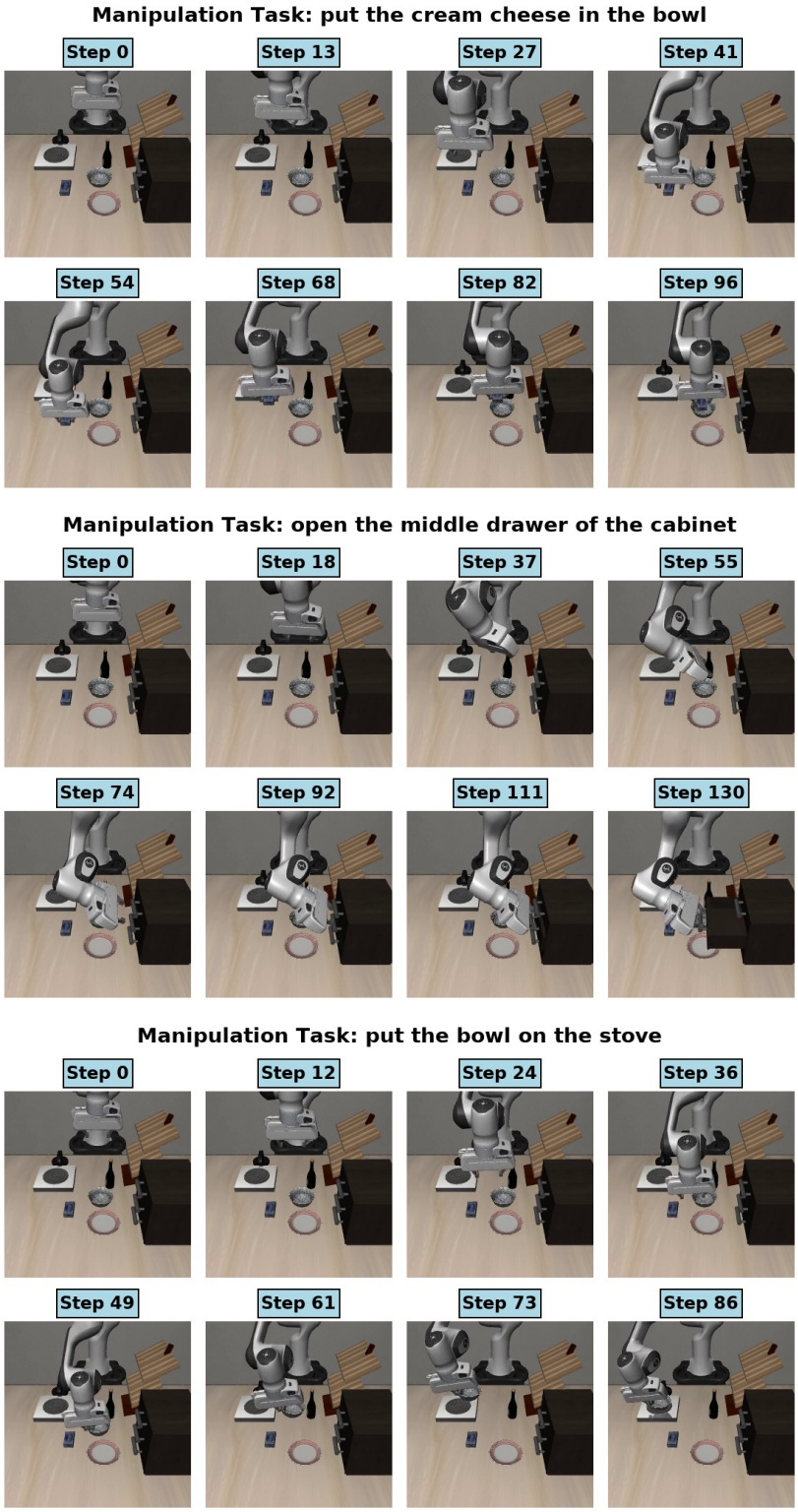

Figure 12: More Visualization Results of LIBERO_goal Tasks in Simulator.

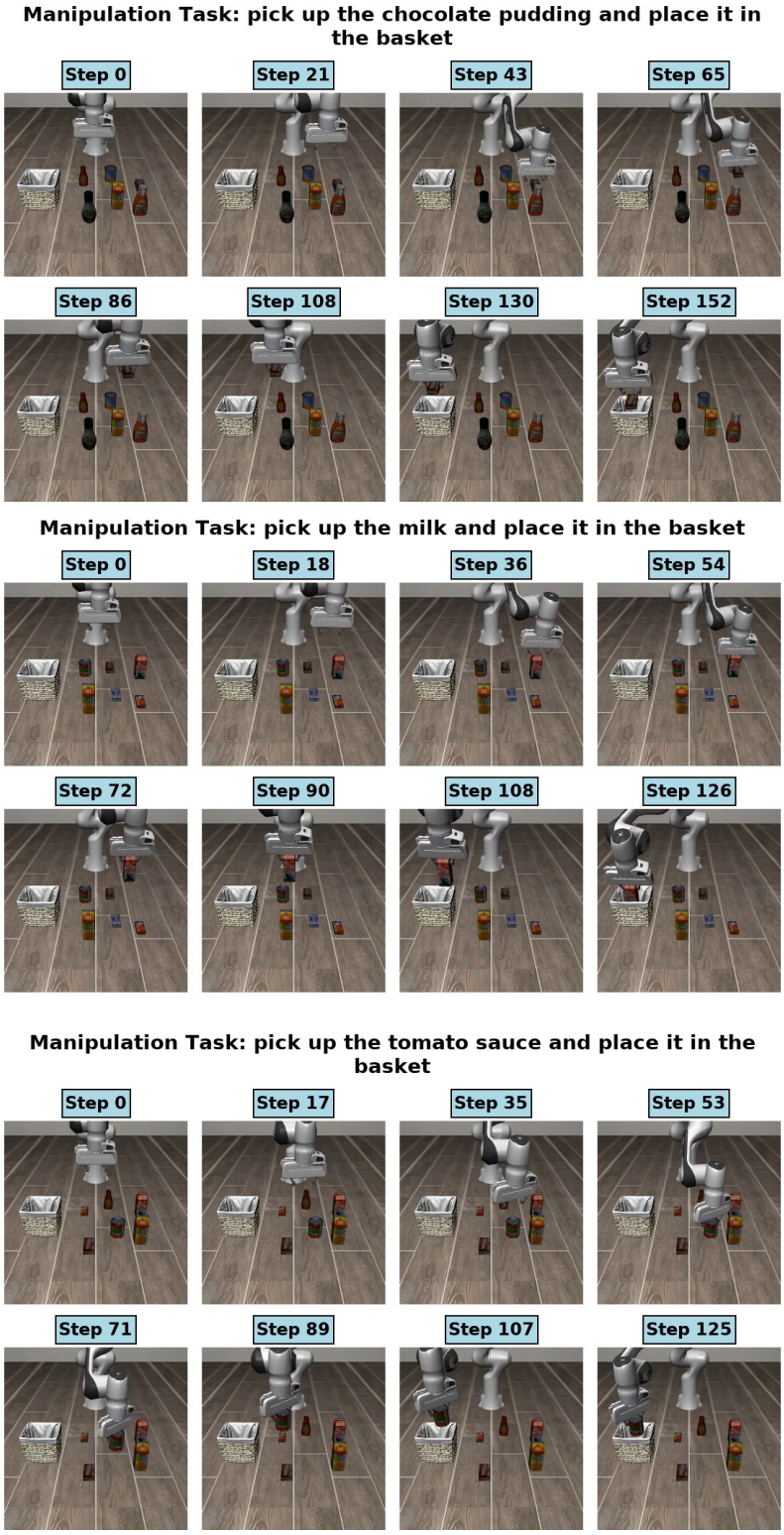

Figure 13: More Visualization Results of LIBERO_object Tasks in Simulator.

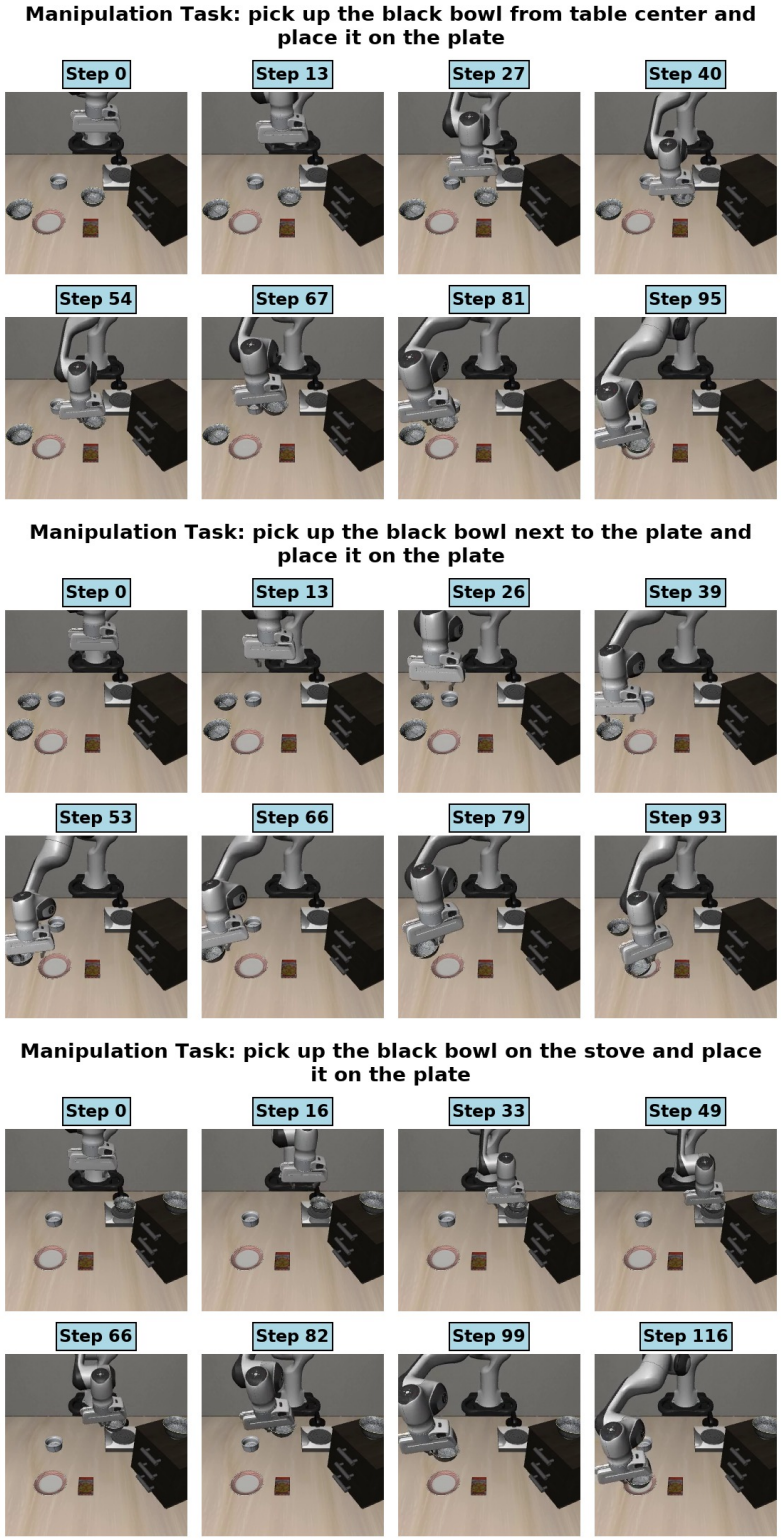

Figure 14: More Visualization Results of LIBERO_spatial Tasks in Simulator.

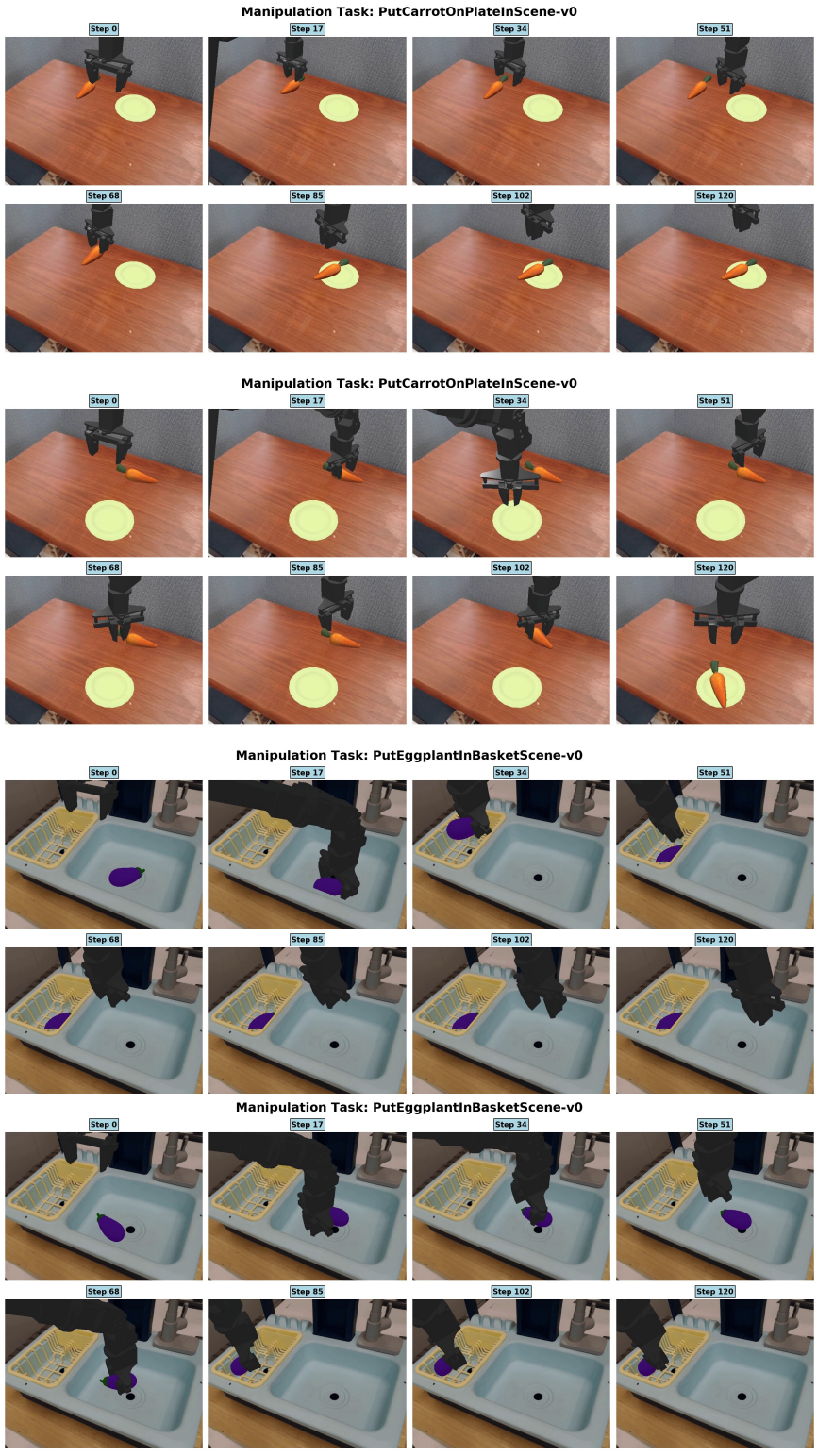

Figure 15: More Visualization Results of SimplerEnv Tasks in Simulator.

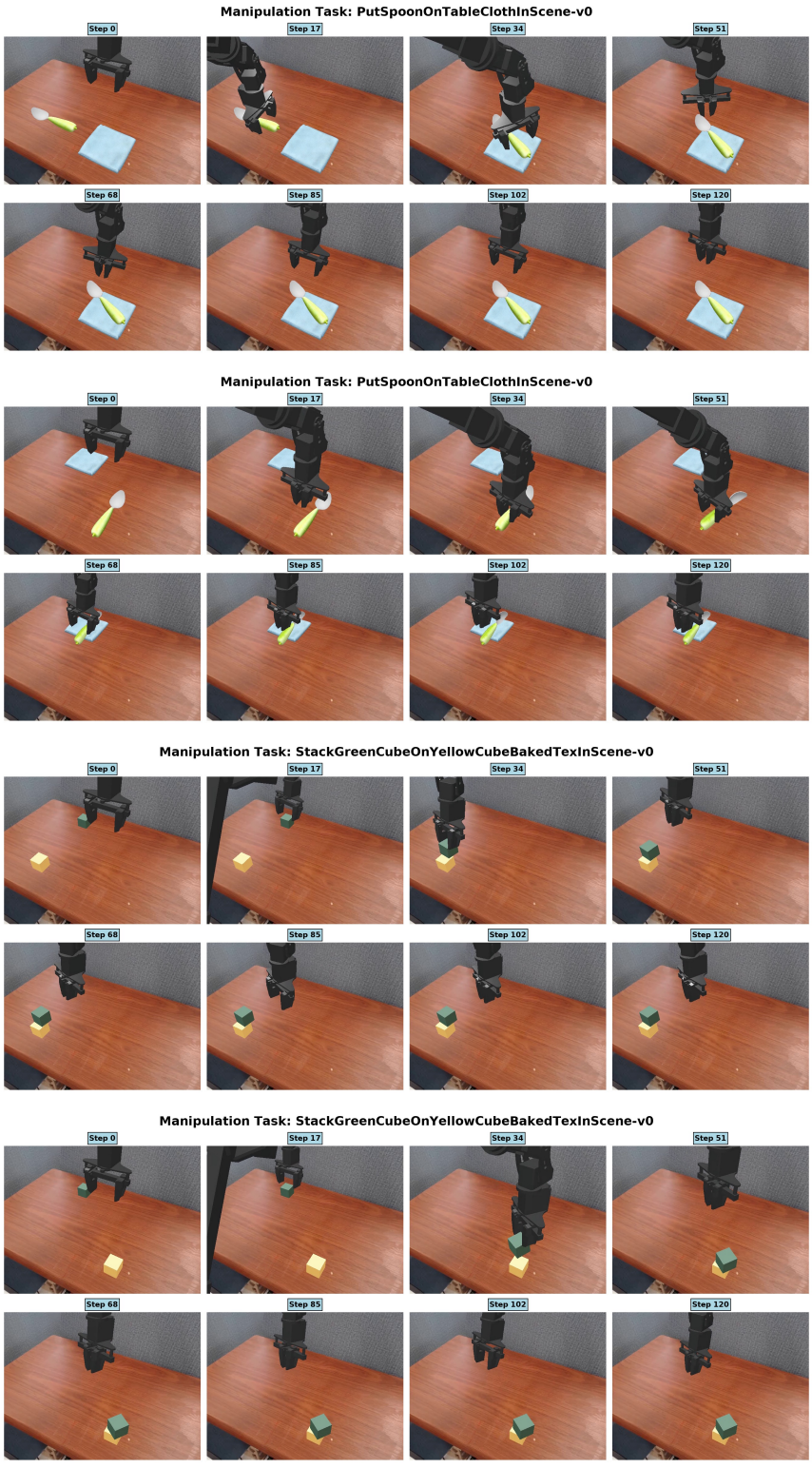

Figure 16: More Visualization Results of SimplerEnv Tasks in Simulator.

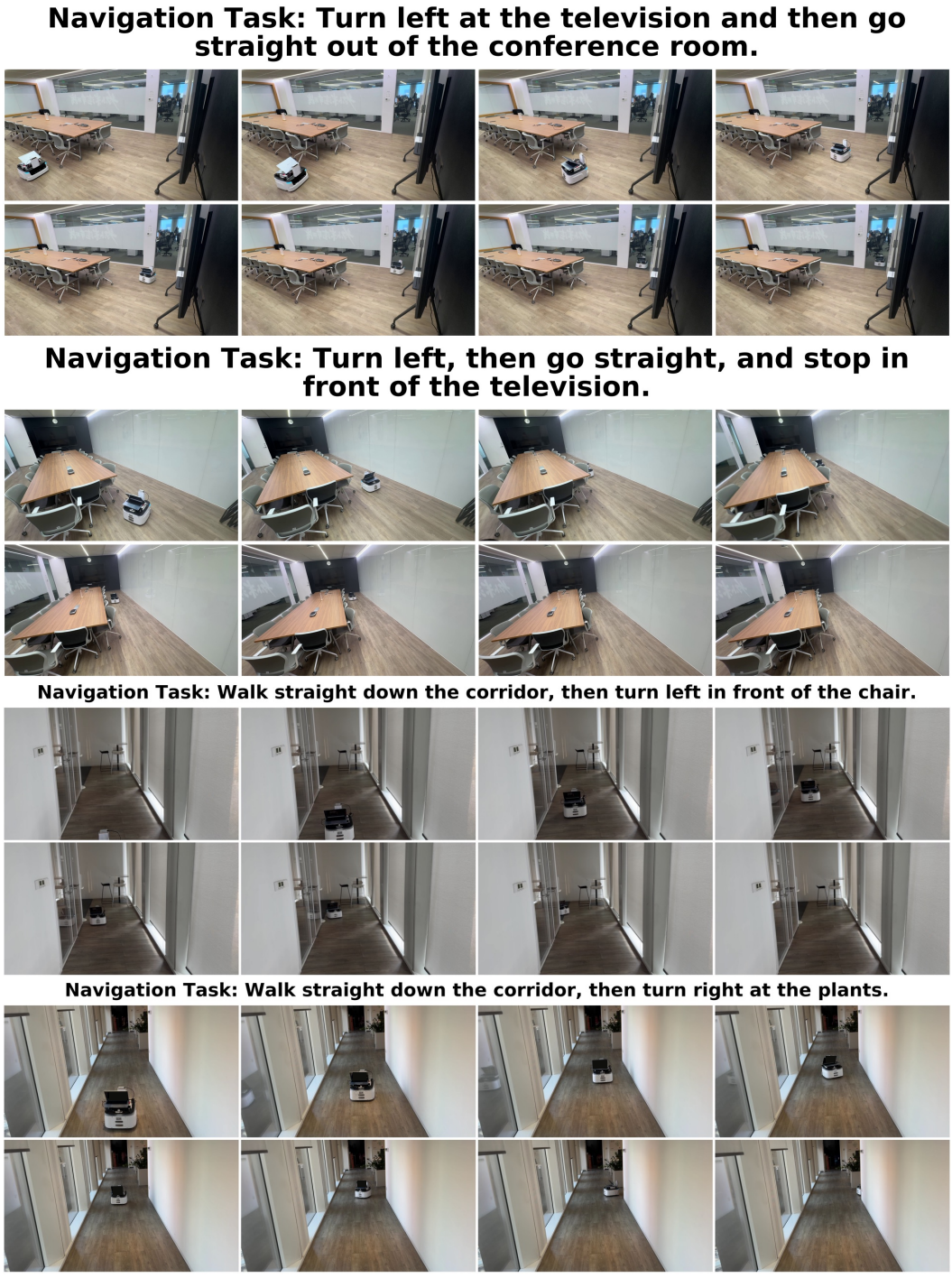

Figure 17: More Visualization Results of Navigation Tasks in Real-World Environments.

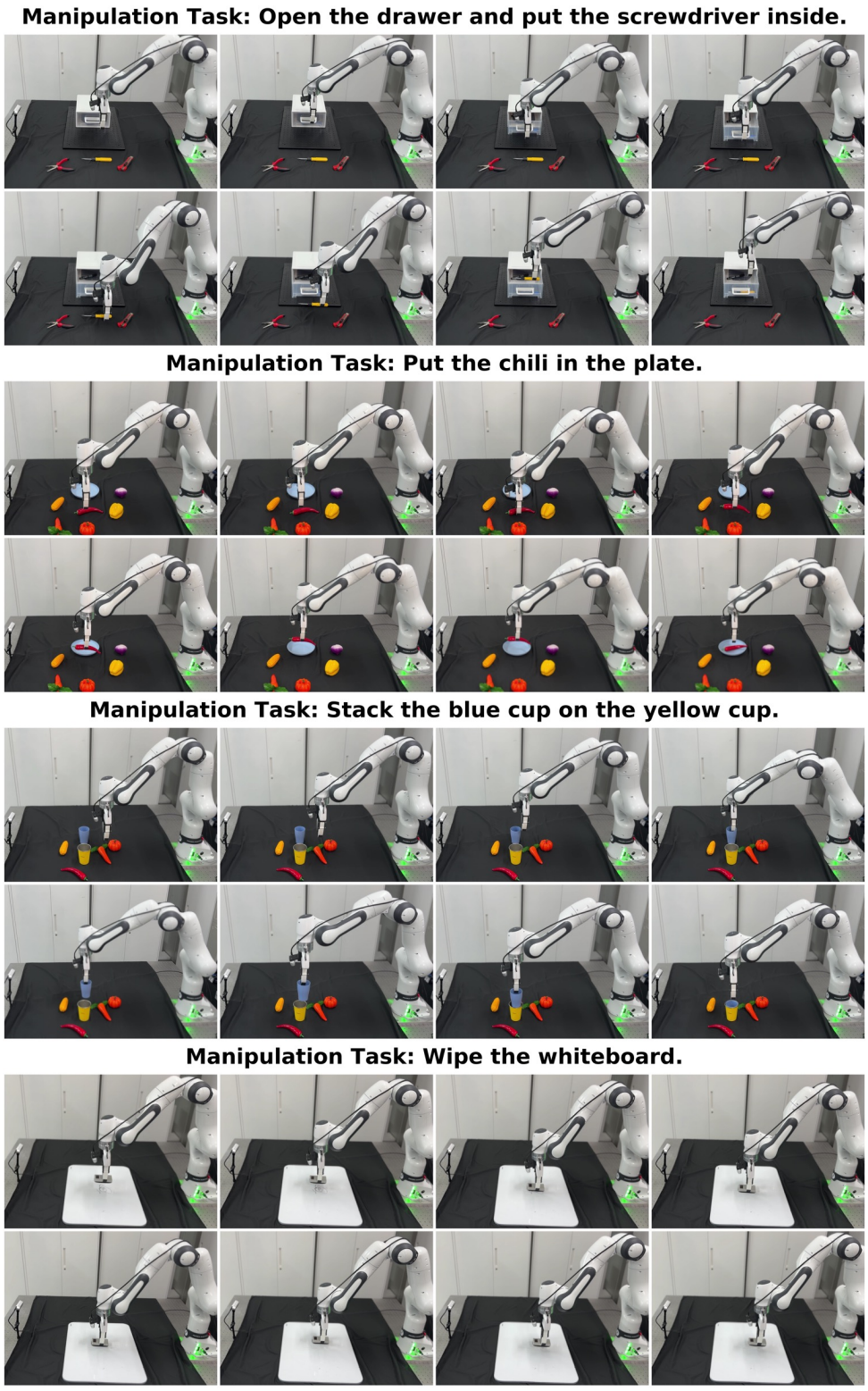

Figure 18: More Visualization Results of Manipulation Tasks in Real-World Environments.