LINGUAMOTION: EXPRESSING THE LANGUAGE OF MOTION FOR REAL-WORLD ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Translating high-level instructions into manipulations is a fundamental capability in robotics, yet achieving this across diverse tasks and embodiments in real world remains challenging. While Vision-Language Models (VLMs) offer rich world knowledge, existing VLM-based policies often couple action prediction to specific robot kinematics and action spaces, failing to fully leverage pretrained knowledge and limiting the generality. We introduce **LinguaMotion**, a novel framework reinterpreting manipulation as structured generation in the language of motion. Instead of low-level robot actions, our model represents manipulation via object-centric motion directly within the native language space of VLMs. Given an image and instruction, it generates a hierarchical description of task-relevant objects, their spatial states, and how they should evolve to accomplish the task, thus bridging high-level semantic understanding with physical motion planning. To support this formulation, we further introduce a *numeric value loss* to restore geometric fidelity in tokenized motion representation, and a *cycle-consistent learning framework* to enforce semantic alignment between motion and language while enabling inference-time self-verification. On challenging real-world manipulation tasks, LinguaMotion consistently outperforms mainstream generalizable baselines across embodiments.

1 INTRODUCTION

Manipulation can be viewed as translating abstract intent into concrete interactions in the physical world. Humans approach this process by organizing a scene around objects, inferring their relations, and anticipating how they should evolve to achieve the goal, thereby seamlessly generalizing across tasks and environments. Replicating such ability in robotic systems, however, remains a central challenge (Liu et al., 2025; Xu et al., 2024b). Real-world manipulation is inherently open-ended, requiring models to operate robustly across diverse instructions, objects, scenes, and embodiments, often under limited and heterogeneous real robot supervision (Ma et al., 2026; Li et al., 2025c).

Recent advances in foundation models have opened new avenues for tackling this challenge. In particular, general world knowledge in Vision-Language Models (VLMs) (Zhang et al., 2024; Hurst et al., 2024; Comanici et al., 2025) enables robots to interpret task instructions, reason about complex scenes, and guide robot execution. Building on this, Vision-Language-Action models (Kim et al., 2024; Black et al., 2024; Zitkovich et al., 2023) fine-tune VLMs to directly predict robot actions. Despite their promise, these methods typically couple action representation with specific robot embodiments (Intelligence et al., 2025; Team et al., 2025). Therefore, the policy becomes tied to particular kinematics and control signals, restricting its applicability across diverse embodiments and environments in real-world settings (Ma et al., 2026).

This issue points to a deeper question beyond model or data scale: how manipulation is represented. Instead of embodiment-specific actions, recent work reinterprets manipulation via how objects should evolve to accomplish a task. These methods thus design various object-centric representations, ranging from sparse keypoints (Wang et al., 2025; Levy et al., 2025; Hsu et al., 2025) to dense flows (Xu et al., 2024a; Lee et al., 2025b; Zhi et al., 2025). Predicted from language instructions, these representations are subsequently translated into robot execution via downstream action planners (Sundaralingam et al., 2023; Fang et al., 2022). While decoupled from embodiment, current methods often train separate motion decoders that steer vision-language features to a new output domain. This

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.

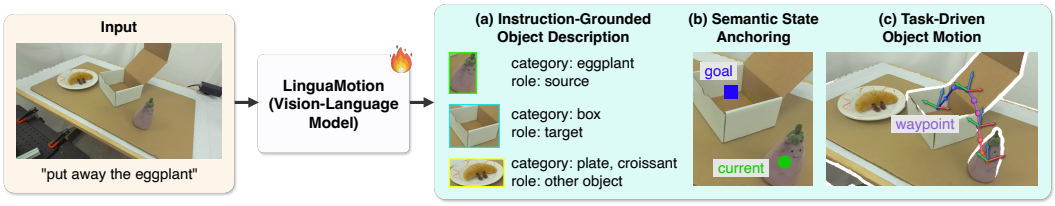


Figure 1: **Overview of LinguaMotion**. Given an image and instruction, we generate three components sequentially (§3.1): **(a) Instruction-Grounded Object Descriptions**, which identify task-relevant entities and their semantic roles, establishing *what* objects are involved, **(b) Semantic State Anchoring**, which localizes each manipulated object and infers its goal state in space when the task is realized, and **(c) Task-Driven Object Motion**, which generates a sequence of object positions and $\mathrm{SO}(3)$ orientations for achieving the goal, specifying *how* objects should evolve to complete the task. By modeling all components within a unified VLM (§3.2), LinguaMotion translates vision-language understanding internally into object motion, leveraging the rich pretrained knowledge for real-world robot manipulation across embodiments and environments.

disconnect prevents the model from fully exploiting the established vision-language alignment and rich prior in large pretrained backbones, limiting zero-shot capabilities across tasks.

In this paper, we introduce **LinguaMotion**, a unified framework bridging representations of object motion and pretrained VLMs. Our core insight is to express manipulation in the native language of VLMs, thereby jointly modeling instruction understanding, spatial grounding, and motion prediction. Specifically, conditioned on visual observations and instructions, LinguaMotion generates a hierarchical, object-centric description for the task, as shown in Figure 1: we first identify task-relevant objects with their semantic roles, then estimate current and goal states of the manipulated objects, and finally predict goal-directed object motion as sequences of 3D positions and $\mathrm{SO}(3)$ orientations for realizing the task. This formulation mirrors the reasoning progression underlying manipulation, translating high-level semantic understanding into spatial grounding and goal-directed object motion.

While casting manipulation as structured generation in VLMs brings geometry and semantics into a shared space for joint modeling, it also raises new challenges. ❶ While VLMs excel at sequence modeling, representing spatial quantities as tokens obscures their numerical structure (Shah & Aamodt, 2023; Zausinger et al., 2024; Stewart et al., 2023; Lukasik et al., 2025), leading to a mismatch between token-level supervision and geometric accuracy. We bridge this gap with a *numeric value loss* on motion predictions, extending the regression supervision to complete numerical values rather than independent digits. Specifically, per-digit predictions are aggregated into multi-digit numbers and directly regressed against ground-truth values, allowing the model to capture place-value structure and recover the geometric meaning lost in tokenization. This aligns token-level supervision with geometric fidelity of the predicted motion, while remaining compatible with autoregressive generation. ❷ Beyond geometric accuracy, preserving semantics in motion generation presents a key challenge. A motion that completes the task should retain the instruction's original intent, rather than relying on spurious correlations. Interestingly, our unified representation enables such reversibility. We instantiate this idea with a *cycle-consistent learning framework*. Specifically, the model is trained to recover the instruction from a predicted motion sequence, which in turn conditions a second motion prediction, forming a closed loop that enforces self-consistency within the shared representation. This encourages the predicted motion to preserve necessary task-relevant information, as only motions faithfully capturing the underlying intent can be reliably reconstructed. Moreover, deviations in this cycle provide an intrinsic signal for assessing prediction reliability, enabling self-verification during inference without external supervision.

We evaluate LinguaMotion on various real-world manipulation tasks probing language grounding, spatial reasoning, and multi-object interaction under a challenging zero-shot setting, without any task-specific fine-tuning. Experiments on Franka Research 3 and Agilex Piper robot arms demonstrate that LinguaMotion consistently outperforms mainstream generalizable manipulation baselines.

2 Related Work

Foundation Models for Robot Manipulation. Zero-shot robot manipulation across novel objects, scenes, instructions, and embodiments in the real world remains challenging due to limited and hetero-

geneous robot data (Liu et al., 2025; Ma et al., 2026). This motivates leveraging the comprehensive knowledge from foundation models. One line of work introduces intermediate conditions produced by large vision-language or image generation models, such as spatial constraints (Huang et al., 2024b;a; Yuan et al., 2024b; Tang et al., 2025), value maps (Huang et al., 2023), and goal images (Chen et al., 2025c; Zhou et al., 2025b). However, these conditions remain indirect abstractions that require downstream optimization to derive precise trajectories, creating a fundamental disconnect between high-level task understanding and the geometric precision required for physical execution (Xu et al., 2024b). A complementary direction repurposes pretrained video generative models to synthesize task-solving rollouts, from which object motion can be easily extracted for robot execution (Li et al., 2025b; Bharadhwaj et al., 2024a; Patel et al., 2025; Ni et al., 2025; Chen et al., 2025a; Li et al., 2025a; Dharmarajan et al., 2025). While benefiting from the scale and diversity of generative priors, these approaches derive motion from full pixel-space generation, where capacity is largely devoted to appearance synthesis rather than structured motion (Li et al., 2025c). This indirect formulation often leads to hallucinated motion and physical inconsistencies in pixel space. In contrast, our approach expresses object-centric motion directly within the native representation space of vision-language models, thereby jointly modeling language understanding, spatial grounding, and motion prediction.

Object-Centric Motion as Manipulation Interface. Building on foundation models, a recent line of research focuses on designing intermediate representations that bridge high-level task understanding and low-level control. In particular, object-centric motion has emerged as a natural interface for manipulation. Early work designs sparse keypoint features for imitation learning (Wang et al., 2025; Levy et al., 2025; Haldar & Pinto, 2025; Hsu et al., 2025), where semantic points (Oquab et al., 2023; Tschannen et al., 2025) on task-relevant objects are used as observations. Such representations enable generalization to novel object poses and appearances from limited demonstrations. More recent approaches such as Im2Flow2Act (Xu et al., 2024a) and General Flow (Yuan et al., 2024a) model object flow as a transferable manipulation interface, while TraceGen (Lee et al., 2025b) and 3DFlowAction (Zhi et al., 2025) advance this paradigm by adopting generative models for trajectory prediction. These approaches bypass the redundant effort of pixel-level generation in video models, but still rely on dedicated motion decoders to map pretrained vision-language features into a separate motion space. As a result, while the interface itself is well-suited for manipulation, it fails to fully inherit the prior knowledge in pretrained vision-language backbones. In contrast, we retain the object motion interface entirely within the native representation space of vision-language models, mitigating the output domain gap and leveraging pretrained capabilities for structured motion prediction.

Adapting Vision-Language Models for Robotics. A parallel line of work adapts Vision-Language Models (VLMs) for robotic control. The widely adopted paradigm, often referred to as Vision-Language-Action (VLA) models, fine-tunes a VLM to predict robot control signals such as end-effector poses or joint actions (Intelligence et al., 2026; Shen et al., 2025; Shi et al., 2025; Wang et al., 2026; Qu et al., 2025). To improve planning ability, recent variants introduce richer intermediate outputs, including depth (Wu et al., 2026b; Yu et al., 2026), end-effector path (Lee et al., 2025a; Li et al., 2025d; Zheng et al., 2024; Zhou et al., 2025a), and chain-of-thought reasoning (Huang et al., 2025; 2026; Ye et al., 2025; Zhao et al., 2025; Chen et al., 2025b), before producing final actions. Yet, a recurring challenge is that adapting VLMs to robot control data tends to degrade their pretrained multi-modal understanding capabilities, limiting generalization beyond the fine-tuning data distribution. Subsequent work mitigates this by aligning action prediction with the VLM's feature space, for example, by expressing low-level robot actions as language or through multi-modal co-training (Hancock et al., 2025; Zhang et al., 2026; Driess et al., 2025; Grover et al., 2025; Zhang et al., 2025; Bu et al., 2025; Zhou et al., 2026). However, the robot action is inherently embodiment-specific, tying the policy to particular kinematics and limiting cross-embodiment application. In contrast, we model object motion within VLM's native representation space, leveraging vision-language understanding while maintaining an embodiment-agnostic interface for manipulation.

3 METHOD

We introduce **LinguaMotion**, a unified framework that brings object motion into the representation space of Vision-Language Models (VLMs). We first present a hierarchical description of object identities, spatial states, and desired motion for a manipulation task (§3.1), and realize this representation directly within a VLM (§3.2). To ensure geometric accuracy, we then introduce a numeric value loss that restores metric meaning in the tokenized motion representation (§3.3). Moreover, we

encourage semantic consistency via bidirectional learning between instruction and motion, enabling more faithful and reliable motion generation (§3.4).

3.1 Hierarchical Representation

Robotic manipulation tasks can be understood via three aspects in a scene: which objects are relevant, where they are, and how they change. As shown in Figure 1, we express this reasoning progression as hierarchical predictions over object identities, spatial states, and task-directed motion.

Instruction-Grounded Object Description. Given an image $\mathbf{I}$ and a language instruction l, we describe a set of task-relevant objects as $\mathcal{O}(\mathbf{I}, l) = \{(b^{(i)}, c^{(i)}, r^{(i)})\}_{i=1}^{N}$, where $b^{(i)} \in \mathbb{R}^4$ denotes the 2D bounding box of the i-th object, $c^{(i)}$ its semantic category (*e.g.*, 'cup'), and $r^{(i)}$ its task-specific semantic role, including the `source` object being manipulated, the `target` implied by the instruction, and `object` for other entities. Here, a `target` may refer to a distinct object, a part of the `source` object, or a spatial location. This step grounds the instruction in the scene by jointly resolving object identity and roles, establishing the entities for subsequent spatial modeling.

Semantic State Anchoring. For each `source` object in $\mathcal{O}$, we extract clustered semantic keypoints (Oquab et al., 2023) and estimate $\mathrm{SO}(3)$ orientation (Hsu et al., 2025; Bharadhwaj et al., 2024b) from dense tracking to characterize its spatial state. We represent the current and goal states as $\mathcal{S}_c = (p_c, R_c)$ and $\mathcal{S}_g = (p_g, R_g)$, respectively, where $p_c, p_g \in \mathbb{R}^3$ denote the current and goal keypoint positions, and $R_c, R_g \in \mathrm{SO}(3)$ denote the corresponding orientations. The current state $\mathcal{S}_c$ anchors the object to the visual observation by capturing its initial spatial configuration, while the goal state $\mathcal{S}_g$ anchors the instruction to a concrete target configuration, specifying where and how the object should be manipulated upon task completion. Jointly modeling both states therefore grounds semantic intent into scene geometry, providing explicit spatial guidance for motion modeling.

Task-Driven Object Motion. Anchored on the initial and goal states, we derive object motion as a sequence of intermediate waypoints with 3D positions and orientations: $\mathbf{W} = \big((p_1, R_1), \ldots, (p_K, R_K)\big)$, where $p_k \in \mathbb{R}^3$ and $R_k \in \mathrm{SO}(3)$ represent the spatial pose at step k. Combining these waypoints with the endpoint states yields a complete, structured representation for each object: $\big(p_c, R_c, p_g, R_g, \mathbf{W}\big)$, where the state anchors (p_c, R_c) and (p_g, R_g) specify the initial and target configurations, and $\mathbf{W}$ describes how the object pose evolves over time. As shown in Figure 1, integrating these components across objects leads to our hierarchical representation:

$$\mathcal{H}(\mathbf{I}, l) = (\mathcal{O}, \mathcal{T}) = \{(b, c, r),\ (p_c, R_c, p_g, R_g, \mathbf{W})\}, \tag{1}$$

which jointly captures object identities and roles, spatial state anchors, and full 6D pose motion trajectories, providing a compact, geometrically grounded, and embodiment-agnostic representation.

3.2 Motion Modeling via Native Representation of Vision-Language Models

Existing methods for instruction-guided object motion prediction (Xu et al., 2024a; Yuan et al., 2024a; Lee et al., 2025b; Zhi et al., 2025) typically rely on dedicated expert decoders, implemented by diffusion or flow-matching models, upon pretrained backbones such as vision encoders, text encoders, or Vision-Language Models (Oquab et al., 2023; Tschannen et al., 2025; Raffel et al., 2020; Karamcheti et al., 2024). While effective in trajectory prediction, these decoders operate in a separate output space decoupled from the native representation of the backbone, limiting the ability to fully leverage pretrained knowledge and often hindering zero-shot application.

To address this, we propose to model object motion using the native language of VLMs. We cast manipulation as structured generation and fine-tune a pretrained VLM f_θ to predict the hierarchical description defined in §3.1: $\mathcal{H}(\mathbf{I}, l) = f_\theta(\mathbf{I}, l)$, where objects, spatial states, and task-directed motion are represented as sequences of text tokens, thereby unifying perception, understanding, and motion modeling in a single representation space.

3.3 Restoring Geometric Fidelity in Tokenized Motion Representation

Yet, bringing object state and motion modeling into VLM's language introduces a critical challenge: spatial coordinates and rotations lose their numeric meaning when serialized into digits, as shown in Figure 2 (Left). With cross-entropy loss, each digit is treated as a categorical symbol, so predictions

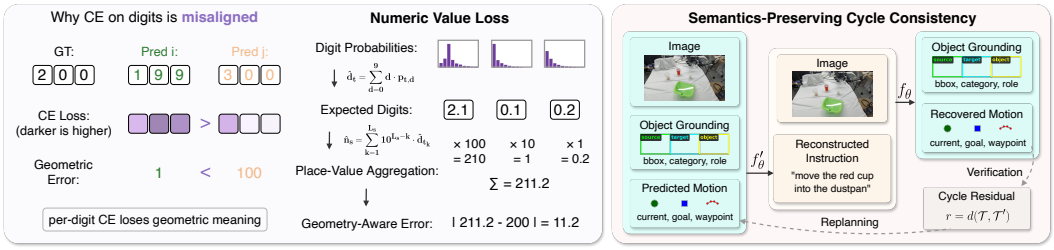


Figure 2: **Overview of the key designs in LinguaMotion.** Our method improves both geometric fidelity and semantic alignment for reliable motion prediction. **(Left) Numeric Value Loss.** Standard cross-entropy on digit tokens is misaligned with geometric error, as it ignores place-value structure and fails to capture numeric meaning. We address this by supervising complete numeric values via place-value aggregation, restoring geometric accuracy of the tokenized motion (§3.3). **(Right) Semantics-Preserving Cycle Consistency.** We enforce consistency between object motion and instruction through bidirectional learning (§3.4), where recovered motion can be evaluated by cycle residual, enabling semantic alignment as well as inference-time verification and re-planning.

that are numerically close and far from the target may incur disproportionate penalties. To address this, we take inspiration from numerical regression (Shah & Aamodt, 2023; Zausinger et al., 2024; Stewart et al., 2023; Lukasik et al., 2025) and extend the regression supervision directly to complete numeric values. This preserves the compositional structure of multi-digit numbers and better aligns training loss with geometric error, as illustrated in Figure 2 (Left). Specifically, let $p_{t,d}$ denote the predicted probability of digit d at position t, and define the expected digit as $\hat{d}_t = \sum_{d=0}^{9} d \cdot p_{t,d}$. For each maximal digit span s of length L_s with positions $t_1, \ldots, t_{L_s}$, we recover the expected number via place-value aggregation and define the numeric value loss via ℓ_1 regression:

$$\hat{n}_s = \sum_{k=1}^{L_s} 10^{L_s-k} \cdot \hat{d}_{t_k}, \quad \mathcal{L}_{\text{nv}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{|\hat{n}_s - n_s^\star|}{Z}, \tag{2}$$

where $n_s^\star$ is the ground-truth value, $\mathcal{S}$ is the set of numeric spans, and Z is the numeric range for normalization. Importantly, place-value awareness emerges from the gradient, as $\partial \hat{n}_s / \partial p_{t_k,d} \propto 10^{L_s-k}$, assigning larger updates to higher-order digits in accordance with their geometric significance. Thus, this loss restores metric meaning while remaining compatible with autoregressive prediction. The overall objective combines cross-entropy with the numeric value loss: $\mathcal{L}_{\text{fwd}} = \mathcal{L}_{\text{ce}} + \lambda \mathcal{L}_{\text{nv}}$, where $\mathcal{L}_{\text{nv}}$ is applied only to numerical tokens, and all other tokens are supervised with cross-entropy.

3.4 Semantics-Preserving Cycle Consistency

Unifying language and motion within a shared representation also suggests a mutual constraint: a valid motion sequence that faithfully solves the manipulation task should preserve the semantics of the instruction it realizes. We operationalize this intuition through cycle consistency shown in Figure 2 (Right). Because both object-centric motion and instructions are expressed in the same representation space, they can be treated as conditioning signals for each other within a single generative model. Consequently, the forward mapping $(\mathbf{I}, \ell) \to \mathcal{H}$ and the inverse mapping $(\mathbf{I}, \mathcal{H}) \to \ell$ form a closed cycle, in which each direction enforces the other to consistently retain task-relevant semantics.

Bidirectional Learning. We instantiate this principle by training the model in both generation directions. The forward task predicts motion given the observation and instruction, $\mathcal{H} = f_\theta(\mathbf{I}, \ell)$, while the inverse task reconstructs the instruction from the observation and hierarchical description, $\ell' = f'_\theta(\mathbf{I}, \mathcal{H})$. Both tasks share parameters of a single model, with a prefix (`[INSTR_TO_MOTION]` or `[MOTION_TO_INSTR]`) indicating the generation direction. The overall objective combines both directions: $\mathcal{L} = \mathcal{L}_{\text{fwd}} + \mathcal{L}_{\text{inv}}$, where $\mathcal{L}_{\text{inv}}$ is cross-entropy over the recovered instruction tokens, encouraging motion to capture the original task intent rather than relying on spurious correlations.

Inference-Time Self-Verification. A practical challenge at deployment is knowing when the generated motion can be trusted. Our bidirectional formulation offers a natural signal: since motion generation and instruction reconstruction share the same model, we can evaluate internal consistency through cyclic generation, without any external supervision. Starting from a predicted $\mathcal{H} = (\mathcal{O}, \mathcal{T})$, we perform a round trip $\mathcal{H}' = f_\theta(\mathbf{I}, f'_\theta(\mathbf{I}, \mathcal{H})) = (\mathcal{O}', \mathcal{T}')$, and define the *cycle residual* $r = d(\mathcal{T}, \mathcal{T}')$

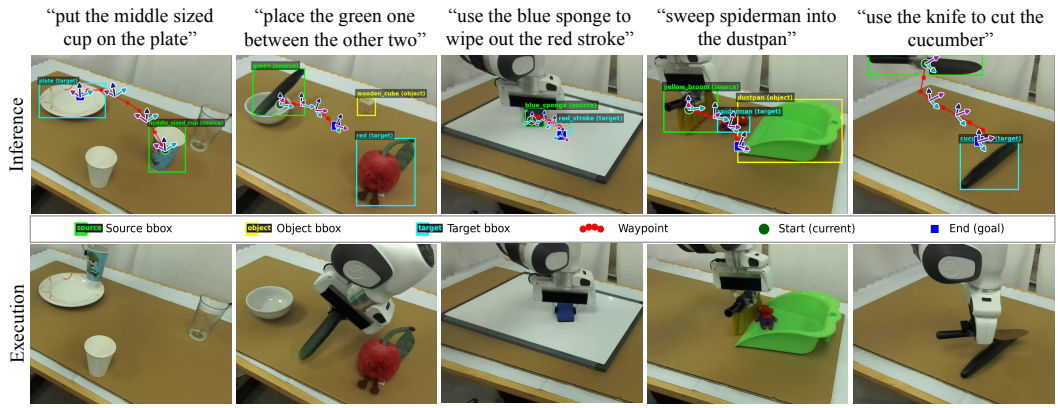


Figure 3: Zero-shot results of LinguaMotion on real-world manipulation with Franka Research 3.

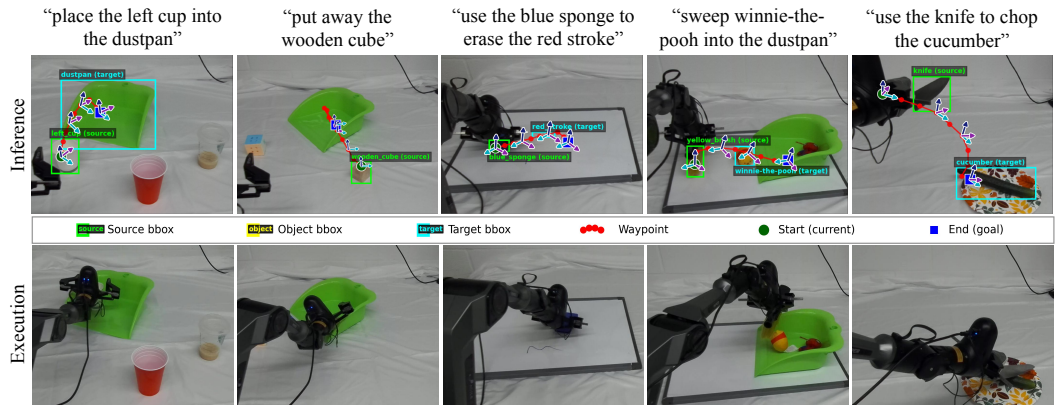


Figure 4: Zero-shot results of LinguaMotion on real-world manipulation with Agilex Piper.

as the normalized displacement and rotation error between the original and reconstructed motion. A small residual indicates that $\mathcal{T}$ preserves sufficient semantic information to recover an instruction that reproduces it faithfully, while a large residual indicates a breakdown in semantic consistency.

4 Experiments

4.1 Experimental Setup

Dataset Curation. We construct our training data from two widely used real-world robot datasets, BridgeV2 (Walke et al., 2023) and DROID (Khazatsky et al., 2024). Following the preprocessing protocol in TraceGen (Lee et al., 2025b), we select 17K episodes from BridgeV2 and 13K from DROID, yielding a total of 30K episodes, of which 5% are randomly held out as the test set for evaluation. We further extract multiple training samples from each episode by shifting the starting frame with a temporal stride of 3. This results in ∼150K $(\mathbf{I}, l, \mathcal{H})$ triples for training. Additional details of the data construction pipeline are provided in Appendix §A.

Implementation. We fine-tune Qwen3.5-4B (Qwen Team, 2026) using LoRA (Hu et al., 2021) on all linear layers in the vision encoder and language model with rank $r = 16$ and $\alpha = 32$. This results in 30.3 M trainable parameters (0.66% of the VLM backbone). We train for 10 epochs on 4 NVIDIA RTX 6000 Ada GPUs (per-GPU batch size 4, gradient accumulation 2) using AdamW (Loshchilov & Hutter, 2017) with learning rate 2×10^{-5}. For bidirectional learning, we switch the training objective to the inverse task $(\mathbf{I}, \mathcal{H}) \rightarrow \ell$ (§3.4) with probability 0.2 for each training sample. The training objective combines next-token cross-entropy with the numeric value loss (§3.3), with $\lambda = 10$.

Baselines. We compare LinguaMotion against mainstream baselines capable of zero-shot application across both tasks and embodiments. These include: **(i)** object-centric motion generation models (3DFlowAction (Zhi et al., 2025), TraceGen (Lee et al., 2025b)), which directly generate dense object trajectories via dedicated decoders, **(ii)** video-generation-based methods (AVDC (Ko et al., 2023),

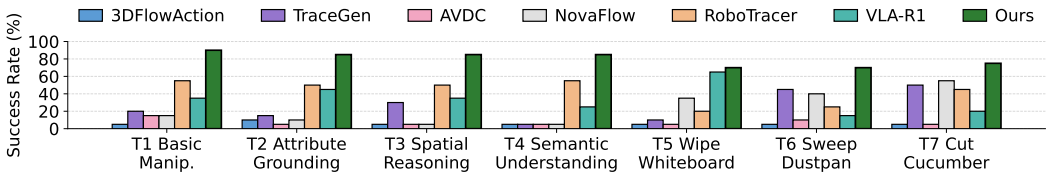


Figure 5: Zero-shot success rates on real-world manipulation tasks with Franka Research 3.

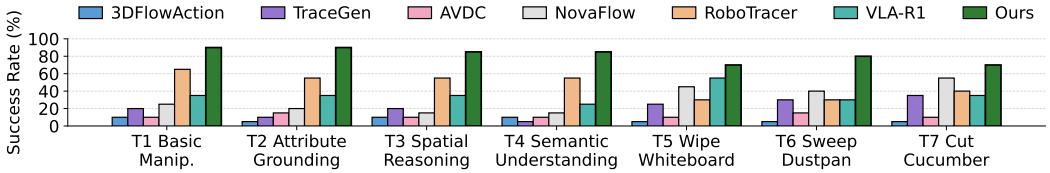


Figure 6: Zero-shot success rates on real-world manipulation tasks with Agilex Piper.

NovaFlow (Li et al., 2025a) with Wan 2.2 (Wan et al., 2025)), which generate task-solving videos and extract the object motion, and **(iii)** vision-language-model-based methods (RoboTracer (Zhou et al., 2025a), VLA-R1 (Ye et al., 2025)), which trace translational paths for objects or end-effectors based on the instructions. All baselines and our method are evaluated under identical robot setups.

Real-World Robot Deployment. We perform real-world manipulation on two distinct robotic arms, Franka Research 3 and Agilex Piper, both equipped with a third-person view ZED 2i camera for visual observations. Both the baselines and our method predict motion from an image and language instruction, then pass it to a downstream planner for robot execution. Following literature convention (Zhi et al., 2025; Dharmarajan et al., 2025; Li et al., 2025a; Zhou et al., 2025a), grasp poses for relevant objects are provided by AnyGrasp (Fang et al., 2022), a state-of-the-art grasp detection model. Conditioned on the grasp pose and predicted motion, robot actions are planned using cuRobo (Sundaralingam et al., 2023), a GPU-accelerated trajectory optimization library. This deployment stack is applied uniformly to all baselines and our method to ensure fair evaluation.

Evaluation Protocol. Under our zero-shot setting, all methods are deployed on real robots without additional task- or embodiment-specific fine-tuning. This challenging setting requires generalization across diverse embodiments, instructions, objects, and scene environments. For each task, we perform 20 trials under diverse initial conditions and report the average success rates on each robot arm.

4.2 Zero-Shot Performance on Real-World Robotic Manipulation

Tasks. We evaluate on a range of tabletop manipulation tasks that probe language grounding, spatial reasoning, and physical multi-object interaction. **(T1)** Basic Manipulation. We begin with basic manipulation skills, where the instruction directly specifies both the object and the target location (*e.g.*, "put the bubble tea on the plate"). **(T2)** Attribute and Relation Grounding. We consider a more challenging setting in which the object is specified indirectly via attributes or relations, such as color, relative size, or spatial context (*e.g.*, "the green one", "the middle-sized cup", or "the left toy of the three"). **(T3)** Spatial Reasoning. We evaluate instructions in which the target is defined relationally, such as placing an object "between two others" or "next to the bowl", thereby requiring spatial reasoning across multiple entities. **(T4)** Semantic Task Understanding. To test higher-level understanding, we include tasks with implicit goals, such as "put away the doll", which requires inferring the appropriate target in the scene (*e.g.*, place it into a box). **(T5-T7)** Tool Use. We evaluate manipulation tasks involving tools, including wiping a whiteboard with an eraser (T5), sweeping objects into a dustpan with a broom (T6), and cutting a cucumber with a knife (T7). These tasks require reasoning about object affordances and coordinating tool-object interactions.

Results. Figures 3 and 4 present qualitative zero-shot manipulation results of LinguaMotion on real Franka Research 3 and Agilex Piper robots, respectively. The top row visualizes predicted bounding box, category, role, and motion, and the bottom row shows real-robot execution results based on the predictions. The basic manipulation (T1), such as pick-and-place, underlies most demonstrated behaviors. For attribute- and relation-based grounding (T2), the model resolves referring expressions based on visual attributes (*e.g.*, "middle sized cup", "the green one") and relational cues (*e.g.*, "the

left cup"), even in cluttered real-world scenes. For spatial reasoning (T3), the model infers spatial relationships among multiple objects (*e.g.*, "between the other two") and produces motion trajectories that satisfy these constraints. Notably, the model generalizes to higher-level semantic instructions (T4), such as "put away the wooden cube", where the target (dustpan) is not explicitly specified but inferred from the scene context. It also shows capability in tool-use scenarios (T5-T7), including wiping a board with an eraser, sweeping objects into a dustpan, and cutting with a knife. These tasks require understanding object affordances and coordinated tool-object interactions. Despite differences in objects, scenes, instructions, and embodiments, LinguaMotion produces accurate object grounding and coherent goal-directed motion. Full demonstration videos are in the supplementary material.

Figures 5 and 6 report per-task success rates on Franka Research 3 and Agilex Piper, respectively, where our LinguaMotion achieves the best performance with a clear margin over most baselines. Specifically, object motion generation methods (3DFlowAction, TraceGen) and video generation approaches (AVDC, NovaFlow) perform reasonably well on straightforward settings but degrade noticeably on tasks involving implicit reasoning (T2-T4). In these scenarios, they struggle to resolve ambiguous references and are sensitive to instruction variation, leading to brittle behavior. Interestingly, TraceGen and NovaFlow remain competitive on tool-use tasks (T5-T7), where object and motion patterns are more stereotyped, reducing difficulty in semantic grounding. Moreover, RoboTracer and VLA-R1 better understand high-level task intent, but often face difficulty converting abstract translation paths into precise robot actions, resulting in consistent yet suboptimal performance. In contrast, our method reliably grounds instructions into object-centric representations and produces coherent motion across diverse tasks on both embodiments.

4.3 Cycle Consistency for Inference-Time Verification

Given predicted motion $\mathcal{T}$, we compute the cycle residual $r = d(\mathcal{T}, \mathcal{T}')$ as defined in §3.4, which can serve as an inference-time signal for assessing the prediction quality.

Impact on Real-Robot Experiments. We evaluate the effect of cycle consistency as a verification signal at real deployment. For each instruction, we generate motion $\mathcal{T}$ and compute its cycle residual r. If r exceeds a threshold, we resample a new motion sequence and repeat the process, up to $N = 3$ attempts. We evaluate this strategy against direct execution of the first prediction on two representative tasks: T4 (semantic task understanding) and T5 (tool-use, wiping), each with 20 trials. Applying the verification strategy consistently improves success rates on both tasks and robots, as shown in Figure 7, supporting the cycle residual as an internal signal for reliable real-world deployment.

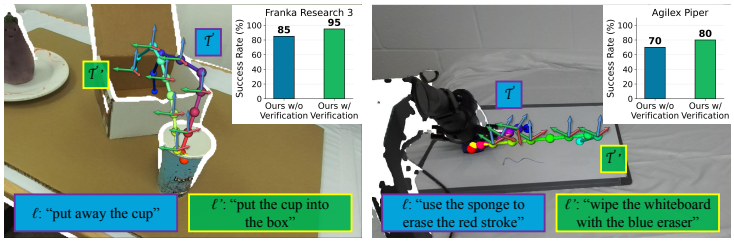


Figure 7: Cycle residual as an inference-time verification signal in real-robot experiments. **(Left)** A semantically and geometrically consistent prediction, where the original $\mathcal{T}$ and the reconstructed $\mathcal{T}'$ closely align, and the recovered instruction ℓ' refines the task interpretation. **(Right)** A case where the predicted motion is geometrically invalid (penetrating the surface), leading to a divergence between $\mathcal{T}$ and $\mathcal{T}'$ and a high cycle residual, which triggers re-sampling. Bar charts in the top right show improvements in success rates with verification on both tasks across Franka Research 3 and Agilex Piper.

Correlation with Prediction Error. We compare our cycle residual against a VLM-based verifier that prompts Qwen3.5-27B (Qwen Team, 2026; Wu et al., 2025; Chen et al., 2025c) with the predicted object motion overlaid on the image and asks for a quality rating from one to ten. Both verifiers operate on the same predictions on the test set without ground truth. To assess how well these signals reflect actual prediction quality, we analyze their *Spearman Rank Correlation* with two motion error metrics: *Average Displacement Error (ADE)*, defined as the mean per-waypoint Euclidean distance for 3D translation (normalized by each episode's ground-truth arc length L_i), and *Average Orientation Error (AOE)*, measuring the mean angular error (in degrees) over $\mathrm{SO}(3)$ relative to ground truth. Table 1 reports both errors stratified by quartiles of each verifier's score, along with Spearman ρ. Our cycle residual yields clearer separation and higher rank correlation. Notably, Q4 attains $2.54\times$ the normalized ADE and $4.64\times$ the AOE of Q1, achieving strong Spearman correlations ($\rho = +0.62$ for

Table 1: Comparison of inference-time verifiers on the test set. Predictions are grouped into quartiles by each verifier's score (Q1: best, Q4: worst) and evaluated by mean 6D motion error per group. Spearman ρ measures correlation of each verifier's score with prediction error (higher is better).

	Rating by Qwen3.5-27B					**Cycle Residual** (Ours)				
Metric	Q1	Q2	Q3	Q4	Spearman ρ (↑)	Q1	Q2	Q3	Q4	Spearman ρ (↑)
Mean ADE (%, ↓)	19.18	23.87	25.05	27.10	+0.21	13.15	21.30	27.33	33.42	+0.62
Mean AOE (°, ↓)	9.01	12.41	19.98	22.92	+0.13	6.16	11.42	18.17	28.57	+0.49

ADE and $\rho = +0.49$ for AOE). These results establish the cycle residual, emerging from the model's internal consistency, as a statistically meaningful quality indicator for predicted motion.

4.4 Ablation Study

We evaluate the effects of the core components of LinguaMotion on grounding, motion prediction, and semantic consistency: **(i) Hierarchical Task Formulation (Hier).** The baseline model is a flat motion predictor trained with standard cross-entropy. Table 2 shows that introducing Hier yields consistent improvements in grounding ability. Figure 8 also shows more accurate state localization and motion prediction. These suggest that decomposing the task into object grounding, state prediction, and motion generation provides an effective structural prior for task understanding. **(ii) Numeric Value Loss ($\mathcal{L}_{\text{nv}}$).** Adding $\mathcal{L}_{\text{nv}}$ further reduces object motion error, while largely maintaining grounding performance, which is consistent with its design: the loss operates only on numerical tokens to improve metric fidelity of spatial predictions. Qualitatively, we observe smoother motion in Figure 8, indicating improved geometric accuracy. **(iii) Bidirectional Training (Bidir).** Enabling bidirectional training yields significant improvements in grounding, motion accuracy, and cycle consistency. The inverse objective acts as a semantic regularizer, encouraging the predicted object motion to retain task information necessary to reconstruct the instruction. This is reflected quantitatively in Spearman ρ more than doubling, from $+0.24$ to $+0.56$, and qualitatively in object motion sequences that more closely align with the goal. Taken together, each component addresses a complementary aspect of the problem, spanning structural grounding, geometric fidelity, and semantic consistency, and their combination reduces ADE from 25.48% to 23.80% and AOE from $22.27°$ to $16.08°$ relative to the flat baseline, yielding the most accurate and coherent predictions.

Table 2: Ablation of core components on the held-out test set. We report metrics on object grounding, motion prediction, and semantic cycle consistency. Best per column is highlighted in **bold**.

Hier	$\mathcal{L}_{\text{nv}}$	Bidir	Bbox IoU ↑	Class (%, ↑)	Role (%, ↑)	ADE (%, ↓)	AOE (°, ↓)	Spearman ρ ↑
×	×	×	0.822	0.930	0.956	25.48	22.27	+0.19
✓	×	×	0.827	0.936	0.963	24.12	19.95	+0.28
✓	✓	×	0.825	0.936	0.962	23.88	17.14	+0.24
✓	✓	✓	**0.829**	**0.940**	**0.965**	**23.80**	**16.08**	**+0.56**



Figure 8: Qualitative comparison corresponding to each configuration in Table 2.

5 Conclusion

We presented **LinguaMotion**, a unified framework that models object-centric motion within the native representation space of Vision-Language Models. By casting manipulation as structured generation, our approach jointly models semantic understanding and spatially grounded motion, enabling more consistent generalization across tasks and environments. Furthermore, we introduced a numeric value loss for geometric fidelity and a cycle-consistent learning framework for semantic alignment and self-verification. Experiments on real-world manipulation demonstrate strong performance of our method across diverse tasks and robot embodiments without specific fine-tuning.

AI USE STATEMENT

In this work, we used generative AI tools for editing a research paper to improve readability, editing code for method implementation, and searching for information. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, assisting with translation, cleaning and reformating datasets, supporting qualitative and thematic data analysis, or interpreting results.

We have reviewed all AI-assisted work. Specifically, LLM-assisted paper writing edits and code implementations were thoroughly verified and tested for correctness, rigor, and factual precision by the authors. Note that open-source vision-language models evaluated or fine-tuned as part of our core research methodology (*e.g.*, Qwen3.5-4B, Qwen3.5-27B) are reported as subjects of study rather than writing or research assistant tools. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide detailed descriptions of our methodology, dataset curation, and implementation details across the main text and appendix. The mathematical formulation of our hierarchical object motion representation, numeric value loss ($\mathcal{L}_{\text{nv}}$), and cycle-consistent bidirectional learning objective are detailed in §3. Comprehensive details on our training data curation pipeline are provided in Appendix §A. Implementation details and bidirectional training prompts are provided in §4.1 and Appendix §B, respectively.

REFERENCES

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024a.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, 2024b.

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

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Boyuan Chen, Tianyuan Zhang, Haoran Geng, Kiwhan Song, Caiyi Zhang, Peihao Li, William T Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, et al. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025a.

Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning. *arXiv preprint arXiv:2506.01953*, 2025b.

Haonan Chen, Jingxiang Guo, Bangjun Wang, Tianrui Zhang, Xuchuan Huang, Boren Zheng, Yiwen Hou, Chenrui Tie, Jiajun Deng, and Lin Shao. Goal-vla: Image-generative vlms as object-centric world models empowering zero-shot robot manipulation. *arXiv preprint arXiv:2506.23919*, 2025c.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, 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.

Karthik Dharmarajan, Wenlong Huang, Jiajun Wu, Li Fei-Fei, and Ruohan Zhang. Dream2flow: Bridging video generation and open-world manipulation with 3d object flow. *arXiv preprint arXiv:2512.24766*, 2025.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.

Haoshu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2022.

Shresth Grover, Akshay Gopalkrishnan, Bo Ai, Henrik I Christensen, Hao Su, and Xuanlin Li. Enhancing generalization in vision-language-action models by preserving pretrained representations. *arXiv preprint arXiv:2509.11417*, 2025.

Siddhant Haldar and Lerrel Pinto. Point policy: Unifying observations and actions with key points for robot manipulation. *arXiv preprint arXiv:2502.20391*, 2025.

Asher J Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting. *arXiv preprint arXiv:2509.22195*, 2025.

Cheng-Chun Hsu, Bowen Wen, Jie Xu, Yashraj Narang, Xiaolong Wang, Yuke Zhu, Joydeep Biswas, and Stan Birchfield. Spot: Se (3) pose trajectory diffusion for object-centric manipulation. In *ICRA*, 2025.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

Chi-Pin Huang, Yunze Man, Zhiding Yu, Min-Hung Chen, Jan Kautz, Yu-Chiang Frank Wang, and Fu-En Yang. Fast-thinkact: Efficient vision-language-action reasoning via verbalizable latent planning. *arXiv preprint arXiv:2601.09708*, 2026.

Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models. In *IROS*, 2024a.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024b.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *ICML*, 2024.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *RSS*, 2024.

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.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to Act from Actionless Videos through Dense Correspondences. *arXiv:2310.08576*, 2023.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025a.

Seungjae Lee, Yoonkyo Jung, Inkook Chun, Yao-Chih Lee, Zikui Cai, Hongjia Huang, Aayush Talreja, Tan Dat Dao, Yongyuan Liang, Jia-Bin Huang, et al. Tracegen: World modeling in 3d trace space enables learning from cross-embodiment videos. *arXiv preprint arXiv:2511.21690*, 2025b.

Mara Levy, Siddhant Haldar, Lerrel Pinto, and Abhinav Shirivastava. P3-po: Prescriptive point priors for visuo-spatial generalization of robot policies. In *ICRA*, 2025.

Hongyu Li, Lingfeng Sun, Yafei Hu, Duy Ta, Jennifer Barry, George Konidaris, and Jiahui Fu. Novaflow: Zero-shot manipulation via actionable flow from generated videos. *arXiv preprint arXiv:2510.08568*, 2025a.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025b.

Xinqing Li, Xin He, Le Zhang, Min Wu, Xiaoli Li, and Yun Liu. A comprehensive survey on world models for embodied ai. *arXiv preprint arXiv:2510.16732*, 2025c.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Raymond Yu, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, et al. Hamster: Hierarchical action models for open-world robot manipulation. *arXiv preprint arXiv:2502.05485*, 2025d.

Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Transactions on Mechatronics*, 2025.

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

Michal Lukasik, Zhao Meng, Harikrishna Narasimhan, Yin-Wen Chang, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. Better autoregressive regression with llms via regression-aware fine-tuning. In *ICLR*, 2025.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision–language–action models for embodied ai. *IEEE TNNLS*, 2026.

James Ni, Zekai Wang, Wei Lin, Amir Bar, Yann LeCun, Trevor Darrell, Jitendra Malik, and Roei Herzig. From generated human videos to physically plausible robot trajectories. *arXiv preprint arXiv:2512.05094*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Shivansh Patel, Shraddhaa Mohan, Hanlin Mai, Unnat Jain, Svetlana Lazebnik, and Yunzhu Li. Robotic manipulation by imitating generated videos without physical demonstrations. *arXiv preprint arXiv:2507.00990*, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya K. Ryali, Tengyu Ma, Haitham Khedr, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

D Shah and TM Aamodt. Learning label encodings for deep regression. arxiv preprint. *arXiv preprint arXiv:2303.02273*, 2023.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. *arXiv preprint arXiv:2512.06963*, 2025.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

Lawrence Stewart, Francis Bach, Quentin Berthet, and Jean-Philippe Vert. Regression as classification: Influence of task formulation on neural network features. In *International Conference on Artificial Intelligence and Statistics*, 2023.

Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Reed Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan D. Ratliff, and Dieter Fox. Curobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023.

Yihe Tang, Wenlong Huang, Yingke Wang, Chengshu Li, Roy Yuan, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Uad: Unsupervised affordance distillation for generalization in robotic manipulation. In *ICRA*, 2025.

Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, et al. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*, 2025.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Homer Rich Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Maximilian Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Ho Vuong, Andre Wang He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *CoRL*, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Shengjie Wang, Jiacheng You, Yihang Hu, Jiongye Li, and Yang Gao. Skil: Semantic keypoint imitation learning for generalizable data-efficient manipulation. *arXiv preprint arXiv:2501.14400*, 2025.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In *AAAI*, 2026.

Junyi Wu, Jiachen Tao, Haoxuan Wang, Gaowen Liu, Ramana Kompella, and Yan Yan. Orientation-anchored hyper-gaussian for 4d reconstruction from casual videos. *Advances in Neural Information Processing Systems*, 2026a.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026b.

Yilin Wu, Ran Tian, Gokul Swamy, and Andrea Bajcsy. From foresight to forethought: Vlm-in-the-loop policy steering via latent alignment. *arXiv preprint arXiv:2502.01828*, 2025.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024a.

Zhiyuan Xu, Kun Wu, Junjie Wen, Jinming Li, Ning Liu, Zhengping Che, and Jian Tang. A survey on robotics with foundation models: toward embodied ai. *arXiv preprint arXiv:2402.02385*, 2024b.

Angen Ye, Zeyu Zhang, Boyuan Wang, Xiaofeng Wang, Dapeng Zhang, and Zheng Zhu. Vla-r1: Enhancing reasoning in vision-language-action models. *arXiv preprint arXiv:2510.01623*, 2025.

Bin Yu, Shijie Lian, Xiaopeng Lin, Zhaolong Shen, Yuliang Wei, Haishan Liu, Changti Wu, Hang Yuan, Bailing Wang, Cong Huang, et al. 3d-mix for vla: A plug-and-play module for integrating vggt-based 3d information into vision-language-action models. *arXiv preprint arXiv:2603.24393*, 2026.

Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024a.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024b.

Jonas Zausinger, Lars Pennig, Anamarija Kozina, Sean Sdahl, Julian Sikora, Adrian Dendorfer, Timofey Kuznetsov, Mohamad Hagog, Nina Wiedemann, Kacper Chlodny, et al. Regress, don't guess–a regression-like loss on number tokens for language models. *arXiv preprint arXiv:2411.02083*, 2024.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *CoRL*, 2024.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*, 2025.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. Vlm4vla: Revisiting vision-language-models in vision-language-action models. *arXiv preprint arXiv:2601.03309*, 2026.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE TPAMI*, 2024.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *CVPR*, 2025.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daume III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

Hongyan Zhi, Peihao Chen, Siyuan Zhou, Yubo Dong, Quanxi Wu, Lei Han, and Mingkui Tan. 3dflowaction: Learning cross-embodiment manipulation from 3d flow world model. *arXiv preprint arXiv:2506.06199*, 2025.

Enshen Zhou, Cheng Chi, Yibo Li, Jingkun An, Jiayuan Zhang, Shanyu Rong, Yi Han, Yuheng Ji, Mengzhen Liu, Pengwei Wang, et al. Robotracer: Mastering spatial trace with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2512.13660*, 2025a.

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *Advances in Neural Information Processing Systems*, 38, 2026.

Pengfei Zhou, Liliang Chen, Shengcong Chen, Di Chen, Wenzhi Zhao, Rongjun Jin, Guanghui Ren, and Jianlan Luo. Act2goal: From world model to general goal-conditioned policy. *arXiv preprint arXiv:2512.23541*, 2025b.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.

A Instruction-Motion Data Curation

Our formulation in §3.1 requires supervision in the form of $(\mathbf{I}, l, \mathcal{H})$ triples, where $\mathcal{H}$ specifies instruction-grounded objects, their spatial state anchors, and their 6D motion trajectories. However, existing real-world robot manipulation datasets, such as BridgeV2 (Walke et al., 2023) and DROID (Khazatsky et al., 2024), provide only raw visual observations paired with low-level robotic control signals (*e.g.*, joint actions or end-effector poses), lacking explicit object grounding or structured object motion annotations. To align these datasets with our formulation, we reinterpret each manipulation episode as a grounded description of object-centric motion and convert it into the target form $\mathcal{H}$. We first follow TraceGen (Lee et al., 2025b) to estimate per-frame depth and camera pose, and prompt Qwen 3.5-27B (Qwen Team, 2026) to generate task-level instructions. Building on this, we extract structured supervision through a four-stage pipeline: **(i)** Instruction-Grounded Object Detection, which identifies task-relevant entities and reasons their semantic roles, **(ii)** Object-Centric Tracking with Drift Correction, which produces temporally consistent 3D keypoint tracks, **(iii)** Object Orientation Sequence Estimation, which solves frame-level $\mathrm{SO}(3)$ orientation via constrained orthogonal Procrustes alignment, and **(iv)** Spatially-Uniform Waypoint Sampling, which converts object motion into a sequence of waypoints with normalized scale and consistent spacing, facilitating stable model training and robot execution.

A.1 Instruction-Grounded Object Detection

Raw teleoperation data do not explicitly specify which objects in the scene are relevant to the task or how they relate to the instruction. To obtain such information, we prompt Qwen 3.5-27B (Qwen Team, 2026) with the visual observation and task description (Lee et al., 2025b). The model aligns object perception with task semantics by producing bounding boxes, semantic labels (*e.g.*, 'cup'), and task role labels for all relevant objects. The task roles include the manipulated `source` object, the `target` inferred by task instruction, and `object` for other entities in the scene. Note that a `target` can be a distinct object, a sub-part of the `source` object, or a specific spatial location. These detections define the object set $\mathcal{O}$ in $\mathcal{H}$ for subsequent object-centric motion tracking.

A.2 Object-Centric 3D Tracking with Drift Correction

Built upon instruction-grounded object detections, the next step is to recover temporally consistent 3D motion for each object. This requires tracking object-centric keypoints over long manipulation sequences, where occlusion, deformation, and appearance variation frequently lead to drift. While state-of-the-art point trackers provide accurate short-term correspondences, they are prone to drift over time, eventually breaking object identity consistency.

We address this by coupling geometric tracking with feature-based re-anchoring. For each object, we initialize a set of keypoints by clustering DINOv2 (Oquab et al., 2023) features within its segmentation mask. These keypoints are tracked over time using CoTracker3 (Karaev et al., 2024), while SAM2 (Ravi et al., 2024) propagates object masks across frames. To correct drift, we periodically verify whether each keypoint remains within its assigned object mask. When a deviation is detected, the keypoint is re-anchored by matching its reference feature from the first frame to the current frame. Formally, for a keypoint i, we relocate it to the position within the current mask $M_t^{(o)}$ that maximizes feature similarity:

$$(u_t^{(i)}, v_t^{(i)})' = \underset{(u,v)\in M_t^{(o)}}{\arg\max} \frac{\phi_t(u,v) \cdot \phi_0^{\mathrm{ref},(i)}}{\|\phi_t(u,v)\| \, \|\phi_0^{\mathrm{ref},(i)}\|}, \tag{3}$$

where $\phi_t(u,v)$ denotes the DINOv2 feature at location (u,v) in frame t, and $\phi_0^{\mathrm{ref},(i)}$ is the reference feature of keypoint i extracted from the first frame. This design combines short-term accuracy of geometric tracking with long-term stability of visual features, yielding trajectories that remain semantically aligned with object identity throughout the manipulation sequence.

A.3 Object Orientation Estimation

We formulate rotation transformation as an optimized point-set registration problem in $\mathrm{SO}(3)$ geometry (Bharadhwaj et al., 2024b; Wu et al., 2026a). At the segment start frame, we collect the support

point set $\mathcal{P}_0 \subset \mathbb{R}^3$ within the aggregated SAM2 mask of the target object. For each frame t, given the corresponding propagated 3D points $\mathcal{P}_t$, we compute the relative rotation matrix $R_t \in \mathrm{SO}(3)$ by solving the closed-form Kabsch alignment:

$$R_t = \underset{R \in \mathrm{SO}(3)}{\arg\min} \sum_{p \in \mathcal{P}_0, q \in \mathcal{P}_t} \left\| R\,(p - \bar{p}_0) - (q - \bar{q}_t) \right\|_2^2, \tag{4}$$

where $\bar{p}_0$ and $\bar{q}_t$ denote the spatial centroids of $\mathcal{P}_0$ and $\mathcal{P}_t$, respectively. Constructing the cross-covariance matrix $H = (P_0 - \bar{p}_0)^\top (P_t - \bar{q}_t)$ with SVD decomposition $H = U \Sigma V^\top$, the globally optimal rotation matrix is obtained as:

$$R_t = V\,\mathrm{diag}\left(1,\, 1,\, \det(VU^\top)\right) U^\top. \tag{5}$$

The determinant sign correction term $\det(VU^\top)$ guarantees $R_t \in \mathrm{SO}(3)$, preventing improper reflection degeneracies under near-coplanar point distributions.

A.4 Spatially-Uniform Waypoint Sampling

The final stage converts dense, noisy 6D pose trajectories into a compact, structurally faithful sequence of $K = 8$ pose waypoints, denoted as $\mathbf{W} = \big((p_1, R_1), \ldots, (p_K, R_K)\big)$, where $p_k \in \mathbb{R}^3$ and $R_k \in \mathrm{SO}(3)$. A naive time-uniform sampling scheme allocates equal capacity across fixed time intervals regardless of physical motion magnitude, leading to redundant oversampling during static phases and severe undersampling during rapid execution. To better preserve the intrinsic geometric structure of object motion, we reparameterize the trajectory along its spatial extent using planar arc length. Let L denote the total arc length of the trajectory centroid. We distribute the K waypoints uniformly such that the k-th waypoint is positioned at the target cumulative arc length:

$$\ell_k = \frac{k-1}{K-1}\, L, \quad k \in \{1, \ldots, K\}. \tag{6}$$

This target location falls within a temporal frame segment spanned by boundary frames $j^\star$ and $j^\star + 1$, characterized by a fractional offset parameter $\alpha_k \in [0, 1]$.

To guarantee strict spatio-temporal alignment between translational motion and rotational orientation, p_k and R_k are jointly evaluated using identical interpolation weight α_k. Specifically, translation is updated via Euclidean linear interpolation, while orientation is smoothly transitioned along the $\mathrm{SO}(3)$ geodesic manifold using Lie algebraic logarithmic and exponential mappings:

$$p_k = (1 - \alpha_k)\, p_{j^\star} + \alpha_k\, p_{j^\star + 1}, \tag{7}$$

$$R_k = R_{j^\star} \exp\left(\alpha_k \log\left(R_{j^\star}^\top R_{j^\star + 1}\right)\right). \tag{8}$$

This formulation avoids non-physical rotational interpolation artifacts and ensures spatial smoothness. The resulting sequence $\mathbf{W}$, anchored by the initial and goal spatial states, constitutes the normalized 6D motion trajectory component of $\mathcal{H}$ defined in §3.1.

B Prompts for Bidirectional Training

We implement bidirectional learning (§3.4) through a unified prompting scheme, where both instruction-to-motion and motion-to-instruction tasks are formulated as conditional generation within a single model. Each direction is specified by a prefix token (`[INSTR_TO_MOTION]` or `[MOTION_TO_INSTR]`) that indicates the generation mode.

For instruction-to-motion generation, the model takes an RGB image and a language instruction as input, and produces a structured object-centric representation, including object attributes and motion trajectories, as shown in Figure 9. For motion-to-instruction generation, the model is conditioned on the visual observation together with the object-centric motion, and reconstructs the corresponding instruction, as illustrated in Figure 10. This shared prompting design allows both directions to be modeled within the same generative framework, enabling parameter sharing and encouraging consistent representation learning across tasks.

Input prompt:
[INSTR_TO_MOTION]
<rgb image>, The robot task is: **"Place the wooden ring onto the tall cylinder."** Output a JSON list of task-relevant objects, each with **"label"** (object name), **"bbox_2d"** [x1,y1,x2,y2] in 0-1000 normalized coordinates, and "role" ("source", "target", or "object"). Then on a new line output the source object's pose trajectory: **"current:"** [u,v,z,rx,ry,rz], **"goal:"** [u,v,z,rx,ry,rz], and **"waypoints:"** with 8 intermediate [u,v,z,rx,ry,rz] entries, all in 0-1000 normalized scale.

GT output:
[{**"label"**:"wooden_ring",**"bbox_2d"**:[384,0,467,48],**"role"**:"source"},
{**"label"**:"tall_cylinder",**"bbox_2d"**:[415,33,475,270],**"role"**:"target"}]
current: [419,48,337,500,500,500]
goal: [546,415,336,512,498,634]
waypoints:[441,33,372,500,500,500],[429,61,376,500,500,500],[446,111,370,508,488,471],...,[529,364,342,517,445,410]

Figure 9: Example prompt for instruction-to-motion generation (`[INSTR_TO_MOTION]`). Given an RGB image and a natural language instruction, the model predicts object-centric representations, including bounding boxes, semantic categories and roles, and task-guided motion.

Input prompt:
[MOTION_TO_INSTR]
<rgb image>, A robot manipulation task was performed. The source object's pose trajectory is given as "current:" [u,v,z,rx,ry,rz], "goal:" [u,v,z,rx,ry,rz], and "waypoints:" with 8 intermediate [u,v,z,rx,ry,rz] entries, all in 0-1000 normalized scale:
current: [342,79,762,500,500,500] **goal**: [355,95,740,512,498,634]
waypoints: [345,83,758,500,500,500],...,[355,95,740,517,445,610]
What task was performed? Describe as a concise command.

GT output:
pick up red_pepper and place it in metal_pot

Figure 10: Example prompt for motion-to-instruction generation (`[MOTION_TO_INSTR]`). Conditioned on the visual observation and object motion representation, the model reconstructs the corresponding task instruction, enabling bidirectional learning and semantic consistency.

C Analysis of Distribution Shift in Real-World Evaluation

To better contextualize the zero-shot results in §4.2, we analyze the gap between our real-world evaluation scenarios and the training distribution along two complementary perspectives: categorical decomposition and embedding-space similarity. The results, summarized in Table 3 and Figure 11, show that our experiments involve substantial distribution shifts across embodiment, objects, visual appearance, and language instruction, suggesting the challenging nature of the evaluation protocol.

Table 3 provides a detailed analysis of distribution shift across four dimensions. **(i) Embodiment shift** is mild out-of-distribution (OOD) for Franka Research 3, which succeeds the Panda platform used in DROID, and strong OOD for Agilex Piper, which is entirely unseen during training. **(ii) Object shift** exhibits two distinct regimes. For T1-T4, the manipulated object categories are generally present in the training data, but the specific object instances differ substantially in appearance, material, or form factor (*e.g.*, toy, plastic, or plush variants of semantically similar objects), resulting in mild object-level OOD generalization. In contrast, T5-T7 involve tools and object categories such as whiteboards, brooms, dustpans, and knives that are rare or absent from the training set, representing substantially stronger object shift. **(iii)** To quantify **visual shift**, we compute the mean SigLIP-2 image-level nearest-neighbor cosine distance between evaluation images and the training set. All seven tasks fall into the mild-OOD regime (0.178-0.244), substantially above the train-train reference threshold ($p_{95} = 0.084$). The full distribution analysis is provided in Figure 11. **(iv)** Finally, we analyze **linguistic shift** through the frequency of task verbs and relational phrases in the training corpus. Several evaluation instructions are extremely rare or entirely unseen during training: "between" and "next to" appear in only $\sim 0.02\%$ of instances, "put away" does not appear at all, and the verbs

Table 3: Analysis of train-test distribution shift (● in-distribution, ◐ mild-OOD, ○ strong-OOD).

Task	Embodiment		Object Category / Instance	Visual Similarity	Task Verb / Phrase
	Franka Research 3	Agilex Piper			
T1 Basic	◐	○	◐	◐ 0.227	● 17.3%
T2 Attribute / Relation	◐	○	◐	◐ 0.220	● 15.6%
T3 Spatial Reasoning	◐	○	◐	◐ 0.244	○ 0.02%
T4 Semantic	◐	○	◐	◐ 0.193	○ 0%
T5 Tool Use: Wipe	◐	○	○	◐ 0.178	○ 0.5%
T6 Tool Use: Sweep	◐	○	○	◐ 0.206	○ 0.3%
T7 Tool Use: Cut	◐	○	○	◐ 0.224	○ 0%

"wipe", "sweep", and "cut" together account for less than 1% of the corpus, with existing training examples limited to narrow contexts (*e.g.*, "wipe stovetop" or "sweep beans").

We further complement the categorical analysis with an embedding-space measurement of visual distribution shift using a pretrained SigLIP-2 encoder (Tschannen et al., 2025). For each evaluation image, we compute the cosine distance to its nearest neighbor in our training set. As a reference baseline, we perform the same nearest-neighbor computation for 5,000 randomly sampled training images against the remaining training corpus. As shown in Figure 11, the train-set self-similarity distribution is tightly concentrated around low distances (median 0.033), whereas the real-world evaluation scenarios form a clearly separated distribution with a substantially larger median distance of 0.218. Notably, all evaluation samples lie beyond the train-train $p_{99} = 0.118$ threshold, indicating a significant visual shift from the training distribution.

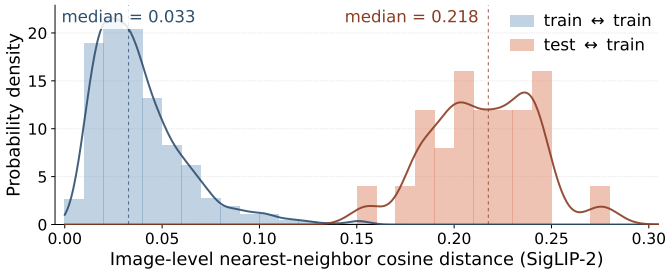


Figure 11: Embedding-space analysis of visual distribution shift using a pretrained SigLIP-2 encoder (Tschannen et al., 2025). The blue histogram shows nearest-neighbor cosine distance distribution among training images (train ↔ train), while the orange measures distances between real-world evaluation images and the training set (test ↔ train). The two distributions are clearly separated, with all test scenarios lying far beyond the typical train-set similarity range, indicating substantial visual shift from the training distribution.

D Analysis of Failure Mode on Real-World Manipulation

We conduct a detailed failure mode analysis over the real-world experiments performed in §4.2, covering all seven robot manipulation tasks. For each task, we collectively perform 40 trials on Franka Research 3 and Agilex Piper and categorize the outcomes based on object grounding, trajectory generation, and task completion, as summarized in Figure 12.

Our findings are: **(i)** A consistent observation across all tasks is that object grounding is highly reliable. In all 280 trials, the model successfully identifies task-relevant objects and their roles, indicating that representing manipulation within the vision-language space effectively preserves semantic understanding ability. **(ii)** Failures primarily arise in the geometric reasoning stages, rather than perception or task interpretation. For basic manipulation and spatial reasoning tasks (T1-T3), errors are mainly due to small spatial deviations, such as goal offsets or inaccurate object interaction near object boundaries. These errors suggest that while the predicted motion is semantically correct, fine-grained geometric precision remains challenging in certain cases. **(iii)** For more complex tasks involving semantic reasoning (T4), occasional failures occur when interacting with constrained geometries, such as collisions with container boundaries, despite correct object identification and goal prediction. This highlights the difficulty of translating high-level intent into physically feasible motion trajectories under complex spatial constraints. **(iv)** Tool-use tasks (T5-T7) reveal additional challenges in modeling complex interactions among objects. Although the model consistently grounds



Figure 12: Failure analysis across seven real-world manipulation tasks performed in §4.2. Each task includes 40 trials on Franka Research 3 and Agilex Piper robots, with outcomes categorized into intermediate stages, including object grounding, motion generation, and task completion. Green flows indicate successful progression, while other branches denote different failure modes.

both tools and targets, failures arise from incomplete or suboptimal motion, such as partial wiping, insufficient physical contact, or incorrect patterns (*e.g.*, sliding instead of cutting). These behaviors indicate challenges in modeling fine-grained contact dynamics.

Overall, the analysis suggests that our LinguaMotion reliably understands what to do and which objects are involved, while most failures stem from how precisely the motion is realized. This points to future directions in improving geometric fidelity, contact modeling, and closed-loop refinement, while reaffirming the strength of the proposed representation in semantic grounding and task understanding.

E General Vision-Language Knowledge Preservation

Adapting Vision-Language Models (VLMs) to robotic data often comes at the cost of degrading their general multi-modal understanding capabilities. This issue is particularly pronounced when the adaptation introduces task-specific output spaces or training objectives that diverge from the model's pretraining regime. To evaluate the preservation of vision-language knowledge, we compare the pretrained Qwen3.5-4B backbone with our fine-tuned LinguaMotion on a suite of vision-language understanding benchmarks, including MME, OCRBench, TextVQA, DocVQA, InfoVQA, ChartQA, and AI2D (Qwen Team, 2026). We further include representative Vision-Language-Action (VLA) models and co-trained models for comparison.

Results are reported in Table 4. As observed, prior VLA approaches often suffer severe degradation after fine-tuning the VLM backbone (Karamcheti et al., 2024), in some cases collapsing to near-zero performance across multiple benchmarks. In contrast, our method maintains substantially higher performance across all tasks, without exhibiting catastrophic forgetting. Compared to the original VLM backbone (Qwen Team, 2026), LinguaMotion shows a consistent but moderate decrease across benchmarks. We attribute this gap to the domain shift introduced by robot-centric fine-tuning, which inevitably biases the model toward manipulation-specific patterns. Importantly, however, the model

Table 4: Comparison of our method with VLM backbones and representative VLA models on standard multimodal benchmarks. While VLAs often suffer severe degradation after adapting backbones to robot data, our method avoids catastrophic forgetting and retains substantially higher performance across all benchmarks, with moderate drops relative to the backbone.

Method	#P	MME	OCRBench	TextVQA	DocVQA	InfoVQA	ChartQA	AI2D
Co-Trained VLAs								
$\pi_{0.5}$ (Intelligence et al., 2025)	3B	1061.9	6.8	10.0	4.6	7.7	5.1	27.0
MolmoAct (Lee et al., 2025a)	7B	1224.5	52.7	57.5	58.7	41.9	55.9	2.0
Specialized VLAs								
Prismatic VLM (backbone) (Karamcheti et al., 2024)	7B	1456.6	32.0	42.5	17.5	19.7	16.7	54.6
OpenVLA (Kim et al., 2024)	7B	0	0	0	0	0	0	0
ECoT (Zawalski et al., 2024)	7B	0	0.01	0	0	0	0	0
Qwen3.5-4B (backbone) (Qwen Team, 2026)	4B	1569	86.4	82.9	95.6	80.7	85.2	83.9
LinguaMotion	4B	1382	78.9	75.4	93.2	75.4	82.0	76.2

retains strong sensitivity to object identity, spatial relations, and visual semantics, which are critical for both instruction understanding and real-world manipulation. These results support our design principle: by keeping motion prediction within the VLM's native representation space, fine-tuning leads to a favorable balance between task adaptation and general knowledge preservation.

F Limitations and Future Work

Our approach builds upon pretrained Vision-Language Models (VLMs) as the backbone. While this enables us to leverage rich world knowledge from pretrained VLMs, it also means that the performance of our system is inherently affected by the capabilities and limitations (*e.g.*, hallucination) of the underlying backbone. Nonetheless, as VLMs continue to improve in capability and reliability, our framework can directly benefit from these advances without changing the overall formulation. In addition, achieving real-time closed-loop control alongside generative object-centric motion modeling remains a open challenge in this field. Our current system generates motion in a predominantly open-loop fashion, which effectively balances model expressiveness with inference efficiency across standard tasks. However, for scenarios demanding high-frequency reactivity or dynamic error recovery, bridging this open-loop formulation with multi-stage planning or agentic re-planning strategies represents an essential direction. Iteratively refining motions based on updated execution outcomes will be key to extending such generative representations to highly dynamic or long-horizon horizons. Finally, while our compact representation of semantic keypoints and their transformations provides a computationally efficient abstraction for a broad range of manipulation tasks, modeling complex physical phenomena, such as fine-grained geometric interactions and deformations among objects, still presents a fundamental challenge for the robot manipulation area. Extending this framework toward adaptive or denser representations to capture spatial deformations is an exciting and important venue for future research. Despite these remaining open challenges, we hope this work highlights the importance of motion representation in robotic manipulation and provides a step toward more unified approaches to general-purpose robot learning.

G Broader Impacts

This work aims to advance generalizable robot manipulation by bridging vision-language understanding with structured motion generation. Such capabilities could broaden the applicability of robots in real-world environments, including assistive robotics, household automation, and industrial settings, where adaptability to diverse tasks and environments is essential. By leveraging shared representations across perception, language, and action, our approach may contribute to more flexible and intuitive human-robot interaction.

At the same time, increased autonomy in robotic systems raises important considerations. Improved generalization may enable robots to operate in less structured or more dynamic environments, where failures could have safety implications. Ensuring reliability, interpretability, and appropriate human oversight remains critical for real-world deployment. In addition, as our method builds upon pretrained vision-language models, it may inherit biases or limitations present in the underlying data, potentially affecting performance across different objects, environments, or user instructions.

We believe that addressing these challenges requires continued research in robust perception, safe control, and bias mitigation, as well as careful system-level evaluation before deployment. Overall, our work takes a step toward more general and adaptable robotic systems, while highlighting the importance of aligning technical advances with safety and responsible use.