MobileVLA-R1: Reinforcing Vision-Language-Action for Mobile Robots

Ting Huang[1*], Dongjian Li[1*], Rui Yang[1*], Zeyu Zhang[1*†], Zida Yang[1,2], and Hao Tang[1,3‡]

[1]School of Computer Science, Peking University, Beijing, China
[2]Guanghua School of Management, Peking University, Beijing, China
[3]Beijing Academy of Artificial Intelligence, Beijing, China
[*]Equal contribution. [†]Project lead. [‡]Corresponding author: bjdxtanghao@gmail.com.

https://aigeeksgroup.github.io/MobileVLA-R1/

Fig. 1: **MobileVLA-R1** unifies language reasoning and continuous action for real-world quadruped instruction following, shown through target discrimination, obstacle avoidance, and narrow-gap traversal.

Abstract. Grounding natural-language instructions into executable continuous control remains a core challenge for quadruped vision-language-action (VLA) systems due to the gap between high-level semantic reasoning and low-level locomotion actuation. Existing approaches often rely on implicit reasoning or purely behavioral supervision, leading to unstable long-horizon grounding and limited robustness in real-world deployment. To address these issues, we present **MobileVLA-R1**, a unified quadruped VLA framework that explicitly aligns hierarchical reasoning with continuous control. Our method introduces a multi-granularity chain-of-thought (CoT) supervision scheme for embodied trajectories, instantiated in MobileVLA-CoT, together with a two-stage training paradigm that combines supervised CoT alignment and reinforcement learning for reasoning-to-control consistency. We conduct comprehensive evaluations on VLN and quadruped VLA benchmarks, including VLN-CE, QUARD, and real-world Unitree Go2 deployment. MobileVLA-R1 consistently outperforms strong baselines, achieving about 5% gains on key metrics while demonstrating deployment-time closed-loop execution under a fixed hybrid inference setup.

Keywords: Vision-language-action · Quadruped robots · Embodied navigation · Reinforcement learning

1 Introduction

Vision-language-action (VLA) systems for mobile robots aim to translate natural-language instructions into grounded perception, reasoning, and executable control. For quadruped robots, this problem is particularly challenging because language-guided decisions must ultimately be realized as continuous locomotion commands under partial observability, actuation noise, and long-horizon task dependencies. Robustly bridging this gap is central to embodied intelligence in real-world environments [3, 4, 6, 15, 28, 53, 54, 71].

Recent progress in multimodal foundation models [51, 66] has substantially improved visual perception and language understanding, inspiring a growing body of work on extending vision-language models to embodied decision making. However, existing approaches still face two practical limitations in quadruped VLA settings. First, direct language-to-action mappings often blur the boundary between high-level semantic reasoning and low-level motor execution, resulting in weak interpretability and unstable grounding over long horizons. Second, methods that improve stability through latent intermediate representations typically sacrifice a transparent reasoning structure, making it difficult to inspect semantic logic, diagnose failure cases, or support compositional decision making.

These limitations motivate a more explicit connection between *reasoning* and *control* in embodied agents. In particular, we seek a formulation that preserves interpretable high-level planning while remaining compatible with continuous quadruped execution in complex environments.

To this end, we present **MobileVLA-R1**, a hierarchical quadruped VLA framework that explicitly aligns structured reasoning with continuous control. Instead of predicting motor commands directly from language and observations, MobileVLA-R1 first generates a structured Chain-of-Thought (CoT) action plan conditioned on multimodal observations and then translates the plan into executable continuous control commands via an action decoder. This reasoning-then-execution design improves interpretability, stabilizes language grounding, and supports more reliable long-horizon behavior. To train this capability, we construct MobileVLA-CoT, a large-scale embodied trajectory dataset with multi-granularity CoT supervision, and adopt a two-stage training paradigm. We first perform supervised CoT alignment to establish structured reasoning ability and then apply reinforcement learning with Group Relative Policy Optimization (GRPO) to refine action grounding and execution fidelity. This combination improves reasoning-to-command consistency while preserving deployment-time executability.

We evaluate MobileVLA-R1 on complementary embodied benchmarks, including R2R-CE and RxR-CE for language-guided navigation, the QUARD benchmark for quadruped control and manipulation, and real-world deployment on a Unitree Go2 robot. Across these settings, MobileVLA-R1 consistently outperforms strong baselines, achieving approximately 5% absolute gains on key metrics and demonstrating deployment-time closed-loop execution in cluttered and partially observable environments.

In summary, our contributions are three-fold:

- We propose **MobileVLA-R1**, a hierarchical quadruped VLA framework that explicitly connects structured Chain-of-Thought reasoning with continuous motor control through a reasoning-then-execution design.
- We introduce a two-stage training paradigm that combines supervised CoT alignment and offline GRPO refinement to improve reasoning-to-command consistency and executable output validity.
- We construct MobileVLA-CoT, a multi-granularity CoT dataset for embodied trajectories, and demonstrate consistent gains across VLN, quadruped VLA benchmarks, and real-world Unitree Go2 deployment, including approximately 5% improvements on key metrics over strong baselines.

2 Related Work

Language-guided navigation and quadruped VLA. Vision-and-language navigation (VLN) studies instruction following in 3D environments, with R2R [3] and RxR [33] serving as standard benchmarks. With pretrained VLMs [25–27, 48], VLN methods have evolved from sequence-to-sequence learning [16, 37] to attention- and transformer-based architectures [10, 20, 69], and further to pretrained VLM systems for improved grounding and generalization [19, 22, 35, 40, 41, 43, 62, 64, 70]. In parallel, quadruped robot learning has adopted language-conditioned policies and multimodal perception for locomotion and manipulation, including QUART-style models and their online variants [14, 50], as well as generalist quadruped frameworks such as GeRM [47]. However, most existing quadruped VLA approaches emphasize direct policy generation or system-level integration, with limited explicit modeling of structured reasoning for long-horizon continuous control. Our work focuses on this reasoning-control interface by coupling CoT-style planning with executable continuous quadruped actions.

Vision-language-action and embodied foundation models. Recent embodied foundation models show that large-scale multimodal pretraining can support language-conditioned action generation across navigation and manipulation. Representative systems, including SayCan and SayTap [4, 49], PaLM-E [15], RT-2 [71], OpenVLA [30], and Octo [17], unify perception, language, and control within shared policy frameworks. While these methods improve transfer and robustness, many rely on implicit latent decision making or manipulation-centric settings; in contrast, MobileVLA-R1 targets quadruped navigation and control, where continuous locomotion requires an explicit bridge between semantic reasoning and motor execution.

Reinforcement learning for embodied decision making. Chain-of-thought (CoT) prompting and supervision improve multi-step reasoning in language models [57], while reinforcement learning from feedback strengthens instruction following and reasoning quality [39]. Recent reasoning-oriented models use policy optimization, such as PPO and GRPO-style training, to improve long-horizon reasoning consistency [46]. CoT-style supervision and reinforcement learning have also extended to multimodal and embodied settings [11, 59, 68]. Unlike prior work that mainly uses CoT for interpretability or high-level planning,

Table 1: Statistics of source datasets and the synthesized MobileVLA-CoT. "Nav." indicates navigation supervision, "Emb-Ctl." indicates embodied continuous control supervision, and "CoT" indicates synthesized reasoning traces. "Samples" denotes the number of instances reported by the original datasets.

Dataset	Nav.	Emb-Ctl.	CoT	Samples
R2R [3]	✓	✗	✗	50K
RxR [33]	✓	✗	✗	58K
QUARD [14]	✗	✓	✗	262K
MobileVLA-CoT-Episode	✗	✓	✓	18K
MobileVLA-CoT-Step	✗	✓	✓	78K
MobileVLA-CoT-Nav	✓	✗	✓	38K

MobileVLA-R1 uses multi-granularity CoT as structured supervision and combines SFT with offline GRPO to improve executable reasoning-to-command consistency for quadruped control.

3 Datasets

3.1 Source Datasets

R2R [3] provides instruction-trajectory pairs in Matterport3D [5] indoor scenes and is a canonical benchmark for language-guided navigation. RxR [33] extends this setting with multilingual instructions and longer, semantically richer descriptions, offering denser supervision for long-horizon grounding. To incorporate continuous control signals on legged platforms, we additionally use QUARD [14], which contains quadruped locomotion and manipulation episodes with paired observations and action targets. Together, these sources provide complementary supervision for (i) language-to-trajectory grounding and (ii) low-level embodied control execution, forming the basis for synthesizing MobileVLA-CoT.

3.2 Synthesized Dataset

Building on R2R, RxR, and QUARD, we synthesize MobileVLA-CoT, a large-scale multi-granularity dataset that pairs multimodal observations with structured reasoning supervision and executable action targets. Unlike instruction-only datasets that specify goals without intermediate decision traces, MobileVLA-CoT provides explicit reasoning steps that connect language instructions to embodied actions, enabling interpretable supervision for reasoning-aligned control.
Granularity. MobileVLA-CoT contains three complementary subsets: (i) *Episode-level* summarizes trajectory outcomes and high-level strategies; (ii) *Step-level* specifies the next action together with its rationale under the current observation and history; (iii) *Navigation-level* provides long-horizon reasoning that links global instructions to multi-step decisions along the trajectory. In total, MobileVLA-CoT includes 18K episode-level samples, 78K step-level samples, and 38K navigation-level samples, as summarized in Tab. 1.

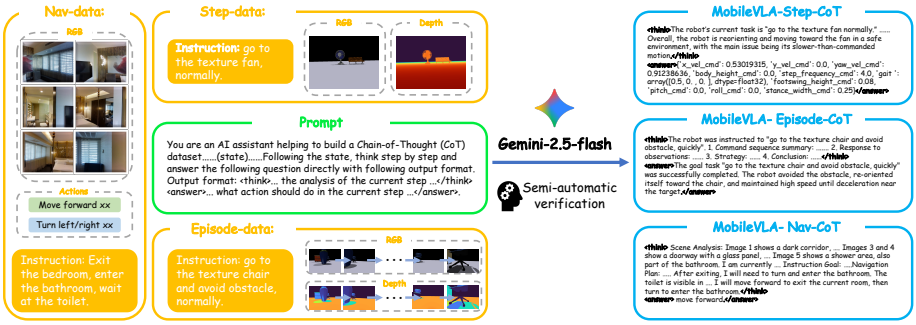

Fig. 2: CoT Data Engine. MobileVLA-CoT is synthesized from multimodal observations, instructions, and state-action histories using structured prompts, producing episode-, step-, and navigation-level reasoning traces with executable targets after automatic parsing and verification.

Data schema. Each sample is stored in a structured format consisting of (a) multimodal observation(s) (e.g., RGB, depth, and task-specific state histories), (b) a natural-language instruction, (c) an optional action history, (d) a reasoning trace (stored with explicit tags such as `<think>`), and (e) an executable action target in `<answer>` format (a discrete action for navigation or a continuous control command, e.g., linear velocity and yaw rate, for quadruped execution).

3.3 CoT Data Engine

As shown in Fig. 2, we design a *model-agnostic* CoT data engine to synthesize multi-granularity reasoning traces for navigation and embodied control. The engine takes multimodal observations (e.g., RGB, depth, and task-specific states), natural-language instructions, and optional state-action histories as inputs, and uses structured prompt templates to request a reasoning trace followed by an executable output. In our implementation, we instantiate the engine with Gemini-2.5-Flash [13]; however, the templates, parsing rules, and verification pipeline are backend-independent and can be applied to alternative multimodal models.
Quality control. We apply a four-stage semi-automatic verification pipeline covering format, action consistency, safety, and manual checks. From 168K raw generations, the pipeline removes malformed, non-executable, unsafe, hallucinated, command-mismatched, or visually inconsistent samples, yielding 134K final samples: 18K episode-level, 78K step-level, and 38K navigation-level. Detailed statistics are reported in *Supp. Mat.*

4 The Proposed Method

4.1 Overview

MobileVLA-R1 follows a hierarchical *reasoning-execution* paradigm and is trained in two sequential stages. Given MobileVLA-CoT (section 3), we first perform supervised fine-tuning (SFT) on CoT-annotated embodied data to establish a

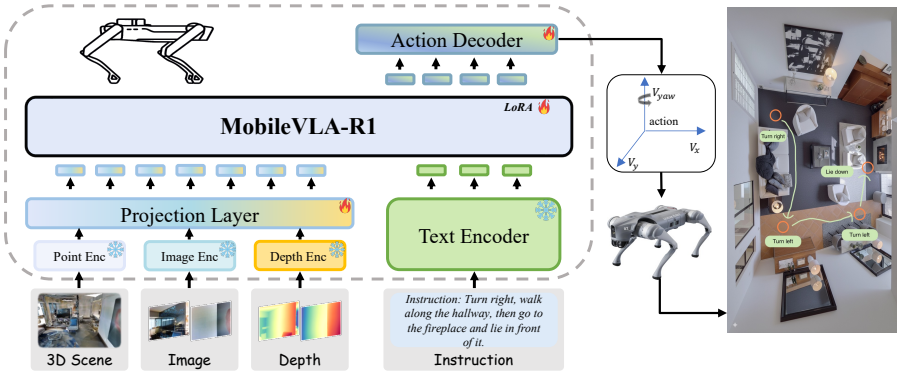

Fig. 3: Overview of MobileVLA-R1. Given RGB, depth, point-cloud observations, and a language instruction, MobileVLA-R1 fuses multimodal tokens with text tokens, generates reasoning-conditioned actions, and decodes them into continuous quadruped control commands for closed-loop execution.

structured reasoning format and multimodal instruction grounding. We then refine executable action generation via offline Group Relative Policy Optimization (GRPO) [46] using parsed command-level rewards. GRPO is not used for online adaptation; it improves offline reasoning-to-command consistency, which is evaluated under deployment-time closed-loop execution (see Fig. 4).

Architecture. As illustrated in Fig. 3, MobileVLA-R1 follows a LLaVA-style multimodal backbone [34]. The model encodes RGB images, depth maps, and point-cloud observations with modality-specific encoders, projects them into a shared token space, and concatenates them with text tokens as input to a NaVILA-initialized VLA backbone [12]. The backbone generates a reasoning trace and an executable answer, which are deterministically parsed and decoded into continuous locomotion commands and discrete high-level behaviors.

Problem setup. We consider closed-loop quadruped control conditioned on multimodal perception and an instruction. At timestep t, the agent receives an observation $s_t = \{x_t^{\text{rgb}}, x_t^{\text{depth}}, x_t^{\text{pc}}\}$ and an instruction $i \in \mathcal{I}$, and outputs an action a_t according to a policy $\pi_\theta(a_t \mid s_t, i)$. We unify continuous locomotion and discrete behaviors by representing

$$a_t = \left[\, \mathbf{v}_t,\ \omega_t,\ \alpha_t \,\right], \quad \mathbf{v}_t = (V_x, V_y), \tag{1}$$

where $\mathbf{v}_t$ denotes planar translational velocity, ω_t is the yaw angular velocity, and α_t is a discrete high-level behavior from a predefined action set (e.g., posture or skill switching). The model outputs are serialized in a structured template:

$$o_t = \texttt{<think>}\ r_t\ \texttt{</think><answer>}\ u_t\ \texttt{</answer>}, \tag{2}$$

where r_t is a reasoning trace and u_t encodes the executable command, enabling machine-parsable execution and interpretable decision traces.

Multimodal tokenization and fusion. To ensure reproducible 3D integration, we explicitly define how each modality is tokenized and fused. Let $E_{\text{rgb}}, E_{\text{depth}}, E_{\text{pc}}$ denote the RGB, depth, and point-cloud encoders, respectively. At timestep t,

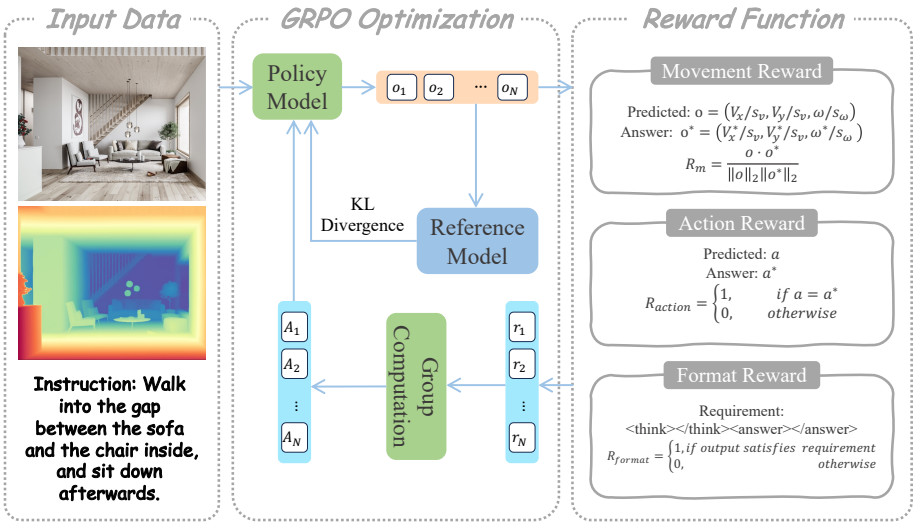

Fig. 4: GRPO training for reasoning-to-control alignment. For each input, the policy samples multiple structured outputs, parses them into executable actions, scores them with movement, action, and format rewards, and updates the policy using group-relative advantages with KL regularization to a frozen reference model.

we obtain modality token sequences:

$$Z_t^m = E_m(x_t^m) \in \mathbb{R}^{N_m \times d_m}, \quad m \in \{\text{rgb}, \text{depth}, \text{pc}\}. \tag{3}$$

We then project each modality into the language-model hidden dimension d using learnable linear adapters $P_m : \mathbb{R}^{d_m} \to \mathbb{R}^d$:

$$\tilde{Z}_t^m = P_m(Z_t^m) \in \mathbb{R}^{N_m \times d}. \tag{4}$$

The final multimodal input token sequence is formed by concatenation with text tokens Z^{text}:

$$Z_t = [\tilde{Z}_t^{\text{rgb}}; \tilde{Z}_t^{\text{depth}}; \tilde{Z}_t^{\text{pc}}; Z^{\text{text}}], \tag{5}$$

augmented with modality-type embeddings to distinguish modalities. This simple but effective fusion is compatible with NaVILA-style backbones and allows controlling the token budget via N_m. Implementation details are reported in section 5.

4.2 Cold Start Stage

Direct end-to-end reinforcement learning in multimodal embodied settings is often unstable due to sparse feedback, format drift, and action-execution constraints. We therefore introduce a *cold-start* stage based on supervised fine-tuning (SFT) to (i) stabilize the output schema and (ii) initialize structured reasoning prior to RL.

Long-horizon reasoning alignment. We first fine-tune MobileVLA-R1 on MobileVLA-CoT-Episode and MobileVLA-CoT-Nav to learn trajectory-level and

navigation-level reasoning in the structured format. This stage encourages coherent long-horizon decision traces and consistent instruction grounding.

Executable control alignment. We then continue SFT on MobileVLA-CoT-Step whose `<answer>` fields contain executable continuous commands and discrete behaviors. The answer is deterministically parsed into $a_t = [\mathbf{v}_t, \omega_t, \alpha_t]$ for closed-loop execution, providing a smooth transition from language-guided reasoning to motor actuation. This stage primarily improves format faithfulness and action executability before reinforcement learning.

4.3 Reinforcement Learning with GRPO

To further improve reasoning-driven action generation, we employ GRPO [46], which optimizes policies using *group-relative* feedback over multiple sampled responses. Compared to standard PPO-style optimization [45], GRPO leverages within-group normalization to reduce sensitivity to absolute reward scales and stabilize training for structured outputs.

Policy sampling. Given an input (s_t, i), the policy π_θ generates N candidate outputs $\{o_1, o_2, \ldots, o_N\}$. Each output o_j is parsed into an action $\hat{a}_j = [\hat{\mathbf{v}}_j, \hat{\omega}_j, \hat{\alpha}_j]$ and evaluated by reward functions that encourage accurate, executable, and well-formatted predictions.

Movement reward. This reward encourages directional consistency between predicted and target continuous commands. Because translational velocity and yaw rate have different units, we compute cosine similarity in a normalized command space:

$$\hat{\mathbf{u}}_j = (\hat{V}_{x,j}/s_v, \hat{V}_{y,j}/s_v, \hat{\omega}_j/s_\omega), \quad \mathbf{u}^* = (V_x^*/s_v, V_y^*/s_v, \omega^*/s_\omega), \tag{6}$$

where s_v and s_ω normalize velocity and yaw rate. The movement reward is

$$R_m(o_j) = \frac{\hat{\mathbf{u}}_j^\top \mathbf{u}^*}{\|\hat{\mathbf{u}}_j\|_2 \, \|\mathbf{u}^*\|_2}. \tag{7}$$

Action reward. To supervise discrete behaviors, we use a binary match reward:

$$R_{act}(o_j) = \mathbb{I}\left[\hat{\alpha}_j = \alpha^*\right]. \tag{8}$$

Format reward. To ensure machine-parseable structured outputs, we reward strict adherence to the reasoning-execution template:

$$R_{fmt}(o_j) = \mathbb{I}\left[o_j \text{ satisfies } \texttt{<think>...</think><answer>...</answer>}\right]. \tag{9}$$

Total reward. The final scalar reward is a weighted sum:

$$r_j = \lambda_m R_m(o_j) + \lambda_a R_{act}(o_j) + \lambda_f R_{fmt}(o_j), \tag{10}$$

where $\lambda_m, \lambda_a, \lambda_f$ are fixed coefficients.

Table 2: Comparison on VLN-CE Val-Unseen [32,33]. We report standard metrics on R2R-CE and RxR-CE val-unseen splits. "Pano." and "Odo." denote panoramic observations and odometry; * indicates a simulator-pretrained waypoint predictor [32].

Method	Observation				R2R-CE Val-Unseen				RxR-CE Val-Unseen			
	S.RGB	Pano.	Depth	Odom.	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
CMA* [21]	✗	✓	✓	✓	6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0
Sim2Sim* [31]	✗	✓	✓	✓	6.07	52.0	43.0	36.0	-	-	-	-
GridMM* [56]	✗	✓	✓	✓	5.11	61.0	49.0	41.0	-	-	-	-
Ego^2-Map* [23]	✗	✓	✓	✓	5.54	56.0	47.0	41.0	-	-	-	-
DreamWalker* [52]	✗	✓	✓	✓	5.53	59.0	49.0	44.0	-	-	-	-
Reborn* [2]	✗	✓	✓	✓	5.40	57.0	50.0	46.0	5.98	48.6	42.0	63.3
ETPNav* [1]	✗	✓	✓	✓	4.71	65.0	57.0	49.0	5.64	54.7	44.8	61.9
HNR* [55]	✗	✓	✓	✓	4.42	67.0	61.0	51.0	5.50	56.3	46.7	63.5
AG-CMTP [8]	✗	✓	✓	✓	7.90	39.0	23.0	19.0	-	-	-	-
R2R-CMTP [8]	✗	✓	✓	✓	7.90	38.0	26.0	22.0	-	-	-	-
InstructNav [36]	✗	✓	✓	✓	6.89	-	31.0	24.0	-	-	-	-
LAW [44]	✓	✗	✓	✓	6.83	44.0	35.0	31.0	10.90	8.0	8.0	38.0
CM2 [18]	✓	✗	✓	✓	7.02	41.0	34.0	27.0	-	-	-	-
WS-MGMap [9]	✓	✗	✓	✓	6.28	47.0	38.0	34.0	-	-	-	-
AO-Planner [7]	✗	✓	✓	✗	5.55	59.0	47.0	33.0	7.06	43.3	30.5	50.1
Seq2Seq [32]	✓	✗	✓	✗	7.77	37.0	25.0	22.0	12.10	13.9	11.9	30.8
CMA [32]	✓	✗	✓	✗	7.37	40.0	32.0	30.0	-	-	-	-
NaVid [65]	✓	✗	✗	✗	5.47	49.0	37.0	35.0	-	-	-	-
Uni-NaVid [64]	✓	✗	✗	✗	5.58	53.5	47.0	42.7	6.24	48.7	40.9	-
NaVILA [12]	✓	✗	✗	✗	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8
VLN-R1 [41]	✓	✗	✗	✗	7.00	41.2	30.2	21.8	9.10	22.7	17.6	-
OctoNav [17]	✓	✗	✗	✗	-	42.9	37.1	33.6	-	-	-	-
StreamVLN [58]	✓	✗	✗	✗	4.98	64.2	56.9	51.9	6.22	52.9	46.0	61.9
CorrectNav [63]	✓	✗	✗	✗	4.24	67.5	65.1	62.3	4.09	69.3	63.3	75.2
MobileVLA-R1 (Ours)	✓	✗	✓	✗	**4.05**	**69.7**	**68.3**	**65.2**	**3.92**	**71.5**	**66.8**	**76.1**

GRPO objective. For each group of N samples, we compute group-relative advantages

$$A_j = r_j - \frac{1}{N}\sum_{k=1}^{N} r_k, \quad \hat{A}_j = \frac{A_j}{\sigma_A + \epsilon}, \tag{11}$$

where σ_A is the standard deviation within the group. Let $\pi_{\theta_{\text{old}}}$ denote the policy before the update and π_{ref} a frozen reference policy. The clipped GRPO objective is:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{(s_t,i)\sim\mathcal{D}}\,\mathbb{E}_{j=1}^{N}\left[\min\left(\rho_j(\theta)\hat{A}_j,\ \text{clip}(\rho_j(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_j\right) - \beta\, D_{\text{KL}}(\pi_\theta(\cdot \mid s_t, i)\,\|\,\pi_{\text{ref}}(\cdot \mid s_t, i))\right], \tag{12}$$

where $\rho_j(\theta) = \frac{\pi_\theta(o_j|s_t,i)}{\pi_{\theta_{\text{old}}}(o_j|s_t,i)}$. The KL regularizer prevents over-updating and stabilizes training for structured reasoning outputs.

5 Experiments

Benchmarks. We evaluate MobileVLA-R1 in two complementary settings to measure both language-guided navigation and executable quadruped control. We benchmark language-guided navigation on VLN-CE [32, 33], which extends the R2R and RxR instruction-following setting to continuous control in photorealistic Matterport3D environments [5]. VLN-CE requires agents to interact with the environment in a closed loop and follow long-horizon instructions under

Table 3: Overall performance on QUARD [14]. We report success rates on six tasks grouped by difficulty; Average is the mean over all tasks, with 25 episodes per task.

Method	Easy	Medium	Hard				Average
	Distinguish	Go-to	Go-avoid	Go-through	Crawl	Unload	
CLIP [42]	0.44	0.43	0.45	0.19	0.00	0.00	0.25
VC-1 [38]	0.46	0.43	0.45	0.31	0.00	0.00	0.28
QUART [14]	0.66	0.60	0.53	0.41	0.32	0.12	0.44
MoRE [67]	0.82	0.80	0.59	0.57	0.49	0.33	0.60
MobileVLA-R1	**0.92**	**0.89**	**0.71**	**0.65**	**0.58**	**0.44**	**0.73**

realistic egocentric observations. Following standard protocols, we report results on the *val-unseen* splits of R2R-CE and RxR-CE. To assess low-level actuation and reasoning-aligned action generation on legged platforms, we conduct action-only evaluations on QUARD [14], which contains diverse quadruped locomotion and manipulation tasks with paired observations and control targets. Compared with VLN-CE, QUARD emphasizes executable continuous control and discrete behaviors under task-specific dynamics and interaction constraints.

Evaluation metrics. For VLN-CE, we use standard navigation metrics: navigation error (NE), oracle success rate (OS), success rate (SR), success-weighted path length (SPL), and normalized dynamic time warping (nDTW). These metrics jointly measure goal-reaching accuracy, path efficiency, and trajectory fidelity. For QUARD, we follow the official evaluation protocol [14] and report the average success rate across six quadruped tasks. Each value is computed over 25 evaluation episodes per task, and tasks are grouped into three difficulty levels (Easy/Medium/Hard) according to motion and interaction complexity.

Network architecture. We initialize MobileVLA-R1 from NaVILA [12] to leverage its pretrained embodied vision-language alignment. Since NaVILA is primarily trained with RGB observations, we extend it to support depth and 3D geometry by adding modality-specific encoders and a unified token projection interface. Specifically, we use DepthAnything V2 [61] as the depth encoder and Point Transformer v3 [60] as the point-cloud encoder. For each modality $m \in \{\text{rgb}, \text{depth}, \text{pc}\}$, the encoder produces a token sequence $Z^m \in \mathbb{R}^{N_m \times d_m}$, which is mapped to the LLM hidden dimension d via a learnable linear projector P_m. The projected tokens are concatenated with text tokens to form the multimodal input to the LLaMA3-8B backbone, optionally augmented with learned modality-type embeddings. This design provides a reproducible and token-budget-controlled integration of RGB-depth-point inputs while preserving NaVILA-style multimodal alignment.

Parameter-efficient tuning. We adopt LoRA [24] for efficient adaptation. LoRA modules are applied to the multimodal projection layers and the attention blocks of the LLaMA3-8B backbone, while all modality encoders are kept frozen. This setup retains pretrained visual representations while enabling efficient fine-tuning for structured reasoning and executable action generation.

Training details. We perform supervised fine-tuning on MobileVLA-CoT to establish a stable reasoning-execution output schema. Training uses MobileVLA-

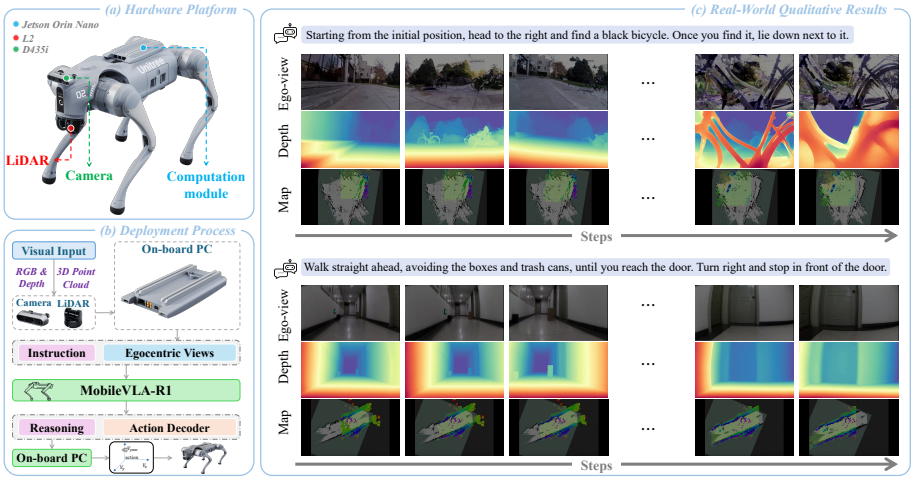

Fig. 5: Real-world deployment and qualitative examples on Unitree Go2. (a) Hardware platform: Unitree Go2 equipped with a Jetson Orin Nano for onboard computation, an L2 LiDAR for 3D sensing, and an Intel RealSense D435i RGB-D camera. **(b) Deployment pipeline:** synchronized RGB-D and LiDAR point clouds are processed to form the model inputs; MobileVLA-R1 generates structured outputs that are deterministically parsed into continuous velocity commands and discrete behaviors, which are executed in a closed loop on the robot. **(c) Real-World qualitative results:** example trajectories showing instruction-following behavior with ego-view RGB, depth, and a LiDAR-based local map over time.

CoT-Episode and MobileVLA-CoT-Nav, together with a subset of MobileVLA-CoT-Step containing executable control targets. We use LoRA with rank $r = 16$ and scaling $\alpha = 32$, AdamW, and a cosine learning-rate schedule. SFT is conducted on 4×H20 (96GB) GPUs with a learning rate of 2×10^{-4}. After SFT, we apply Group Relative Policy Optimization [46] to improve reasoning-to-action consistency and execution stability. For each input, we sample $N = 8$ candidate outputs and compute rewards for movement, discrete action, and output format. We use group-relative advantages and optimize the clipped GRPO objective with KL regularization to a frozen SFT reference policy. GRPO runs for 1K optimizer steps on a single H20 (96GB) GPU with $\beta = 0.04$, $\epsilon = 0.2$ (clip), and a learning rate of 1×10^{-6} using AdamW. Unless otherwise stated, we use a batch of 5 inputs per update step.

5.1 Main Results

Vision language navigation. Tab. 2 reports results on VLN-CE *val-unseen* for both R2R-CE [32] and RxR-CE [33]. MobileVLA-R1 achieves strong navigation performance, improving success-related metrics while reducing navigation error (NE) compared to prior methods under the same observational setting. These results indicate more accurate instruction following and trajectory efficiency in unseen environments.

Table 4: Real-world evaluation on Unitree Go2. We report SR and NE across Workspace, Corridor, and Outdoor tasks with Simple and Complex instructions; Outdoor uses SR only due to unreliable global localization.

Method	Workspace				Corridor				Outdoor			
	Simple		Complex		Simple		Complex		Simple		Complex	
	NE↓	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓	SR↑	NE	SR↑	NE	SR↑
GPT-4o [29]	2.01	0.67	2.38	0.33	1.49	0.53	3.00	0.00	–	0.67	–	0.50
NaVILA [12]	1.29	0.83	1.76	0.80	1.15	0.89	1.76	0.67	–	0.91	–	0.83
MobileVLA-R1	**1.03**	**0.93**	**1.23**	**0.91**	**0.96**	**1.00**	**1.23**	**0.86**	–	**1.00**	–	**0.96**

Quadruped control and manipulation. Tab. 3 summarizes QUARD performance under the official evaluation protocol [14]. MobileVLA-R1 achieves the highest average success rate among the compared methods, outperforming QUART [14] and MoRE [67]. Notably, MobileVLA-R1 improves performance across all difficulty levels, including the harder tasks.

5.2 Real-World Evaluation

Platform and deployment. As shown in Fig.5 (a), we evaluate on a Unitree Go2 equipped with an L2 LiDAR, an RGB-D camera, and a Jetson Orin module. Synchronized depth and LiDAR streams are fused into geometric observations for MobileVLA-R1 (Fig.5 (b)). All real-world experiments use a fixed hybrid setup rather than dynamic onboard/remote switching: sensing, mapping, parsing, action decoding, and low-level control run onboard, while only the 8B VLA forward pass is offloaded to a remote H20 due to Jetson memory limits. The reported hybrid end-to-end latency includes packaging, transmission, inference, parsing, and command return, and remains 205–245 ms across scenarios, compatible with our 5 Hz high-level control loop.

Table 5: Reward ablation for GRPO on R2R-CE Val-Unseen. We ablate movement, action, and format rewards; ✓ indicates enabled.

R_m	R_{act}	R_{fmt}	SR↑	SPL↑
✗	✗	✗	58.0	53.2
✓	✗	✗	60.7	55.5
✗	✓	✗	61.9	56.8
✗	✗	✓	59.6	54.7
✓	✓	✗	64.5	60.2
✓	✗	✓	63.4	59.1
✗	✓	✓	65.2	61.0
✓	✓	✓	**68.3**	**65.2**

Environments and task design. We conduct experiments in three representative environments: `Workspace`, `Corridor`, and `Outdoor`. For each environment, we design two instruction difficulty levels. *Simple* tasks contain one to two short commands (e.g., single goal reaching or a single interaction), whereas *Complex* tasks require multi-step instruction following with longer horizons (e.g., 3–5 subgoals, multiple turns, and obstacle-aware navigation). This design probes long-horizon grounding and execution robustness under clutter and partial observability.

Evaluation protocol and metrics. We report SR and NE for `Workspace` and `Corridor`, and SR only for `Outdoor` where accurate global localization is unavailable. All methods use the same sensing pipeline and are evaluated in closed

Table 6: Incremental modality encoder ablation on VLN-CE. We start from text and RGB and progressively add depth and point-cloud encoders.

Setting	R2R-CE		RxR-CE	
	SR↑	SPL↑	SR↑	SPL↑
Text & Image Encoder	62.5	59.0	66.0	61.5
+ Depth Encoder	66.0	62.0	69.0	64.0
+ Point Encoder	67.2	63.5	70.2	65.2
MobileVLA-R1 (Ours)	**68.3**	**65.2**	**71.5**	**66.8**

Table 7: Movement reward weight sweep on R2R-CE Val-Unseen. We report SR and SPL and each reward term's normalized advantage contribution (Adv%, normalized to sum to 100).

w_m	SR↑	SPL↑	Adv%$_m$	Adv%$_{act}$	Adv%$_{fmt}$
0.00	65.2	61.0	0	62	38
0.10	66.4	62.4	8	58	34
0.25	67.1	63.5	16	54	30
0.50	67.8	64.4	24	50	26
1.00	68.3	65.2	32	46	22

loop. Each scenario contains 5–6 tasks with 5 trials per task, totaling 160 real-world episodes. Success requires reaching the target, completing the requested behavior, and zero human intervention. Failures are grouped into target grounding, perception, over-turning, narrow-passage, and localization-drift errors, with detailed statistics in *Supp. Mat.*

Quantitative results. Tab. 4 summarizes the real-world results. MobileVLA-R1 achieves higher SR and lower NE than NaVILA [12] across `Workspace` and `Corridor`, and improves SR for both Simple and Complex instructions. Compared to GPT-4o [29], MobileVLA-R1 yields substantially higher success rates, especially on Complex instructions, indicating more reliable long-horizon execution on the physical platform.

Qualitative results. Fig. 1 and Fig. 5 (c) show qualitative examples. MobileVLA-R1 produces smooth locomotion trajectories and follows multi-step instructions in cluttered scenes. Additional qualitative visualizations and failure cases are provided in *Supp. Mat.*

5.3 Ablation Study

Effect of reward components. We study how different reward components affect GRPO training on R2R-CE Val-Unseen. As shown in Tab. 5, each reward term contributes to improved navigation success and trajectory efficiency relative to the SFT-only baseline. Using a single reward yields moderate gains, while combining two rewards consistently improves SR and SPL further. The best performance is obtained when all three rewards are enabled. Qualitatively, we observe distinct roles for each reward: R_m encourages stable and direction-consistent motion, R_{act} helps align high-level discrete behaviors with target actions, and R_{fmt} improves output validity by enforcing a machine-parseable schema. Together, these components provide complementary training signals for structured reasoning and executable control.

Effect of multimodal perception. We analyze the contribution of each sensory modality via an incremental encoder ablation. Starting from text and RGB, we progressively add depth and point-cloud encoders. Tab. 6 shows consistent improvements in SR and SPL on both R2R-CE and RxR-CE as additional modalities are included. Depth improves performance notably, suggesting the importance of geometric cues for local obstacle-aware navigation, and adding

Table 8: Necessity of multi-granularity CoT. We control teacher, prompt, output format, and rationale-token budget; ΔSR is relative to No-CoT.

CoT supervision	SR↑	SPL↑	ΔSR
No-CoT (actions only)	64.0	59.6	+0.0
Episode-only CoT	65.5	61.3	+1.5
Step-only CoT	66.0	61.7	+2.0
Navigation-level CoT	66.7	62.6	+2.7
Multi-granularity (Ours)	**68.3**	**65.2**	**+4.3**

Table 9: GRPO vs. PPO on R2R-CE. Both use matched rewards, schema, KL regularization, and training budget.

Method	SR↑	SPL↑
PPO (matched)	64.1	60.4
GRPO (Ours)	**68.3**	**65.2**

point-cloud features further increases SR and SPL, indicating additional benefits from 3D geometric representations. Overall, the full multimodal configuration achieves the best performance.

Reward weight sweep. We sweep the movement-reward weight w_m in $R = w_m R_m + \lambda_a R_{act} + \lambda_f R_{fmt}$ while keeping λ_a and λ_f fixed. Tab. 7 reports SR and SPL and the normalized advantage contributions (Adv%) of each term. As w_m increases, SR and SPL improve consistently, while the movement term's Adv% remains below the combined contribution of R_{act} and R_{fmt}, suggesting that dense motion alignment complements rather than replaces sparse supervision.

Necessity of multi-granularity CoT. We study whether CoT supervision provides benefits beyond confounds such as teacher choice, prompt, output format, and token budget. We control the teacher model and prompt, fix the total rationale-token budget per episode, and keep the executable command format and parser unchanged. **No-CoT** removes only the rationale text while preserving the same structured command format and action supervision. Tab. 8 shows that multi-granularity CoT achieves the best SR and SPL under these controlled settings, indicating improved long-horizon instruction following with the same supervision budget.

GRPO vs. PPO under matched budget. We compare GRPO with a PPO baseline under a matched training budget. Both methods use the same reward definitions, structured output format, KL regularization to the frozen SFT reference policy, and the same number of policy updates and samples per update. Tab. 9 shows that GRPO achieves higher SR/SPL than PPO on R2R-CE Val-Unseen, indicating the benefit of group-relative optimization for training structured reasoning-to-control policies.

Teacher-model ablation for CoT. To disentangle reasoning structure from teacher-model capacity, we vary only the rationale source while keeping command targets, parser, backbone, and training budget fixed. As shown in Tab. 10, Template-CoT and LLaMA-3-8B-CoT both improve over No-CoT, while Gemini-CoT performs best. Because all variants share the same executable command targets, the gains cannot be attributed solely to teacher-provided action labels. The improvement from Template-CoT shows that structured rationales already help organize the reasoning-to-command output space, while stronger LLaMA-3-8B and Gemini rationales further improve grounding. Thus, CoT structure itself is useful, and higher-quality rationales provide additional gains.

Table 10: Teacher ablation for CoT. Only the rationale source is changed, while command targets, parser, backbone, and training budget are fixed.

CoT source	SR	SPL
No-CoT	64.0	59.6
Template-CoT	65.1	60.8
LLaMA-3-8B CoT	66.2	62.0
Gemini-CoT	**68.3**	**65.2**

Table 11: GRPO vs. reward-shaped SFT. All variants use the same data, parser, command targets, and reward definitions.

Objective	Cand.	Rew.	Optimization	SR	SPL
SFT	✗	✗	token CE	58.0	53.2
Reward-SFT	✗	✓	reward-weighted CE	62.4	58.3
PPO	✓	✓	policy gradient	64.1	60.4
GRPO	✓	✓	group-relative	**68.3**	**65.2**

GRPO vs. reward-shaped SFT. To clarify what GRPO adds beyond supervised learning, we compare it with Reward-SFT, which uses the same parsed rewards to reweight token-level cross-entropy. As shown in Tab. 11, Reward-SFT improves over vanilla SFT, but remains below PPO and GRPO. Unlike Reward-SFT, GRPO samples complete `<think>`/`<answer>` outputs, parses them into executable commands, and optimizes output-level rewards for command validity, movement alignment, and behavior correctness. This indicates that group-relative command optimization provides benefits beyond token-level supervised reweighting.

6 Test-Time Efficiency

Efficient inference is important for deploying vision-language-action models on physical quadruped platforms. In our real-world evaluation, MobileVLA-R1 uses a fixed hybrid setup: onboard modules handle sensing, mapping, output parsing, action decoding, and low-level control, while only the 8B VLA forward pass is executed on a remote H20 server due to Jetson memory limits. The measured hybrid end-to-end latency is 205–245 ms across scenarios, including packaging, transmission, inference, parsing, command return, and execution handoff, which is compatible with our 5 Hz high-level control loop.

The current system still depends on remote inference because the 8B backbone exceeds the practical memory budget of the onboard Jetson module. This may limit deployment in bandwidth-constrained or highly dynamic environments.

7 Conclusion

In this work, we present MobileVLA-R1, a quadruped VLA framework that connects structured reasoning with executable continuous control. Through a reasoning-execution design, MobileVLA-R1 produces machine-parseable commands that are deterministically decoded into velocities and discrete behaviors. With multi-granularity CoT supervision and offline GRPO refinement, MobileVLA-R1 improves reasoning-to-command consistency across VLN-CE, QUARD, and real-world Unitree Go2 experiments. These results support explicit reasoning-to-control alignment for long-horizon quadruped instruction following.

Acknowledgements. This work was supported by the Fundamental Research Funds for the Central Universities, Peking University.

References

1. An, D., Wang, H., Wang, W., Wang, Z., Huang, Y., He, K., Wang, L.: Etpnav: Evolving topological planning for vision-language navigation in continuous environments. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
2. An, D., Wang, Z., Li, Y., Wang, Y., Hong, Y., Huang, Y., Wang, L., Shao, J.: 1st place solutions for rxr-habitat vision-and-language navigation competition. In: CVPR (2022)
3. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Computer Vision and Pattern Recognition. pp. 3674–3683 (2018)
4. Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., et al.: Do as i can, not as i say: Grounding language in robotic affordances. In: Conference on robot learning. pp. 287–318. PMLR (2023)
5. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. International Conference on 3D Vision (2017)
6. Chaplot, D.S., Gandhi, D.P., Gupta, A., Salakhutdinov, R.R.: Object goal navigation using goal-oriented semantic exploration. Advances in Neural Information Processing Systems **33**, 4247–4258 (2020)
7. Chen, J., Lin, B., Liu, X., Ma, L., Liang, X., Wong, K.Y.K.: Affordances-oriented planning using foundation models for continuous vision-language navigation. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 23568–23576 (2025)
8. Chen, K., Chen, J.K., Chuang, J., Vázquez, M., Savarese, S.: Topological planning with transformers for vision-and-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11276–11286 (2021)
9. Chen, P., Ji, D., Lin, K., Zeng, R., Li, T., Tan, M., Gan, C.: Weakly-supervised multi-granularity map learning for vision-and-language navigation. Advances in Neural Information Processing Systems **35**, 38149–38161 (2022)
10. Chen, S., Guhur, P.L., Schmid, C., Laptev, I.: History aware multimodal transformer for vision-and-language navigation. Advances in neural information processing systems **34**, 5834–5847 (2021)
11. Chen, Y., Sun, Y., Chen, X., Wang, J., Shen, X., Li, W., Zhang, W.: Integrating chain-of-thought for multimodal alignment: A study on 3d vision-language learning. arXiv preprint arXiv:2503.06232 (2025)
12. Cheng, A.C., Ji, Y., Yang, Z., Gongye, Z., Zou, X., Kautz, J., Bıyık, E., Yin, H., Liu, S., Wang, X.: Navila: Legged robot vision-language-action model for navigation. In: RSS (2025)
13. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
14. Ding, P., Zhao, H., Zhang, W., Song, W., Zhang, M., Huang, S., Yang, N., Wang, D.: Quar-vla: Vision-language-action model for quadruped robots. In: European Conference on Computer Vision. pp. 352–367. Springer (2024)
15. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P.,

Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., Florence, P.: Palm-e: an embodied multimodal language model. In: Proceedings of the 40th International Conference on Machine Learning (2023)
16. Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. Advances in neural information processing systems **31** (2018)
17. Gao, C., Jin, L., Peng, X., Zhang, J., Deng, Y., Li, A., Wang, H., Liu, S.: Octonav: Towards generalist embodied navigation. arXiv preprint arXiv:2506.09839 (2025)
18. Georgakis, G., Schmeckpeper, K., Wanchoo, K., Dan, S., Miltsakaki, E., Roth, D., Daniilidis, K.: Cross-modal map learning for vision and language navigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15460–15470 (2022)
19. Hao, W., Li, C., Li, X., Carin, L., Gao, J.: Towards learning a generic agent for vision-and-language navigation via pre-training. In: Computer Vision and Pattern Recognition. pp. 13137–13146 (2020)
20. Hong, Y., Rodriguez, C., Qi, Y., Wu, Q., Gould, S.: Language and visual entity relationship graph for agent navigation. Advances in Neural Information Processing Systems **33**, 7685–7696 (2020)
21. Hong, Y., Wang, Z., Wu, Q., Gould, S.: Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15439–15449 (2022)
22. Hong, Y., Wu, Q., Qi, Y., Rodriguez-Opazo, C., Gould, S.: A recurrent vision-and-language bert for navigation. In: Conference on Computer Vision and Pattern Recognition. pp. 1643–1653 (June 2021)
23. Hong, Y., Zhou, Y., Zhang, R., Dernoncourt, F., Bui, T., Gould, S., Tan, H.: Learning navigational visual representations with semantic map supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3055–3067 (2023)
24. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. Iclr **1**(2), 3 (2022)
25. Huang, T., Zhang, Z., Tang, H.: 3d-r1: Enhancing reasoning in 3d vlms for unified scene understanding. arXiv preprint arXiv:2507.23478 (2025)
26. Huang, T., Zhang, Z., Wang, Y., Tang, H.: 3d coca: Contrastive learners are 3d captioners. In: Thirteenth International Conference on 3D Vision (2026)
27. Huang, T., Zhang, Z., Zhang, R., Zhao, Y.: Dc-scene: Data-centric learning for 3d scene understanding. arXiv preprint arXiv:2505.15232 (2025)
28. Huang, W., Abbeel, P., Pathak, D., Mordatch, I.: Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In: International conference on machine learning. pp. 9118–9147. PMLR (2022)
29. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
30. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
31. Krantz, J., Lee, S.: Sim-2-sim transfer for vision-and-language navigation in continuous environments. In: European conference on computer vision. pp. 588–603. Springer (2022)

32. Krantz, J., Wijmans, E., Majumdar, A., Batra, D., Lee, S.: Beyond the nav-graph: Vision-and-language navigation in continuous environments. In: European Conference on Computer Vision. pp. 104–120. Springer (2020)
33. Ku, A., Anderson, P., Patel, R., Ie, E., Baldridge, J.: Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In: Empirical Methods in Natural Language Processing. pp. 4392–4412 (2020)
34. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
35. Liu, Q., Huang, T., Zhang, Z., Tang, H.: Nav-r1: Reasoning and navigation in embodied scenes. arXiv preprint arXiv:2509.10884 (2025)
36. Long, Y., Cai, W., Wang, H., Zhan, G., Dong, H.: Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. arXiv preprint arXiv:2406.04882 (2024)
37. Ma, C.Y., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., Xiong, C.: Self-monitoring navigation agent via auxiliary progress estimation. arXiv preprint arXiv:1901.03035 (2019)
38. Majumdar, A., Yadav, K., Arnaud, S., Ma, J., Chen, C., Silwal, S., Jain, A., Berges, V.P., Wu, T., Vakil, J., et al.: Where are we in the search for an artificial visual cortex for embodied intelligence? Advances in Neural Information Processing Systems **36**, 655–677 (2023)
39. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
40. Qi, Y., Pan, Z., Hong, Y., Yang, M.H., Van Den Hengel, A., Wu, Q.: The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation. In: International Conference on Computer Vision. pp. 1655–1664 (2021)
41. Qi, Z., Zhang, Z., Yu, Y., Wang, J., Zhao, H.: Vln-r1: Vision-language navigation via reinforcement fine-tuning. arXiv preprint arXiv:2506.17221 (2025)
42. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
43. Rajvanshi, A., Sikka, K., Lin, X., Lee, B., Chiu, H.P., Velasquez, A.: Saynav: Grounding large language models for dynamic planning to navigation in new environments. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 34, pp. 464–474 (2024)
44. Raychaudhuri, S., Wani, S., Patel, S., Jain, U., Chang, A.: Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In: Proceedings of the 2021 conference on empirical methods in natural language processing. pp. 4018–4028 (2021)
45. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
46. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
47. Song, W., Zhao, H., Ding, P., Cui, C., Lyu, S., Fan, Y., Wang, D.: Germ: A generalist robotic model with mixture-of-experts for quadruped robot. In: IROS. pp. 11879–11886. IEEE (2024)

48. Tang, H., Huang, T., Zhang, Z.: 3d coca v2: Contrastive learners with test-time search for generalizable spatial intelligence. arXiv preprint arXiv:2601.06496 (2026)
49. Tang, Y., Yu, W., Tan, J., Zen, H., Faust, A., Harada, T.: Saytap: Language to quadrupedal locomotion. arXiv preprint arXiv:2306.07580 (2023)
50. Tong, X., Ding, P., Fan, Y., Wang, D., Zhang, W., Cui, C., Sun, M., Zhao, H., Zhang, H., Dang, Y., et al.: Quart-online: Latency-free multimodal large language model for quadruped robot learning. In: ICRA. pp. 9533–9539. IEEE (2025)
51. Wang, C., Huang, T., Sun, C., Ning, X., Wang, D., Tang, H.: Let geometry guide: Layer-wise unrolling of geometric priors in multimodal llms. arXiv preprint arXiv:2604.05695 (2026)
52. Wang, H., Liang, W., Van Gool, L., Wang, W.: Dreamwalker: Mental planning for continuous vision-language navigation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10873–10883 (2023)
53. Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.F., Wang, W.Y., Zhang, L.: Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: Computer Vision and Pattern Recognition. pp. 6629–6638 (2019)
54. Wang, X., Wang, C., Xu, Y., Ye, M., Zhang, F.C., Tian, J., Zhan, X., Zhu, L., Lu, C., Yang, L.: Lamp: Learning vision-language-action policies with 3d scene flow as latent motion prior. arXiv preprint arXiv:2603.25399 (2026)
55. Wang, Z., Li, X., Yang, J., Liu, Y., Hu, J., Jiang, M., Jiang, S.: Lookahead exploration with neural radiance representation for continuous vision-language navigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13753–13762 (2024)
56. Wang, Z., Li, X., Yang, J., Liu, Y., Jiang, S.: Gridmm: Grid memory map for vision-and-language navigation. In: Proceedings of the IEEE/CVF International conference on computer vision. pp. 15625–15636 (2023)
57. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)
58. Wei, M., Wan, C., Yu, X., Wang, T., Yang, Y., Mao, X., Zhu, C., Cai, W., Wang, H., Chen, Y., et al.: Streamvln: Streaming vision-and-language navigation via slowfast context modeling. arXiv preprint arXiv:2507.05240 (2025)
59. Wu, J., Yin, S., Feng, N., Long, M.: Rlvr-world: Training world models with reinforcement learning. Advances in Neural Information Processing Systems **38**, 125312–125350 (2026)
60. Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: Simpler faster stronger. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4840–4851 (2024)
61. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. Advances in Neural Information Processing Systems **37**, 21875–21911 (2024)
62. Yu, B., Kasaei, H., Cao, M.: L3mvn: Leveraging large language models for visual target navigation. In: IROS. pp. 3554–3560. IEEE (2023)
63. Yu, Z., Long, Y., Yang, Z., Zeng, C., Fan, H., Zhang, J., Dong, H.: Correctnav: Self-correction flywheel empowers vision-language-action navigation model. arXiv preprint arXiv:2508.10416 (2025)
64. Zhang, J., Wang, K., Wang, S., Li, M., Liu, H., Wei, S., Wang, Z., Zhang, Z., Wang, H.: Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. RSS (2025)

65. Zhang, J., Wang, K., Xu, R., Zhou, G., Hong, Y., Fang, X., Wu, Q., Zhang, Z., Wang, H.: Navid: Video-based vlm plans the next step for vision-and-language navigation. In: RSS (2024)
66. Zhang, L., Wan, W., Huang, T., Fang, Z., Zhang, H.: Multigranularity-3dqa: Dynamic multi-granularity perception and gated dual-stage encoder-decoder for 3d question answering. Expert Systems with Applications p. 131676 (2026)
67. Zhao, H., Song, W., Wang, D., Tong, X., Ding, P., Cheng, X., Ge, Z.: More: Unlocking scalability in reinforcement learning for quadruped vision-language-action models. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 11212–11218. IEEE (2025)
68. Zhao, X., Lin, J., Liang, T., Zhou, Y., Chai, W., Gu, Y., Wang, W., Chen, K., Luo, G., Zhang, W., et al.: Mm-helix: Boosting multimodal long-chain reflective reasoning with holistic platform and adaptive hybrid policy optimization. arXiv preprint arXiv:2510.08540 (2025)
69. Zhu, F., Zhu, Y., Chang, X., Liang, X.: Vision-language navigation with self-supervised auxiliary reasoning tasks. In: Computer Vision and Pattern Recognition. pp. 10012–10022 (2020)
70. Zhu, Z., Wang, X., Li, Y., Zhang, Z., Ma, X., Chen, Y., Jia, B., Liang, W., Yu, Q., Deng, Z., et al.: Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In: International Conference on Computer Vision. pp. 8120–8132 (2025)
71. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183. PMLR (2023)