WHERE TO STAND, WHAT TO GRASP: OCCUPANCY-AWARE COUPLING OF MOBILITY AND MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Mobile manipulation is challenging not because navigation and manipulation are individually difficult, but because they are tightly coupled: the robot's base pose determines whether the arm can reach the target. Yet prevailing vision-language-action (VLA) models largely collapse this coupling into a monolithic action head, treating robot pose as merely another concatenated feature. We present M-VLA, which makes three modifications to the standard trunk- plus-flow-expert architecture. First, a Grounded Geometric Tokenizer (GGT) enables geometry-aware representation by converting the SLAM occupancy map and robot poses into metric-scale-preserving tokens within a shared physical coordinate frame, allowing map geometry and robot localization to interact through joint attention in the backbone. Second, Localization-as-Operator Fusion represents robot localization not as a conventional feature but as an $\mathrm{SE}(2)$ transformation operator $T_{\mathcal{B}}^{\mathcal{W}}$ that transports intent between the navigation (world) and manipulation (body) frames. It also incorporates an uncertainty-derived trust scalar distilled from the otherwise discarded SLAM covariance. Third, Dual Functional Experts replace the single action head with separate navigation and manipulation flow-matching experts coupled by a Coordination Bridge, enabling each expert to pretrain on abundant single-domain data while exchanging task intent at every layer. On a large- scale, long-horizon mobile-manipulation benchmark comprising tasks lasting from one to more than fourteen minutes and requiring cross-room navigation across multiple indoor scenes, M-VLA achieves the strongest overall performance.

1 INTRODUCTION

Vision-language-action (VLA) models have made fixed-base manipulation remarkably capable: a pretrained vision-language backbone supplies broad semantic knowledge, and a lightweight action decoder or increasingly a flow-matching expert that turns that knowledge into continuous control. Recent systems scale this recipe with larger backbones, heterogeneous co-training, and specialized experts, and a parallel line of work has pushed navigation and legged locomotion to comparable maturity. However, when the base is allowed to move, a challenge arises that neither line of work resolves independently. Mobile manipulation fails not because navigation or manipulation is individually hard, but because the two are coupled: where the base stops determines whether the arm can reach the target. Manipulation methods typically assume that the base is already appropriately positioned, whereas navigation methods terminate upon arrival, optimizing for reaching a pose rather than for reaching a pose from which manipulation is feasible.

Existing mobile-manipulation VLAs coordinate navigation and manipulation only implicitly. The prevailing design produces a single, undifferentiated whole- body action stream from a monolithic head. When multiple experts are used, they are typically partitioned according to computational considerations or action statistics rather than the robot's functional subsystems; consequently, no component is explicitly structured to represent constraints spanning locomotion and manipulation. Equally importantly, these models fail to exploit the much stronger geometric signal provided by SLAM and localization. The metric occupancy map, robot pose, and associated uncertainty jointly specify the robot's location, the navigable space surrounding it, and the target's distance and bearing. Prior VLAs leave this geometry largely unused: they rely primarily on RGB observations and proprioception, and even when a pose is provided, they concatenate it as merely another input feature.

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.

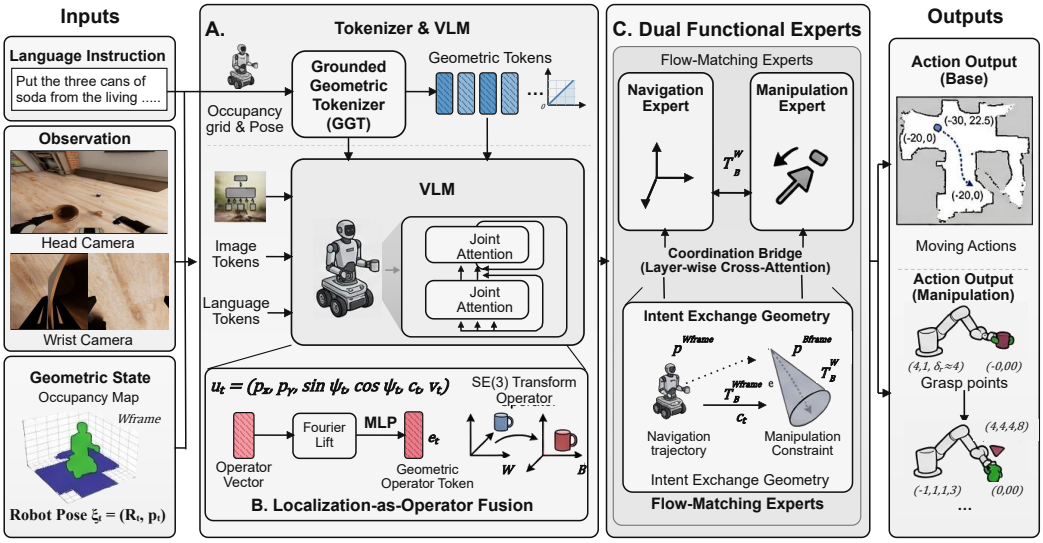


Figure 1: **Overview of M-VLA.** M-VLA introduces three key components: (1) a Grounded Geometric Tokenizer (GGT) that encodes the SLAM occupancy map and robot poses into metric-scale-preserving tokens in a shared physical coordinate frame; (2) Localization-as-Operator Fusion, which uses robot localization as an $\mathrm{SE}(2)$ transformation to transfer intent between navigation and manipulation frames, together with an uncertainty-derived trust scalar; and (3) Dual Functional Experts, consisting of navigation and manipulation flow-matching experts coupled by a Coordination Bridge for layer-wise intent exchange.

The network must therefore rediscover that the pose represents a rigid-body transformation between the navigation and manipulation frames.

To address this challenge, we present M-VLA, a vision-language-action model for mobile manipulation that treats base–arm coupling as a first-class architectural construct, as illustrated in Figure 1. M-VLA retains the efficient trunk-plus-flow-expert paradigm: a large vision-language backbone processes each decision once and is KV-cached, after which lightweight flow-matching action experts generate control commands. However, it reorganizes three components of this architecture around geometry. At the input level, a Grounded Geometric Tokenizer converts the SLAM occupancy map and robot poses into metric-scale-preserving tokens expressed in a shared physical coordinate system, which fuse bidirectionally with the vision and language tokens. At the representation level, robot localization is incorporated not as a concatenated feature, but as an $\mathrm{SE}(2)$ operator that transports geometric information between the world frame used for navigation and the body frame used for manipulation. This operator is accompanied by an uncertainty-derived trust signal computed from the SLAM covariance; together, both are encoded in a single checkpoint-safe prefix token accessible to both experts. At the output level, we replace the monolithic action head with two functionally specialized experts—one for navigation and one for manipulation— coupled through a Coordination Bridge that enables them to exchange intent at every layer. Each expert can therefore pretrain on abundant single-domain data and operate at its own horizon, while the shared localization operator maintains geometric consistency during information exchange. Together, these components transform the implicit coupling between navigation and manipulation into an explicit, differentiable architectural object.

We evaluate M-VLA on a large-scale benchmark for long-horizon mobile manipulation, comprising tasks in BEHAVIOR-1K within OmniGibson as well as experiments on real mobile robots. The benchmark is augmented with a SLAM module that provides occupancy maps and robot localization. Tasks range from one minute to more than fourteen minutes and require cross-room search across multiple indoor scenes, together with tightly coupled navigation and manipulation. M-VLA achieves the strongest overall performance in both simulation and real-robot settings, while ablation studies confirm that each proposed component contributes to the overall gains. In summary, our main contributions are as follows:

- We identify base–arm coupling as the core difficulty of mobile-manipulation VLAs and propose M-VLA, which promotes this coupling to a first-class architectural object rather than absorbing it into a single action head.
- We introduce the Grounded Geometric Tokenizer (GGT), which encodes geometric observations explicitly. GGT comprises an occupancy encoder that preserves metric scale and captures navigable clearance, together with a continuous metric-state tokenizer that represents the map and robot in a shared physical coordinate system, thereby enabling spatial grounding at the backbone level.
- We propose Localization-as-Operator Fusion, which treats robot localization as the $\mathrm{SE}(2)$ operator relating the navigation (world) and manipulation (body) frames, yielding position invariance across docking poses, and distills the usually-discarded SLAM covariance into a trust signal, both injected through a single checkpoint-safe prefix token.
- We design Dual Functional Experts, a navigation expert and a manipulation expert coupled by a Coordination Bridge, enabling separate single-domain pretraining, per-expert horizons, and geometrically consistent intent exchange.
- On a long-horizon BEHAVIOR-1K mobile-manipulation benchmark and in our real-robot evaluation, M-VLA achieves the strongest overall performance in both simulation and real-robot settings.

2 Related Work

2.1 VLA Manipulation

Vision-language-action (VLA) models pair a pretrained VLM trunk with an action decoder, and fixed-base manipulation is where the recipe is most mature. Early work established the paradigm and made it reproducible: RT-2 (Brohan et al., 2023) emitted actions as discretized text tokens to show that web knowledge transfers to control, while Octo (Team et al., 2024) and OpenVLA (Kim et al., 2024) scaled open transformer policies on Open X-Embodiment data (O'Neill et al., 2024). Because discretization is slow and coarse, later decoders moved to efficient tokenization and continuous heads, DCT action compression, parallel L1 decoding, and diffusion/flow-matching experts (Pertsch et al., 2025; Kim et al., 2024; Liu et al., 2025). The reference design is flow-matching expert on a VLM trunk in a mixture-of-transformers, whose large trunk runs once per decision (KV-cached) while a small expert denoises over the cached context, naive coupling can degrade both networks unless the interface is insulated or reinforcement-tuned (Driess et al., 2026; Lyu et al., 2025; Black et al., 2024)

Recent work scales this template along two axes. Generalization and lifelong improvement come from heterogeneous co-training and self-experience RL (Intelligence et al., 2025a), while internal structure comes from dual-system reasoning/control and mixture-of-experts heads sparsified by action statistics, LLM layers, or language conditioning (Bjorck et al., 2025; Shen et al., 2025; Zhang et al., 2026c). A dense wave of recent large VLAs extends the same trunk-plus-flow-expert recipe with larger backbones and data, alongside modular codebases, reproducibility studies, and surveys that consolidate the design space (Wang et al., 2026b; Team et al., 2026; Zhang et al., 2026a; Yang et al., 2026b; Gong et al., 2026; Community, 2026; Ye et al., 2026; Li et al., 2026a; Shao et al., 2025b). A grasping-specialized line pushes geometry and precision via billion-scale synthetic pretraining and hierarchical planner-plus-diffusion controllers (Deng et al., 2025; Zhong et al., 2026; He et al., 2025), and a complementary line supplies the metric structure RGB underdetermines through spatial-grounding and affordance objectives (Tu et al., 2026; Xu et al., 2026; Yu et al., 2026; Xiao et al., 2026).

2.2 Vision-Language Navigation and Locomotion

The base side of our concerns driving a robot to a commanded place through language, vision, and legged/wheeled motion. Its closest precedent is NaVILA (Cheng et al., 2024), a two-level VLA that emits mid-level spatial language commands executed by a low-level locomotion policy; related video-based VLAs unify diverse navigation tasks and next-step planning (Zhang et al., 2024a;b). A large zero-shot VLN-CE line then removes the need for demonstrations via waypoint prediction,

constraint-aware grounding, and action translation (Shi et al., 2025; Chen et al., 2025; Ding et al., 2025), while representation- and memory-centric methods improve long-horizon consistency (Wei et al., 2025; Sun et al., 2026; Liu et al., 2026; Wang et al., 2025). Paralleling the manipulation backbones above, navigation now also has generalist foundation models (Zhang et al., 2026b; Gong et al., 2026). On the locomotion layer most relevant to a mobile base, VLMs ground versatile legged skills, plan open-world quadruped tasks, follow language via motion latents, and even unify autonomous navigation and locomotion across terrains (Nahrendra et al., 2026; Wang et al., 2026a; Li et al., 2026b; Han et al., 2026).

2.3 Mobile / Locomotion Manipulation

Mobile manipulation must jointly decide where the base goes and how the arm acts, since a poor base placement leaves the target unreachable. One line augments the policy with persistent spatial state, declarative/spatiotemporal maps, panorama-aware and building-scale policies, and topology hierarchies that map instructions to base and arm commands (Lin et al., 2025; Kim et al., 2026; Yang et al., 2026a; Shah et al., 2025; Park et al., 2026; Gubernatorov et al., 2025). Most relevant to our method, a base-placement line makes movement explicit via inverse-reachability guidance, affordance-guided exploration, learned pose preferences, policy mobilization, and manipulability priors (Shao et al., 2025a; Lin et al., 2026; Chai et al., 2025; Yang et al., 2025; Zhang & Pajarinen, 2025). Whole-body loco-manipulation couples the two subsystems at the joint level through unified latents, empirical whole-body recipes, optimal-control synthesis, latent instruction, and MoE proprioceptive prediction (Jiang et al., 2026; Hu et al., 2026b; Boukheddimi et al., 2026). Finally, world-model and hierarchical methods add foresight or planner-skill structure to the loop (Fan et al., 2026; Zheng et al., 2026; Chen et al., 2026; Yang et al., 2026c; Hu et al., 2026a; Bai et al., 2025).

3 Method

3.1 Method Overview

We propose M-VLA, a vision-language-action (VLA) model for mobile manipulation that treats base–arm coupling as a first-class architectural object. Our central observation is that mobile manipulation is challenging not because navigation or manipulation is individually difficult, but because the two are tightly coupled: the base pose determines whether the arm can reach the target. Prior VLAs coordinate this coupling only implicitly; when they employ multiple experts, the expert decomposition is typically dictated by computational considerations or action statistics rather than by the robot's functional (navigation vs. manipulation) subsystems. In contrast, M-VLA comprises four components: (1) input tokens consisting of conventional image and language tokens together with our Grounded Geometric Tokens; (2) a 3B SigLIP+Gemma vision-language-model backbone; (3) Localization-as-Operator Fusion; and (4) Dual Functional Experts.

3.2 Grounded Geometric Tokenizer, GGT

Our input tokens consist of the following components: camera tokens, language tokens and Grounded Geometric Tokens (occupancy tokens + geometric tokens). Here the GGT is a specialized tokenizer designed in this work. The design principle of GGT is to let geometric observations enter the model as geometry: preserving metric scale, encoding metric quantities into the embedding space, and letting the map tokens share the same physical coordinate basis as the robot tokens, so as to accomplish spatial association. GGT attends bidirectionally with the rest of the preceding prefix, allowing spatial information to be fused with visual and linguistic information at every layer before being passed to the action experts.

GGT is composed with a geometry-aware occupancy encoder (OccEncoder) and a continuous metric-state tokenizer, it also accepts two kinds of data, from which it constructs a semantically clear, metric-scale-preserving multi-channel tensor. (1) Allocentric map-frame view: aligned to the map origin, providing global layout. It contains a signed distance to the nearest obstacle, explicitly providing navigable margin, a quantity that a VLM would otherwise have to laboriously estimate from a raw image. (2) Egocentric robot-position coordinates: the robot's position within the map, which is translation/rotation invariant, decoupling local navigation judgments such as "is the way

ahead passable" and "is the target to the left or right" from the robot's current heading. In addition, the resolution and origin position are no longer discarded; instead they enter the OccEncoder as scalar conditioning, so that each token can be assigned a true metric coordinate.

3.2.1 GEOMETRY-AWARE OCCUPANCY ENCODER, OCCENCODER

The OccEncoder encodes the occupancy map tensor into a token sequence: it takes the multi-channel tensor as input and outputs N_occ tokens whose dimensionality is projected to the same width as the image tokens, so they can be kept uniform with the image tokens. Each occupancy token carries the 2D metric position encoding of its patch, the world-frame center coordinate (x, y), added via generated 2D Fourier features. This gives each token the information which physical location on the map we represent, which is the basis of the spatial grounding.

3.2.2 CONTINUOUS METRIC-STATE TOKENIZER

The robot pose (x, y, yaw) is encoded directly into the embedding space. Each continuous quantity is encoded with a multi-band Fourier encoding; the results are concatenated and passed through MLPs to produce geometric tokens.

Optionally, we also retain $target$ (the relative displacement and bearing in the robot heading frame), explicitly giving "in which direction and how far the target is relative to me". This is the quantity may needed for navigation decisions and one that is not trivial for an LLM to obtain by differencing absolute coordinates. This pathway moves metric reasoning from language space into the embedding space, with far better precision and differentiability than strings. In real-world scenarios, however, reliable target information is often unavailable, so this optional pathway is disabled by default.

3.3 LOCALIZATION-AS-OPERATOR FUSION

The difficulty of mobile manipulation lies not only in navigation or manipulation individually, but in their coupling: where the base stops determines whether the arm can reach the target. To let a single model command both, it must share the same geometry between navigation and manipulation. But the two functional experts naturally live in different reference frames: the navigation expert reasons in the local world/map frame $\mathcal{W}$, the occupancy grid and target are all defined in $\mathcal{W}$, the manipulation expert reasons in the body frame $\mathcal{B}$.

The robot localization $\xi_t = (R_t, p_t) \in \mathrm{SE}(2)$ is essentially an operator: it is precisely the rigid-body transform $T_{\mathcal{B}}^{\mathcal{W}}$ connecting the two frames above. Any geometric quantity the navigation expert reasons about in the local world frame (a waypoint, a heading, a displacement $p^{\mathcal{W}}$) is expressed in the manipulation expert's body frame by acting with the operator,

$$p^{\mathcal{B}} \;=\; \xi_t^{-1} \cdot p^{\mathcal{W}} \;=\; R_t^{\top}\,(p^{\mathcal{W}} - p_t) \tag{1}$$

equation 1 is precisely using localization as an operator, rather than handing ξ_t to the network as a feature. The operator maps world-frame geometry into the frame in which the manipulation expert operates. This yields position invariance: regardless of where the base is positioned, a fixed physical configuration relative to the robot produces the same body-frame input. Consequently, a learned skill can be reused across different docking poses without requiring relearning for each pose. This is precisely why navigation and manipulation are assigned to separate experts: the operator, rather than the raw pose, makes their interaction coherent.

Just as important as the pose transform is its trustworthiness. Alongside ξ_t, SLAM also provides the covariance Σ_t, which quantifies the reliability of the estimated transformation, yet this signal is almost always discarded in existing work. From Σ_t we define a scalar localization trust

$$c_t \;=\; \exp\!\Big(-\tfrac{1}{2}\,\varphi(\Sigma_t)\Big) \;\in\; (0,1], \qquad \varphi(\Sigma_t) \;=\; \frac{\Sigma_{xx}+\Sigma_{yy}}{\sigma_p^2} \;+\; \frac{\Sigma_{\psi\psi}}{\sigma_\psi^2} \tag{2}$$

where $\varphi(\Sigma_t)$ is a dimensionless, rotation-invariant scalar of the pose covariance Σ_t. Because the translational variances $(\Sigma_{xx}, \Sigma_{yy})$ carry units of m^2 while the yaw variance $\Sigma_{\psi\psi}$ carries units of rad^2, we normalize each by its own reference noise scale, σ_p (metres) and σ_ψ (radians), before summing, so that φ is unitless; in our implementation $\sigma_p = 0.1\,\mathrm{m}$ and $\sigma_\psi = 5^\circ$ (a sensitivity study

over these scales is reported in the appendix). A well-localized pose gives $c_t \to 1$ and a severely drifted one gives $c_t \to 0$. When localization is missing, c_t and the entire feature are set to zero together and replaced by a learnable placeholder vector (see §3.3.2).

3.3.1 GROUNDED GEOMETRIC FEATURE

We pack the robot localization and localization trust into a fixed-length grounded geometric feature $\boldsymbol{u}_t \in \mathbb{R}^6$ that summarizes the localization operator itself, together with its trust. Let the robot's planar pose in the local world frame be (p_x, p_y) with base yaw ψ_t. The feature is

$$\boldsymbol{u}_t = \big(\underbrace{p_x,\ p_y}_{\text{world-frame position}},\ \underbrace{\sin\psi_t,\ \cos\psi_t}_{\text{base orientation}},\ \underbrace{c_t}_{\text{trust}},\ \underbrace{v_t}_{\text{validity bit}} \big) \tag{3}$$

where $v_t \in \{0, 1\}$ marks whether localization is available; when $v_t = 0$ the entire $\boldsymbol{u}_t$ is a zero vector. The yaw is represented as $(\sin\psi_t, \cos\psi_t)$ to avoid the discontinuity of angle wrap-around. These are exactly the components of the operator $\xi_t = (R_t, p_t)$, so a downstream module can recover the world→body map from $\boldsymbol{u}_t$ alone. $\boldsymbol{u}_t$ is the expected feature.

3.3.2 INJECTING THE GEOMETRIC OPERATOR TOKEN

We want both experts to use $\boldsymbol{u}_t$ without disturbing the existing backbone or changing the shape of any existing tensor, so that a checkpoint pretrained without SLAM can be loaded. To this end, we encode $\boldsymbol{u}_t$ into a single prefix token and attach it to the joint attention of the vision-language backbone.

Concretely, we first apply a Fourier lift ($\sin$ / $\cos$ expansion) to $\boldsymbol{u}_t$ to amplify high-frequency geometric information, then pass it through MLP layers that projects it to the backbone width d:

$$\boldsymbol{e}_t \ = \ W_2\,\mathrm{swish}\big(W_1\,[\sin\boldsymbol{u}_t; \cos\boldsymbol{u}_t]\big) \ \in \ \mathbb{R}^d \tag{4}$$

When $v_t = 0$ (SLAM missing or invalid), a learnable placeholder embedding $\boldsymbol{e}_{\mathrm{ph}}$ replaces $\boldsymbol{e}_t$:

$$\tilde{\boldsymbol{e}}_t \ = \ \begin{cases} \boldsymbol{e}_t, & v_t = 1, \\ \boldsymbol{e}_{\mathrm{ph}}, & v_t = 0. \end{cases} \tag{5}$$

$\tilde{\boldsymbol{e}}_t$ is appended as an extra token to the end of the bidirectional (non-causal) prefix block, sitting in the same attention segment as the image and language tokens. Because the backbone uses shared joint attention, this single token is simultaneously visible to both the navigation expert and the manipulation expert, so both can see the localization operator relating their frames and its trust while reading their own action tokens. This is exactly how localization is realized as the glue between the two experts: one geometrically well-defined shared token, rather than a pile of concatenated features.

3.4 DUAL FUNCTIONAL EXPERTS

Rather than outputting whole-body actions with a single monolithic action head, our method decomposes the action space into two functionally specialized flow-matching experts. The large vision-language trunk runs only once per decision and is KV-cached, and two lightweight experts iterate over K denoising steps.

3.4.1 DUAL EXPERTS

There are two mobile-manipulation experts: a navigation expert and a manipulation expert, with per-expert adaptive RMSNorm enabled on both action experts. Each expert has its own input/output action projections. Thus a single language-model call consumes a triple list $[\,\text{prefix tokens},\ \text{nav suffix},\ \text{manip suffix}\,]$ and returns output streams. The flow-matching objective L_{fm} is formally unchanged, only computed once per expert,

$$L_{\mathrm{fm}} = L_{\mathrm{fm}}^{\mathrm{nav}} + L_{\mathrm{fm}}^{\mathrm{man}}, \qquad L_{\mathrm{fm}}^{e} = \mathbb{E}_{\tau,a_0,a_1}\Big[\ \big\|v_\theta^e(a_\tau,\tau) - (a_1 - a_0)\big\|^2\ \Big] \tag{6}$$

where a_0 and a_1 are the flow matching start and end points, $a_\tau = (1-\tau)a_0 + \tau a_1$, and v_θ^e is the predicted flow-matching velocity field. This functional split lets each expert pretrain on abundant single-domain data (navigation and manipulation) and jointly train only on scarce mobile-manipulation data. That is, they can be trained and deployed separately at both the navigation and manipulation levels.

3.4.2 EXPERT INTERACTION

The function of this module is to make the two experts a team with information exchange (rather than two independent policies), we call the Coordination Bridge. In this architecture, the bridge is realized as interaction at every layer via shared self-attention plus masking, letting the navigation features and manipulation features see each other, and letting both attend to the shared (cached) VLM features. Because the two streams see each other at every layer, the navigation expert can communicate the current base position, which the manipulation expert interprets as a constraint, while the manipulation expert can communicate the base placement it requires, which the navigation expert interprets as a world-frame objective. Localization provides the geometric basis for this exchange. The navigation expert naturally reasons in the local world/map frame $\mathcal{W}$, in which the occupancy grid, target points, and affordance field (M(x,y)) are defined. Here, (M(x,y)) is a map-frame scalar field that scores the suitability of each base location for the current manipulation goal. The manipulation expert naturally reasons in the body/end-effector frame $\mathcal{B}$, grasps and end-effector increments all live in $\mathcal{B}$; Localization $\xi_t = (R_t, p_t) \in \mathrm{SE}(2)$ is the bridge $T_{\mathcal{B}}^{\mathcal{W}}$ between these two frames, while SLAM additionally provides its covariance Σ_t.

We fuse robot localization into the two experts not as a pose feature, but as an $\mathrm{SE}(2)$ operator that transports intent between the navigation (world) frame and the manipulation (body) frame, and as a bridge used to dynamically exchange the information of navigation and manipulation experts between map-frame control and egocentric control.

4 EXPERIMENTS

4.1 DATASETS

To fairly compare with existing methods, we evaluate our method on a large-scale benchmark for long-horizon mobile manipulation built on the BEHAVIOR-1K environment (simulated in Omni-Gibson) (Li et al., 2024) and a real-world robot. The datasets are collected on a mobile bimanual robot with a whole-body interface controlling the base, torso, arms, and grippers. Each demonstration records RGB and depth observations, robot proprioception, and actions. We also add a SLAM module in the robot environment, and collect the SLAM occupancy map and robot localization information. The average operation duration is over 6 minutes, and the hard tasks can exceed 14 minutes; the scenarios require the robot to move from one room to another across multiple indoor scenes to finish the task.

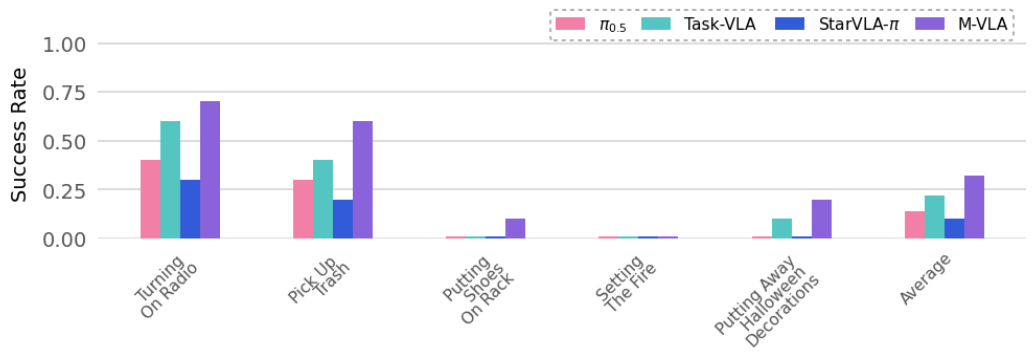


Figure 2: Living Space results. Success rates under different variations and distractions, showing that our method consistently outperforms existing methods across diverse navigation and manipulation tasks.

4.2 EXPERIMENTAL SETUPS

All experiments are trained on 16 NVIDIA H200 and 16 NVIDIA A100 GPUs, tested on 5 NVIDIA RTX 4090, using a cosine decay scheduler. We use 3B SigLIP+Gemma as our VLM backbone. For all multi-task pre-training, we use 80k training steps, with a learning rate of 2.5×10^{-5}. We also fine-tune our method for 30k steps with learning rate of 3.5×10^{-6}.

4.3 SIMULATED EXPERIMENTS

We evaluate our method on the BEHAVIOR simulation benchmark implemented in OmniGibson. The benchmark features an R1Pro mobile robot with 23 degrees of freedom and demonstrations containing camera images, robot states, task annotations, and additional SLAM maps and localization estimates. For each task, we use 200 human demonstrations. We assess performance on diverse long- horizon mobile-manipulation tasks, including housework in living rooms, bedrooms, and kitchens. Notably, the robot locations used for training and testing are different.

In this experiment, we compare M-VLA against more recent existing methods ($\pi 0.5$ (Intelligence et al., 2025b), Task-VLA (Larchenko et al., 2025) and StarVLA-π (Community, 2026)) that were able to train and evaluate on mobile manipulation dataset and might reasonably be expected to perform better on public benchmark. Below is the quantitative experiment results.

Living Space: Figure 2 reports absolute success rates on the living space tasks. M-VLA attains the best rate on tasks and the best average (0.32 vs. 0.22 for the strongest baseline, a 16.7% relative gain on Turning On Radio and 50% and 100% relative gain on Pick Up Trash and Putting Away Halloween Decorations). It is also the only method with a non-zero rate on Putting Shoes On Rack (0.10 vs. 0.00 for all baselines). The experiment suggested that our method could understand the moving and manipulation in different frames(world/map frame and body frame), and reduce the accumulation errors, and make the whole task is successfully executed.

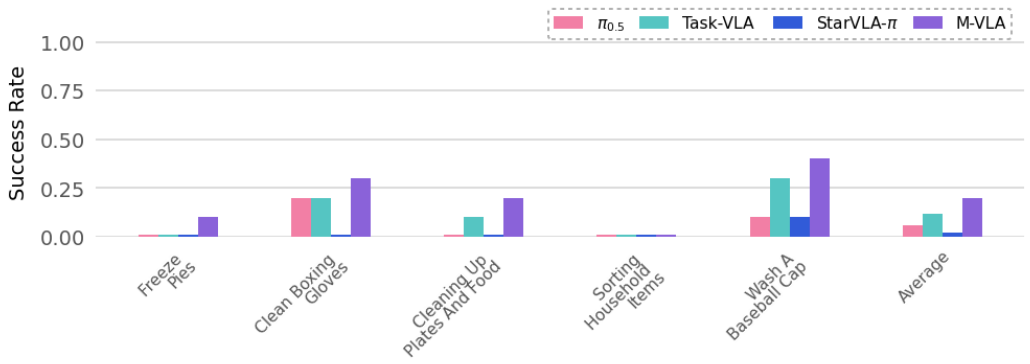


Figure 3: Kitchen & Working Space task results. Long-horizon experiments show that our method outperforms existing approaches in complex tasks, particularly those requiring large-area navigation and long-term manipulation.

Kitchen & Working Space: We also choose long-term kitchen and working space housework tasks (the passage is narrow), whose trajectory durations range from 6 to 14 minutes, to evaluate mobile-manipulation performance in more complex environments. The success rates for kitchen tasks are illustrated in Figure 3. On these tasks, which demand robustness to large-area movement and long-horizon manipulation, M-VLA improves over the strongest baseline on these tasks: the best average success rate of 0.20, absolute success rises from 0.20 to 0.30 on Clean Boxing Gloves (50% relative), from 0.10 to 0.20 on Cleaning Up Plates And Food (100% relative), and 33.3% relative improvement on Wash A Baseball Cap. M-VLA is again the only method with a non-zero rate on Freeze Pies (0.10 vs. 0.00). We find that our alternative VLA design empower mobile manipulation VLA to establish the strongest method on our BEHAVIOR-based benchmark.

Table 1: Real-robot results. Deployment on a physical mobile bimanual robot demonstrates improved performance. *Progress* is the fraction of task subgoals completed (in $[0, 1]$), and *Success Rate* is the fraction of trials in which the full task is completed.

Method	**Package Sorting**		**Shelf Placement**	
	Progress ↑	Success Rate ↑	Progress ↑	Success Rate ↑
$\pi 0.5$ (Intelligence et al., 2025b)	0.71	0.57	0.74	0.53
M-VLA	0.83	0.70	0.87	0.63

4.4 Real-World Experiments

To verify that the gains observed in simulation transfer to a physical robot with genuine SLAM uncertainty, we deploy the trained policy on a real mobile bimanual platform and evaluate it on mobile-manipulation tasks, running 30 trials per task per method. Table 1 reports the real-robot results against the existing strongest model ($\pi 0.5$). M-VLA attains the higher success rate on both task families (0.70 vs. 0.57 on Package Sorting and 0.63 vs. 0.53 on Shelf Placement), and also improves Progress by 0.12 and 0.13, respectively. This real-world experiment indicates that our method remains effective in practical scenarios.

4.5 Ablation Studies

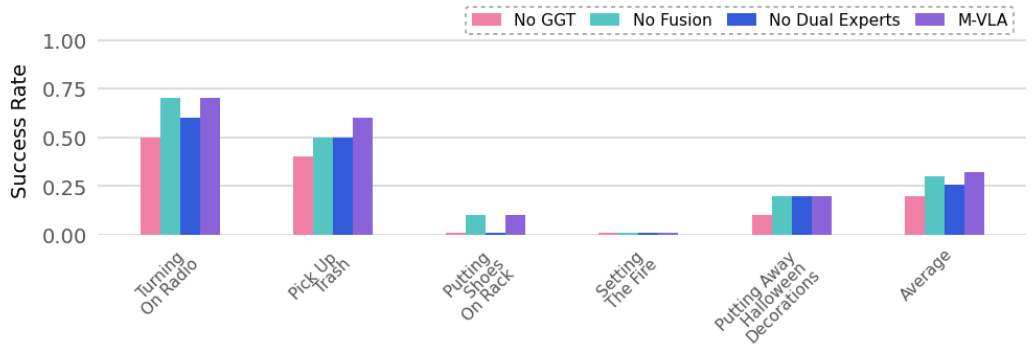


Figure 4: Ablation study of M-VLA, demonstrating the contributions of components to final performance.

We conduct ablation studies on the living-space tasks of BEHAVIOR to evaluate the contribution of each component; results are summarized in Figure 4. All variants share the full M-VLA training recipe and differ only in the ablated part. In these experiments, we ablate different parts of the mixture. **No GGT:** Removing the Grounded Geometric Tokens entirely (no occupancy or geometric tokens) is the single most damaging ablation, dropping average success from 0.32 to 0.20 (-37.5% relative), indicating that our method benefits considerably from slam occupancy and localization information in these tasks. **No Fusion:** Dropping Localization-as-Operator Fusion (the operator token) costs 12.5% decrease($0.32 \rightarrow 0.28$). The results suggest that localization-as-operator fusion contribute to the model's ability to generalize across diverse tasks, thereby enhancing the overall performance of our method. **No Dual Experts:** Replacing the two functional experts with a single expert that emits whole-body actions lowers success to 0.26. Resulting in a slight performance degradation compared to the dual experts design. Overall, these ablation studies demonstrate that each component of our method contributes to its final performance, confirming the effectiveness of the proposed design.

5 Conclusion

We presented M-VLA, a vision-language-action model that treats base–arm coupling as a first-class architectural object. Motivated by the fact that mobile manipulation fails at the seam between the two subsystems, where the base stops decides whether the arm can reach, M-VLA makes three changes to the standard trunk-plus-flow-expert recipe: the Grounded Geometric Tokenizer feeds the SLAM map and poses to the backbone as metric-scale-preserving tokens on a shared coordinate basis; Localization-as-Operator Fusion uses the robot pose as the $\mathrm{SE}(2)$ operator that transports intent between world and body frames, with a trust scalar distilled from the otherwise-discarded SLAM covariance; and Dual Functional Experts replace the single head with a navigation and a manipulation expert coupled by a Coordination Bridge. On a long-horizon BEHAVIOR-1K benchmark spanning one to over fourteen minutes and cross-room motion across multiple indoor scenes, M-VLA sets a new state of the art, with ablations confirming that each component contributes. These results suggest that naming the robot's functional subsystems and feeding geometry to them as geometry is a more productive path than scaling an undifferentiated policy.

AI USE STATEMENT

In this work, we used generative AI tools for polish writing, generating synthetic datasets. We have not used generative AI tools for other tasks such as discovery, research ideation. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Junjie Bai, Yu-Wei Chao, Qizhi Chen, Jinwei Gu, Moo Jin Kim, Zhaoshuo Li, Xuan Li, Tsung-Yi Lin, Ming-Yu Liu, Nic Ma, et al. Openpi comet: Competition solution for 2025 behavior challenge. *arXiv preprint arXiv:2512.10071*, 2025.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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

Melya Boukheddimi, Omar Adjali, Daniel Sontag, and Frank Kirchner. Wolf-vla: Whole-body humanoid optimal locomotion framework for vision-language-action learning. *arXiv preprint arXiv:2606.25591*, 2026.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Kaixin Chai, Hyunjun Lee, and Joseph J Lim. N2m: Bridging navigation and manipulation by learning pose preference from rollout. *arXiv preprint arXiv:2509.18671*, 2025.

Kehan Chen, Dong An, Yan Huang, Rongtao Xu, Yifei Su, Yonggen Ling, Ian Reid, and Liang Wang. Constraint-aware zero-shot vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, et al. Abot-m0. 5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026.

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. *arXiv preprint arXiv:2412.04453*, 2024.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Wenhao Zhang, et al. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. *arXiv preprint arXiv:2505.03233*, 2025.

Hongyu Ding, Ziming Xu, Yudong Fang, You Wu, Zixuan Chen, Jieqi Shi, Jing Huo, Yifan Zhang, and Yang Gao. Lavira: Language-vision-robot actions translation for zero-shot vision language navigation in continuous environments. *arXiv preprint arXiv:2510.19655*, 2025.

Danny Driess, Jost Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *Advances in Neural Information Processing Systems*, 38:102867–102888, 2026.

Zehua Fan, Junjie He, Wenxuan Song, Xi Wang, Wenqi Lyu, Linge Zhao, Fuhao Li, Zihan You, Yifei Yang, Kaiming Xu, et al. Mobilewam: Bridging world action models to mobile manipulation with chain-of-foresight. *arXiv preprint arXiv:2608.04657*, 2026.

Ruiyan Gong, Yingnan Guo, Junjun Hu, Jintao Kong, Xiaoxu Leng, Tianlun Li, Weize Li, Fei Liu, Zhicheng Liu, Jia Lu, et al. Abot-n1: Toward a general visual language navigation foundation model. *arXiv preprint arXiv:2607.10383*, 2026.

Konstantin Gubernatorov, Artem Voronov, Roman Voronov, Sergei Pasynkov, Stepan Perminov, Ziang Guo, and Dzmitry Tsetserukou. Anywherevla: Language-conditioned exploration and mobile manipulation. *arXiv preprint arXiv:2509.21006*, 2025.

Haoxuan Han, Chen Chen, Linao Gong, Xin Yang, Hao Hu, Junhong Guo, Zhicheng He, Yao Su, and Fenghua He. Guidewalk: Learning unified autonomous navigation and locomotion for humanoid robots across versatile terrains. *arXiv preprint arXiv:2606.10449*, 2026.

Jiawei He, Danshi Li, Xinqiang Yu, Zekun Qi, Wenyao Zhang, Jiayi Chen, Zhaoxiang Zhang, Zhizheng Zhang, Li Yi, and He Wang. Dexvlg: Dexterous vision-language-grasp model at scale. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14248–14258. IEEE, 2025.

Jiaheng Hu, Mohit Shridhar, Caden Lu, Dhruv Shah, Hao-Tien Lewis Chiang, Jie Tan, and Annie Xie. What matters in orchestrating robot policies: A systematic study of hierarchical vla agents. *arXiv preprint arXiv:2606.10267*, 2026a.

Yingdong Hu, Haodong Zhu, Boyuan Zheng, Yihang Hu, Tong Zhang, Zunhao Chen, Junming Zhao, Ruiqian Nai, and Yang Gao. Openhlm: An empirical recipe for whole-body humanoid loco-manipulation. *arXiv preprint arXiv:2606.22174*, 2026b.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a vla that learns from experience, 2025a. URL https://arxiv.org/abs/2511.14759.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025b. URL https://arxiv.org/abs/2504.16054.

Haoran Jiang, Jin Chen, Qingwen Bu, Li Chen, Modi Shi, Yanjie Zhang, Delong Li, Chuanzhe Suo, Hongyang Li, et al. Wholebodyvla: Towards unified latent vla for whole-body loco-manipulation control. In *International Conference on Learning Representations*, volume 2026, pp. 157438–157461, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Sunghwan Kim, Byeonghyun Pak, Kehan Long, Yulun Tian, and Nikolay Atanasov. Serf: Spatiotemporal environment and robot feature map for long-horizon mobile manipulation. *arXiv preprint arXiv:2606.12956*, 2026.

Ilia Larchenko, Gleb Zarin, and Akash Karnatak. Task adaptation of vision-language-action model: 1st place solution for the 2025 behavior challenge, 2025. URL https://arxiv.org/abs/2512.06951.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, Hang Yin, Michael Lingelbach, Minjune Hwang, Ayano Hiranaka, Sujay Garlanka, Arman Aydin, Sharon Lee, Jiankai Sun, Mona Anvari, Manasi Sharma, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Yunzhu Li, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024.

Xinghang Li, Peiyan Li, Long Qian, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Xinlong Wang, Di Guo, et al. What matters in building vision–language–action models for generalist robots. *Nature Machine Intelligence*, 8(2):158–172, 2026a.

Zhe Li, Yangyang Wei, Boan Zhu, Yibo Peng, Tao Huang, Pengwei Wang, Zhongyuan Wang, Cheng Chi, Chang Xu, and Shanghang Zhang. From language to locomotion: Retargeting-free humanoid control via motion latent guidance. In *International Conference on Learning Representations*, volume 2026, pp. 129292–129319, 2026b.

Min Lin, Xiwen Liang, Bingqian Lin, Liu Jingzhi, Zijian Jiao, Kehan Li, Yu Sun, Weijia Liufu, Yuhan Ma, Yuecheng Liu, et al. Echovla: Synergistic declarative memory for vla-driven mobile manipulation. *arXiv preprint arXiv:2511.18112*, 2025.

Tzu-Jung Lin, Jia-Fong Yeh, Hung-Ting Su, Chung-Yi Lin, Yi-Ting Chen, and Winston H Hsu. Affordance-guided coarse-to-fine exploration for base placement in open-vocabulary mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18443–18451, 2026.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Yuqi Liu, Shengju Qian, Tianyuan Qu, Mingxian Lin, Zixuan Wang, Xin Wang, Bei Yu, and Jiaya Jia. Memvln: Episodic and procedural memory for vision-and-language navigation. *arXiv preprint arXiv:2607.23504*, 2026.

Mingyang Lyu, Yinqian Sun, Erliang Lin, Huangrui Li, Ruolin Chen, Feifei Zhao, and Yi Zeng. Reinforcement fine-tuning of flow-matching policies for vision-language-action models. *arXiv preprint arXiv:2510.09976*, 2025.

I Nahrendra, Seunghyun Lee, Dongkyu Lee, and Hyun Myung. Locovlm: Grounding vision and language for adapting versatile legged locomotion policies. *arXiv preprint arXiv:2602.10399*, 2026.

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 collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Jeong-Seop Park, Yong-Jun Lee, Jong-Chan Park, Sung-Gil Park, Jong-Jin Woo, and Myo-Taeg Lim. Exaone-vla: A unified vision–language framework for mobile manipulation via semantic topology and hierarchical llm reasoning. *Applied Sciences*, 16(5):2600, 2026.

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.

Rutav Shah, Albert Yu, Yifeng Zhu, Yuke Zhu, and Roberto Martín-Martín. Bumble: Unifying reasoning and acting with vision-language models for building-wide mobile manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13337–13345. IEEE, 2025.

Beichen Shao, Nieqing Cao, Yan Ding, Xingchen Wang, Fuqiang Gu, and Chao Chen. Moma-pos: An efficient object-kinematic-aware base placement determination framework for mobile manipulation. In *International Conference on Algorithms and Architectures for Parallel Processing*, pp. 254–270. Springer, 2025a.

Rui Shao, Wei Li, Lingsen Zhang, Renshan Zhang, Zhiyang Liu, Ran Chen, and Liqiang Nie. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint arXiv:2508.13073*, 2025b.

Weijie Shen, Yitian Liu, Yuhao Wu, Zhixuan Liang, Sijia Gu, Dehui Wang, Tian Nian, Lei Xu, Yusen Qin, Jiangmiao Pang, et al. Expertise need not monopolize: Action-specialized mixture of experts for vision-language-action learning. *arXiv preprint arXiv:2510.14300*, 2025.

Xiangyu Shi, Zerui Li, Wenqi Lyu, Jiatong Xia, Feras Dayoub, Yanyuan Qiao, and Qi Wu. Smartway: Enhanced waypoint prediction and backtracking for zero-shot vision-and-language navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 16923–16930. IEEE, 2025.

Josh Qixuan Sun, Huaiyuan Weng, Xiaoying Xing, Chul Min Yeum, and Mark Crowley. View invariant learning for vision-language navigation in continuous environments. *IEEE Robotics and Automation Letters*, 2026.

HY Team, Xumin Yu, Zuyan Liu, Ziyi Wang, He Zhang, Yongming Rao, Fangfu Liu, Yani Zhang, Ruowen Zhao, Oran Wang, et al. Hy-embodied-0.5: Embodied foundation models for real-world agents. *arXiv preprint arXiv:2604.07430*, 2026.

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. *arXiv preprint arXiv:2405.12213*, 2024.

Ruisen Tu, Arth Shukla, Sohyun Yoo, Xuanlin Li, Junxi Li, Jianwen Xie, Hao Su, and Zhuowen Tu. Sg-vla: Learning spatially-grounded vision-language-action models for mobile manipulation. *arXiv preprint arXiv:2603.22760*, 2026.

Kaijun Wang, Liqin Lu, Mingyu Liu, Jianuo Jiang, Zeju Li, Bolin Zhang, Wancai Zheng, Xinyi Yu, Hao Chen, and Chunhua Shen. Odyssey: Open-world quadrupeds exploration and manipulation for long-horizon tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18602–18610, 2026a.

Liuyi Wang, Xinyuan Xia, Hui Zhao, Hanqing Wang, Tai Wang, Yilun Chen, Chengju Liu, Qijun Chen, and Jiangmiao Pang. Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9455–9465. IEEE, 2025.

Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, et al. Qwen-vla: Unifying vision-language-action modeling across tasks, environments, and robot embodiments. *arXiv preprint arXiv:2605.30280*, 2026b.

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.

Siyao Xiao, Yuhong Zhang, Zhifang Liu, Zihan Gao, Jingye Zhang, Sinwai Choo, Dake Zhong, Mengzhe Wang, Xiao Lin, Xianfeng Zhou, et al. π2-vla: Boosting vision-language models for generalizable manipulation via layer mixture and meta-skills. *arXiv preprint arXiv:2604.24182*, 2026.

Siyu Xu, Zijian Wang, Yunke Wang, Chenghao Xia, Tao Huang, and Chang Xu. Affordance field intervention: Enabling vlas to escape memory traps in robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37206–37215, 2026.

Donglin Yang, Haoran Chen, Xingyu Chen, Lixing Liu, Manyi Li, Changhe Tu, Ke Xu, Xiaojian Ma, and Si Liu. Learning panorama-aware vla for mobile manipulation with whole-body teleoperation. *arXiv preprint arXiv:2608.02257*, 2026a.

Jingyun Yang, Isabella Huang, Brandon Vu, Max Bajracharya, Rika Antonova, and Jeannette Bohg. Mobi-π: Mobilizing your robot learning policy, 2025. URL https://arxiv.org/abs/2505.23692.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026b.

Zheng Yang, Wenjie Zhang, Xiangyu Chen, Wenxuan Song, Xianpeng Wang, Yihang Kang, Wen Chen, Lujia Wang, Renjing Xu, and Xiaowen Chu. Dreamtrajectory: Trajectory-guided action generation with world model alignment for mobile manipulation. *arXiv preprint arXiv:2608.01381*, 2026c.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. Starvla-α: Reducing complexity in vision-language-action systems, 2026. URL https://arxiv.org/abs/2604.11757.

Qize Yu, Jiadi You, Yuran Wang, Jiaqi Liang, Bowen Ping, Yang Tian, Yue Chen, Minghong Cai, Zeying Gong, Ruihai Wu, et al. Affordancevla: A vision-language-action model empowering action generation through affordance-aware understanding. *arXiv preprint arXiv:2606.06155*, 2026.

He Zhang, Lingzhu Xiang, Haitao Lin, Zeyu Huang, Minghui Wang, Dingyan Zhong, Yubo Dong, Yihao Wu, Yongming Rao, Dongsheng Zhang, et al. Hy-embodied-0.5-vla: From vision-language-action models to a real-world robot learning stack. *arXiv preprint arXiv:2606.14409*, 2026a.

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. *arXiv preprint arXiv:2412.06224*, 2024a.

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. *arXiv preprint arXiv:2402.15852*, 2024b.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. In *International Conference on Learning Representations*, volume 2026, pp. 127293–127322, 2026b.

Rongyu Zhang, Menghang Dong, Yuan Zhang, Liang Heng, Xiaowei Chi, Gaole Dai, Li Du, Dan Wang, Yuan Du, and Shanghang Zhang. Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18764–18772, 2026c.

Yuying Zhang and Joni Pajarinen. Manipulate-to-navigate: Reinforcement learning with visual affordances and manipulability priors. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 90–96. IEEE, 2025.

Jia Zheng, Teli Ma, Yudong Fan, Zifan Wang, Shuo Yang, and Junwei Liang. Motionwam: Towards foundation world action models for real-time humanoid loco-manipulation. *arXiv preprint arXiv:2606.09215*, 2026.

Yifan Zhong, Xuchuan Huang, Ruochong Li, Ceyao Zhang, Zhang Chen, Tianrui Guan, Fanlian Zeng, Ka Nam Lui, Yuyao Ye, Yitao Liang, et al. Dexgraspvla: A vision-language-action framework towards general dexterous grasping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18836–18844, 2026.

A APPENDIX

A.1 TRAINING

M-VLA adopts a pretrained VLM backbone and followed with two-stage training pipeline: single-domain pre-training followed by mobile-manipulation post-training. This design exploits the functional split of the Dual Experts (§3.4): because the navigation expert and the manipulation expert have separate action projections and can operate independently, each expert can first absorb abundant single-domain data before the two are coupled on the scarcer mobile-manipulation demonstrations.

A.1.1 PRE-TRAINING

In the pre-training stage, the two functional experts are trained separately on their respective single-domain datasets such as Open X-Embodiment (O'Neill et al., 2024), while the shared VLM backbone and the Grounded Geometric Tokenizer (GGT) are updated jointly with whichever expert is active.

Manipulation pre-training. The manipulation expert is pre-trained on a large-scale fixed-base manipulation dataset, where the robot base remains stationary and only the arm and gripper are controlled. Each demonstration provides RGB observations, robot proprioception (joint positions, gripper state), and the corresponding arm action labels. Because no base motion occurs in this data, the navigation expert is frozen (or absent) and only the manipulation flow-matching head $L_{\text{fm}}^{\text{man}}$ is optimized. Since the fixed-base manipulation data does not involve any SLAM module (the robot does not move through the environment), SLAM-derived inputs are masked out: the validity bit is set to $v_t = 0$, and the occupancy-map tokens are replaced by a learnable mask embedding with `input_mask=False`. This ensures the manipulation expert learns to act from vision, language, and proprioception alone, without relying on localization signals that would be absent in fixed-base settings.

Navigation / locomotion pre-training. The navigation expert also could be trained on a locomotion-only dataset collected from a mobile base performing point-goal and language-goal navigation tasks. Each demonstration contains observations. Only the navigation flow-matching head $L_{\text{fm}}^{\text{nav}}$ is optimized; the manipulation expert is frozen. In this setting, SLAM information is provided, the occupancy tokens produced by OccEncoder, the geometric tokens from the continuous metric-state tokenizer, and the $\mathrm{SE}(2)$ operator token are all active ($v_t = 1$). The manipulation action tokens are masked from the loss so that the navigation expert does not receive gradients from manipulation labels. This stage lets the navigation expert learn obstacle avoidance, path planning, and spatial reasoning grounded in the metric occupancy map, without being distracted by arm control.

It is also possible to avoid the pre-training, just load the pre-trained manipulation base model, and then conduct the post-training process.

A.1.2 POST-TRAINING

In the post-training stage, both experts are jointly fine-tuned on the mobile-manipulation dataset, where the robot must navigate through the environment and manipulate objects in coordinated long-horizon tasks.

Joint mobile-manipulation training. The post-training data contains full mobile-manipulation demonstrations: the robot navigates across rooms, approaches target objects, and performs manipulation, with the complete observation suite including RGB, proprioception, SLAM occupancy maps, and robot localization. Both the navigation and manipulation flow-matching heads are active, and the total loss is:

$$L_{\text{post}} = L_{\text{fm}}^{\text{nav}} + L_{\text{fm}}^{\text{man}} \tag{7}$$

Crucially, the Coordination Bridge is now fully engaged: the navigation and manipulation token streams attend to each other at every layer of the expert blocks, enabling the two experts to learn coupled behaviors, such as the navigation expert learning to position the base where the arm can reach the target, and the manipulation expert learning to account for ongoing base motion. The VLM backbone, GGT, operator token MLP, and both expert heads are all updated end-to-end.

SLAM masking during post-training. Although the mobile-manipulation dataset provides SLAM information for all episodes, we continue to apply stochastic SLAM masking during post-training to maintain robustness.

This graduated masking ensures the model does not collapse when SLAM degrades at test time, while still learning to exploit geometric information when it is available.

A.2 Dataset

To support the two-stage training pipeline in §A.1, we construct a dataset that spans both simulation and real-world platforms. All demonstrations are collected with a whole-body teleoperation interface controlling the base, torso, arms, and grippers. Recorded observations include RGB (three cameras: head, left wrist, right wrist), joint proprioception, gripper state, and the action commands issued by the teleoperator. Where applicable, SLAM occupancy maps and localization estimates are synchronized to each frame.

A.2.1 Simulation dataset

Our simulation dataset is built on top of the BEHAVIOR-1K task suite in OmniGibson (Li et al., 2024). BEHAVIOR-1K provides photo-realistic multi-room scenes (kitchens, living rooms, bedrooms, patios) with rich object states, which are the ingredients needed for long-horizon mobile manipulation. All images are rendered at 224×224 resolution to match the vision backbone input. The simulation dataset as illustrated in Figure 5.

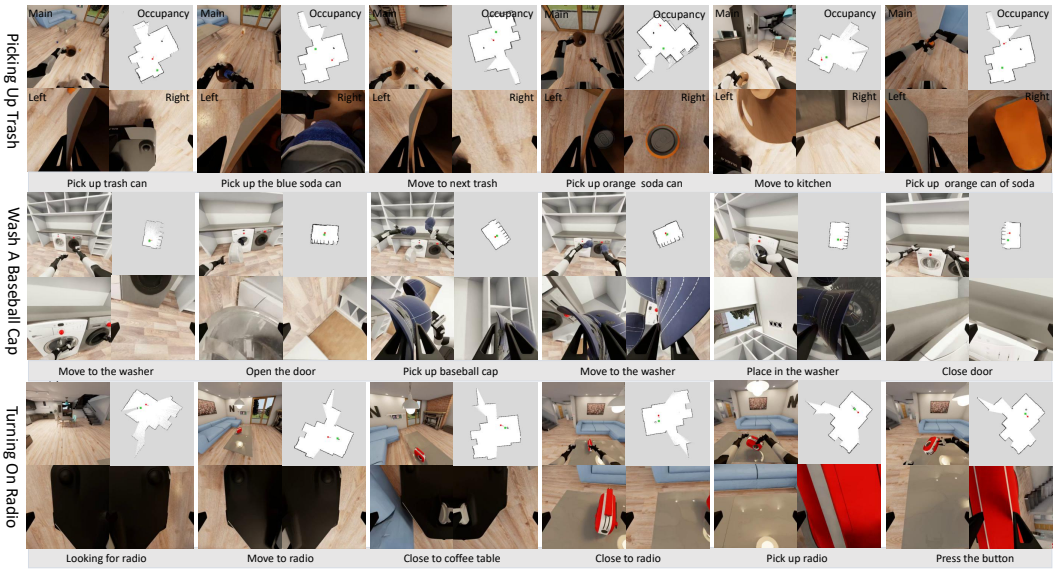


Figure 5: Simulation dataset

Teleoperation-based data collection. We build a whole-body VR teleoperation interface on top of the OmniGibson simulator. A human operator drives the R1Pro robot through each task using a VR headset for view control and a pair of 6-DoF hand controllers mapped to the two end-effectors; base velocity is commanded via joystick, and torso pitch/lift is bound to controller triggers. During each episode, we log:

- **Visual observations**: RGB and depth from head and both wrist cameras.
- **Proprioception**: joint positions/velocities, torso and gripper state.
- **Actions**: base linear/angular velocity, torso commands, dual-arm joints, and gripper open/close signals.
- **Task annotations**: language instruction, and subgoal timestamps.
- **Ground-truth pose**: the simulator's true base pose, used only for evaluation and as a reference to compute SLAM error, never fed to the model.

We collected 200 demonstrations per task across different tasks same as BEHAVIOR. Episode durations range from roughly 1 minute to over 14 minutes.

SLAM construction in simulation. Although OmniGibson exposes ground-truth pose, we deliberately do not use it as input to the policy. Instead, we run an online SLAM module in simulation to produce noisy, realistic occupancy maps and localization estimates that match what a real robot would encounter:

- **Sensing.** We simulate a 2D LiDAR (0.5° angular resolution, 20 m range) mounted on the robot base, together with wheel odometry perturbed by additive Gaussian noise (translational std 0.02 m/step, rotational std 0.5°/step) to emulate slip.
- **Mapping and localization.** We use a 2D graph-SLAM front-end (scan-matching against a rolling submap) with a pose-graph back-end that outputs, at each time step, the estimated pose $\hat{\xi}_t = (\hat{R}_t, \hat{p}_t)$, its covariance Σ_t, and a global 2D occupancy grid with 5 cm resolution.
- **Localization initialization.** At the start of each episode, the SLAM module is re-initialized at a random location within the scene, so that train and test episodes place the robot at different starting poses; the occupancy map is built online as the robot explores.
- **Temporal alignment via interpolation.** The SLAM front-end runs asynchronously and at a substantially lower rate than the 20 Hz observation stream (typically 5–10 Hz for map updates and 10–15 Hz for pose estimates, depending on scene complexity and LiDAR scan matching load). To feed the model with a per-frame, temporally aligned geometric input, we continuous-time interpolate the SLAM outputs to match observation frequency.

This design ensures that (1) the model is trained against imperfect maps and poses that resemble deployment conditions, and (2) the localization covariance Σ_t used by the trust scalar c_t (Eq. 2) is genuine SLAM uncertainty rather than a synthetic distribution.

A.2.2 Real-world dataset

The real-world dataset is collected on a physical bimanual mobile manipulator with the same kinematic layout (23 DoF: mecanum-wheeled base, torso lift and pitch, two 7-DoF arms, two parallel-jaw grippers). The sensor suite consists of a head-mounted camera and two wrist-mounted cameras, wheel encoders, an IMU, and a LiDAR on the base. The real-world robot dataset as illustrated in Figure 6.



Figure 6: Real-world robot dataset

Teleoperation setup. Real-world demonstrations are collected via a VR teleoperation the same as simulation data collection: the operator wears a VR headset streaming the head camera view and manipulates two 6-DoF haptic controllers that map to the end-effector targets of the two arms via inverse kinematics.

Tasks and scale. We collect real-world demonstrations across several environments covering both short single-station manipulation and difficult mobile-manipulation tasks. We recorded large scale real-world demonstrations, with per-task episode counts ranging from 200 to 1800 depending on task complexity, and durations from 30 seconds to roughly 8 minutes.

SLAM on the physical robot. On the real robot, occupancy maps and localization are produced by an on-board LiDAR SLAM stack running in real time; the output is the same tuple $(\hat{\xi}_t, \Sigma_t)$ consumed by GGT. Because different rooms are mapped in separate sessions, each episode carries

its own map-origin and coordinate basis, which is exposed to the OccEncoder as scalar conditioning so the model can reconcile per-episode reference frames.

Handling missing modalities. A subset of the real-world data was collected before the SLAM stack was deployed, and some manipulation-only recordings intentionally omit the LiDAR stream. For these episodes, we set the validity bit $v_t{=}0$ and rely on the placeholder embedding and mask tokens described in §3.3.2 and §A.1.1. This lets us mix SLAM-enabled and SLAM-free real-world data in the same batch without changing the model.

A.3 Real World Robot

We design several families of long-horizon tasks that mirror common service-robot duties and stress base–arm coupling end-to-end, we take two tasks as examples:

Package sorting. The robot navigates to the drop-off area, picks up delivered parcels, and transports and sorts them onto the correct destination region according to a language instruction (e.g. Carefully sort the pile of package.). Success requires the base to destination region at a pose from which the arm can both grasp the parcel and place it, so a good stopping pose is decisive.

Shelf Placement. Starting from drop-off area, the robot repeatedly approaches, picks up each object, and returns it to its designated container or location (e.g. box into the top shelf, object to the second shelf). This family emphasizes many short manipulation sub-goals chained after repeated re-approach, so localization drift accumulates over the several minutes of execution.

Both families require the robot to move between regions and manipulate at each, exercising the navigation expert and the manipulation expert jointly rather than in isolation.

A.4 From $\mathrm{SE}(3)$ to $\mathrm{SE}(2)$

A wheeled mobile base translates in the ground plane and rotates only about the vertical axis. In our implementation we therefore keep the vertical coordinate z constant ($z = 0$) and hold pitch and roll fixed, so the full rigid-body pose $\xi_t \in \mathrm{SE}(3)$ reduces to the planar pose $\xi_t = (R_t, p_t) \in \mathrm{SE}(2)$ used throughout the paper. The trust scalar, operator transport (Eq. 1), and grounded geometric feature (Eq. 3) are all defined in this planar setting. This is the appropriate regime for the wheeled bimanual platforms we study; extending the same operator-fusion idea to full $\mathrm{SE}(3)$ (e.g. legged or aerial bases) is left to future work, and would require replacing the planar transform and the two-variance covariance summary with their $\mathrm{SE}(3)$ counterparts.

A.5 Qualitative analysis.



Figure 7: Real world robot qualitative result.

Beyond the aggregate numbers, the per-trial rollouts reveal how the operator-based coupling changes behavior on the physical robot, where SLAM drift is genuine rather than injected. As shown in Figure 7 during the several minutes of Package Sorting, the on-board LiDAR SLAM accumulates pose error, yet M-VLA maintains a consistent moving–grasp–transport–place chain. Because world-frame intent is transported into the body frame through the $\mathrm{SE}(2)$ operator, a fixed physical configuration relative to the robot maps to the same body-frame input regardless of where the base stops, so a skill learned at one docking pose transfers to another without a pose-specific correction. When the trust scalar c_t drops under a degraded scan match, the policy visibly slows its approach and re-centers before committing to a grasp, rather than acting on an untrustworthy pose. The residual failures we observe are dominated by end-effector execution (a slipped grasp or a missed release) and by early termination on long ordered subgoal chains, rather than by gross base-placement errors.

A.6 SENSITIVITY TO THE REFERENCE NOISE SCALES σ_p, σ_ψ

The trust scalar c_t (Eq. 2) depends on the reference noise scales σ_p (metres) and σ_ψ (radians), which set the covariance magnitude at which localization is considered untrustworthy. We sweep each scale around the default ($\sigma_p = 0.1\,\mathrm{m}$, $\sigma_\psi = 5^\circ$) while holding the other fixed, and report average success rate on the living-space tasks (Table 2). Performance is stable across a broad range: varying σ_p over $[0.05, 0.2]\,\mathrm{m}$ and σ_ψ over $[2.5^\circ, 10^\circ]$ changes average success rate by at most 0.04 points. Only at the extreme settings, where c_t saturates to 1 regardless of covariance (large σ) or collapses to 0 under mild noise (small σ), does the trust signal lose its value and performance drop. This indicates the mechanism is not brittle to the exact choice of reference scale.

Table 2: Sensitivity of average living-space success rate to the trust-scalar reference scales. One scale is varied while the other is held at its default (**bold**).

σ_p (m)	0.05	**0.10**	0.15	0.20	extreme (1.0)
Success Rate	0.32	**0.32**	0.32	0.30	0.30
σ_ψ (deg)	2.5	**5.0**	7.5	10	extreme (45)
Success Rate	0.30	**0.32**	0.32	0.30	0.28

A.7 ABLATION STUDY ON CONCAT

Most informative for our central claim is the comparison to **M-VLA-concat**: with the identical SLAM map and pose stream but the pose supplied as a concatenated feature rather than an $\mathrm{SE}(2)$ operator, average success drops from 0.32 to 0.28. Because the two models differ only in how localization enters, this gap is attributable to operator-based fusion, not to access to geometry, indicating that jointly learning base suitability through the Coordination Bridge outperforms an externally computed, sequentially applied prior. These results suggest M-VLA reasons about motion and manipulation in their respective frames (world/map vs. body) and reduces the error accumulation that otherwise breaks long-horizon execution. The operator-vs-feature gap persists and in fact widens in relative terms on these longer-horizon tasks, consistent with our thesis that a geometrically consistent frame transform matters most when drift has more time to accumulate.

Table 3: **Living Space results.** Absolute success rate per task; **Avg.** reports the mean over tasks on the trials.

Method	**Avg. Success Rate**
M-VLA-concat	0.28
M-VLA	0.32

A.8 ADDITIONAL EXPERIMENTAL DETAILS

All experiments are trained on 16 NVIDIA H200 and 16 NVIDIA A100 GPUs, tested on 5 NVIDIA RTX 4090, using a cosine decay scheduler. We use 3B SigLIP+Gemma as our VLM backbone. For all multi-task pre-training, we use 80k training steps, with a learning rate of 2.5×10^{-5}. We also fine-tune our method for 30k steps with a learning rate of 3.5×10^{-6}.

Evaluation protocol. Unless stated otherwise, every reported success rate is measured over 10 rollouts per task and averaged over official BEHAVIOR requirements. Success Rate is the fraction of trials in which the full task is completed. Progress is the fraction of task subgoals completed for real robot. To avoid the ambiguity of relative percentages over near-zero baselines, all results report absolute success rates; we quote relative improvements only as supplementary context.

Other Baselines. We compare against three recent mobile-manipulation VLAs that can train and evaluate on our dataset ($\pi 0.5$ (Intelligence et al., 2025b), Task-VLA (Larchenko et al., 2025), and StarVLA-π (Community, 2026)). To isolate the effect of our design choices rather than mere access

to geometry, we additionally build a controlled variants of our own model that receive exactly the same SLAM inputs as M-VLA: **M-VLA-concat**, which removes Localization-as-Operator Fusion and instead concatenates the raw pose $(p_x, p_y, \sin\psi_t, \cos\psi_t)$ as an ordinary input feature, so the network must rediscover the world→body transform on its own.

B MOBILE / LOCOMOTION MANIPULATION RELATED WORKS

Mobilization manipulability priors (Shao et al., 2025a; Lin et al., 2026; Chai et al., 2025; Yang et al., 2025; Zhang & Pajarinen, 2025) answer the "where to stand" question with an explicit geometric prior: an externally computed reachability or manipulability field that a planner queries to select a base pose before manipulation begins. Our approach differs in two ways. First, we do not compute an explicit reachability map; instead the navigation expert learns the map-frame suitability of base poses end-to-end from the reward of successful downstream manipulation, and localization enters as an $\mathrm{SE}(2)$ operator that keeps the two experts' frames consistent rather than as a precomputed prior. Second, these base-placement methods typically decouple placement and manipulation into sequential stages, whereas our Coordination Bridge lets the two experts exchange intent at every layer, so base placement and arm action are decided jointly.

B.1 FAILURE CASES

Case 1 Unreliable end-effector execution, or a physically irrecoverable state after a single failed attempt. The robot cannot reliably pick up or release objects. Dexterity is often the trigger; Reachability and Catastrophic error are cascade consequences of that first failure.

Case 2 Many tasks have a hard order constraint. Another common issue is deciding to finish early, exacerbated by "tails" on passive actions in the demo dataset. Failure when the robot should choose a locally non-obvious action, e.g. go around the table instead of trying to reach through it.

Case 3 Navigating to fixed locations seen in the demos is generally fine; in rare cases the model confuses the task and goes to another task's location.

Case 4 The robot lacks awareness that an arm may be stuck, which prevents further safe navigation.

Case 5 The cameras do not cover the floor in front of the robot.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or sensitive personal data. We follow standard research and safety practices for robotic experiments and ensure that real-robot evaluations are conducted in controlled environments to minimize potential risks to people and property.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of our method. We will release the source code, dataset upon publication, subject to any applicable dataset and licensing restrictions.