SemanticVLA: Towards Semantic Reasoning over Action Memorization via Synergistic Explicit Trace and Latent Action Planning

Supplementary Material

A. Model Architecture Details

We provide comprehensive architectural specifications for SemanticVLA's three core components: the semantic latent action tokenizer, VLM co-training architecture, and flow matching action decoder. Our design maintains strict modality separation where trace coordinates and latent tokens provide semantic guidance to the VLM, while raw actions are confined to the action expert.

A.1. Semantic Latent Action Tokenizer

Our two-stage VQ-VAE learns trace-guided latent action tokens without language contamination. Stage 1 extracts pure geometric primitives from trace coordinates, while Stage 2 grounds these into visual observations through dual reconstruction supervision.

Stage 1: Trace-Level Geometric Abstraction. The trace encoder processes normalized 2D coordinate sequences $\tau = (p_1, \ldots, p_L)$ where $p_i \in [0,1]^2$ through 12-layer transformers with sinusoidal position encoding. The encoder compresses traces into continuous representations z^{trace}, which are quantized into a 32-entry codebook $\mathcal{C}^{\text{trace}}$ producing 4 discrete tokens per trace. A 5-layer decoder reconstructs original traces supervised by $\mathcal{L}_{\text{vq}}^{\text{trace}}$. This stage learns geometric manipulation primitives (reaching, grasping, placing) invariant to visual appearance. The trace length L matches the action chunk size H used in downstream tasks. Architecture details in Table 3 (left).

Stage 2: Visual Grounding with Geometric Scaffolding. The fusion encoder combines pretrained trace codebook entries $c_{q^{\text{trace}}}^{\text{trace}}$ with frozen DINOv2 features $h^{\text{visual}} \in \mathbb{R}^{256\times768}$ from observations (o_t, o_{t+H}) via cross-attention, where geometric priors query task-relevant visual regions. Four transformer layers refine these fused representations before quantization into action codebook $\mathcal{C}^a$ (32 entries, 4 tokens), yielding latent actions $e_a = c_{q_a}^a$. Dual reconstruction—spatial decoder for trace precision ($\mathcal{L}_{\text{recon}}^{\text{trace}}$) and visual decoder for semantic understanding ($\mathcal{L}_{\text{recon}}^{\text{visual}}$)—ensures e_a captures both geometric structure and scene-specific semantics. The spatial decoder reuses the trace encoder/decoder architecture from Stage 1. Both decoders are discarded post-pretraining. Architecture in Table 3 (right).

A.2. VLM Co-training Architecture

Building upon pretrained latent tokens, we integrate dual-path reasoning into the VLM backbone, which fuses SigLIP and DINOv2 visual encoders with a large language model. To enable latent action prediction without introducing raw action tokens that degrade reasoning capabilities, we expand the vocabulary with $K = 32$ special tokens $\{\texttt{<ACT\_1>}, \ldots, \texttt{<ACT\_K>}\}$ corresponding to the action codebook $\mathcal{C}^a$. Given observation o_t and instruction ℓ_t, the VLM generates explicit trace reasoning as textual coordinate sequences through its native language interface, directly reusing pretrained spatial grounding to produce τ. Simultaneously, it predicts a sequence of latent action token indices $q_{1:N}$ through the same language modeling head, producing compact semantic representations. We apply Low-Rank Adaptation (LoRA) to vision-language projectors and LLM attention layers while keeping visual encoders frozen, enabling efficient finetuning that preserves foundation model capabilities. During Stage 2 co-training, supervision derives purely from spatial traces and pretrained latent vocabulary—no ground-truth robot actions are required. This dual-path architecture maintains modality separation: trace leverages the VLM's compositional reasoning for interpretable planning, while latent tokens offer visually-grounded execution guidance.

A.3. Flow Matching Action Decoder

The final component translates discrete VLM outputs into continuous robot actions through a transformer-based flow matching decoder. This conditioning mechanism enables trace to provide interpretable spatial waypoints while latent tokens supply fine-grained visuomotor guidance, with visual features grounding both in scene-specific observations. The architecture receives three complementary conditioning signals: (1) latent action hidden states E_a from the VLM's final layer encoding multimodal reasoning over observations, instructions, and predicted traces; (2) frozen trace embeddings $e_\tau = \phi_{\text{enc}}^{\text{trace}}(\tau)$ from Stage 1 encoder providing pure geometric structure invariant to visual appearance; and (3) fused visual tokens E_v offering dense spatial context from current observations. These signals are projected to a common dimension and concatenated as cross-attention keys for the transformer decoder. The decoder employs 4 transformer layers with 8 attention heads and 512 hidden dimensions. Learnable action queries attend to the fused conditioning through multi-layer self-attention and cross-attention mechanisms. The feedforward networks use 2048 dimensions for feature transformation. During training, we minimize flow matching objectives that predict velocity fields for iterative denoising from noise to ground-truth actions, using 10 denoising steps. The action chunk size is $H = 12$ for LIBERO and $H = 10$ for SimplerEnv

Table 3. **Semantic Latent Action Tokenizer Architecture.** Two-stage VQ-VAE architecture for trace-guided latent tokens. Stage 1 learns geometric patterns from traces. Stage 2 grounds them in visual observations, with dual reconstruction of trace and visual representations producing latent actions with both spatial and visual semantics.

Stage 1: Trace-Level VQ-VAE			**Stage 2: Visual Grounding VQ-VAE**		
Trace Encoder $\phi_{\text{enc}}^{\text{trace}}$	Num Layers	12	Visual Features (Frozen)	Backbone	DINOv2-Base (86M)
	Attention Heads	8		Output Dim	256×768
	Hidden Dim	256	Fusion Encoder $\phi_{\text{enc}}^{\text{fused}}$	Transformer Layers	4
	Output Dim	768		Attention Heads	8
	Position Encoding	Sinusoidal		Hidden Dim / Output	512 / 768
VQ Codebook $\mathcal{C}^{\text{trace}}$	Codebook Size	32	VQ Codebook $\mathcal{C}^{a}$	Codebook Size	32
	Embedding Dim	768		Embedding Dim	768
	Codes Num	4		Codes Num	4
Trace Decoder $\phi_{\text{dec}}^{\text{trace}}$	Num Layers	5	Visual Decoder $\phi_{\text{dec}}^{\text{visual}}$	Num Layers	6
	Attention Heads	8		Attention Heads	8
	Hidden Dim	256		Hidden Dim	640
	Output	$L \times 2$ coords		Output	256×768
Total Parameters		$\sim$16.4M	Total Parameters		$\sim$55.7M

and real-world deployment.

B. Training Details

Our training follows a three-stage pipeline designed to maintain clean modality separation between VLM reasoning and action control. Tab 4 summarizes the complete training configuration across all stages.

Stage 1: Semantic Latent Token Pretraining. We pretrain the two-stage VQ-VAE on TraceX-240K for 50K steps with batch size 512, using AdamW optimizer with learning rate 1e-4 and cosine decay schedule. This stage learns trace-level geometric primitives and visually-grounded action tokens without language contamination, establishing a clean vocabulary of manipulation semantics through dual reconstruction supervision ($\mathcal{L}_{\text{LAT}}$).

Stage 2: VLM Co-training with Trace and Latent Action. We co-train Prismatic-7B on TraceX-240K to jointly predict trace coordinates and latent action tokens for 100K steps with batch size 256. The learning rate is 1e-4 with cosine schedule and 5K warmup steps to ensure stable convergence. The VLM backbone integrates SigLIP and DINOv2 visual encoders with LLaMA-2 language model—we keep visual encoders frozen to preserve pretrained visual representations while training the language backbone and vision-language projectors. We employ gradient accumulation with factor 2 to stabilize training on the expanded vocabulary. Critically, this stage requires no ground-truth robot actions—supervision derives purely from spatial traces and pretrained latent token indices through the dual-path objective $\mathcal{L}_{\text{trace}} + \mathcal{L}_{\text{latent}}$. This establishes semantic correspondence between explicit trace reasoning and implicit action primitives latent tokens while preserving the VLM's compositional understanding.

Stage 3: Flow Matching Action Decoding. For downstream task adaptation, we finetune the complete pipeline end-to-end with reduced learning rate 2e-5 for 20K–40K steps depending on dataset size (e.g., 30K for LIBERO, 40K for SimplerEnv, 20K for real-world tasks with limited demonstrations). We apply LoRA to the VLM with rank 32 and keep visual encoders frozen, while the flow matching decoder trains from scratch. The combined objective $\mathcal{L}_{\text{flow}} + \lambda_{\text{VLM}}\mathcal{L}_{\text{VLM}}$ with regularization weight $\lambda_{\text{VLM}} = 0.1$ ensures action decoder adaptation while preserving dual-path reasoning capabilities established in Stage 2. Batch size is reduced to 128 with 1K warmup steps to accommodate the larger memory footprint of flow matching. All training is conducted on 16 NVIDIA H200 GPUs.

C. Dataset

C.1. Dataset Mixture

To enable large-scale pretraining of our semantic latent action tokenizer and VLM co-training, we curate **TraceX-240K**, a comprehensive trace-annotated dataset containing 240K robot manipulation trajectories spanning multiple embodiments and task domains. The dataset composition draws from four major sources: DROID [26], Fractal [5], Bridge V2 [17], and BC-Z [23], selected to maximize coverage of manipulation primitives while maintaining balanced representation across robot platforms. DROID contributes 48K episodes (20%) with dense temporal coverage at 15 FPS, providing rich visual diversity from in-the-wild Franka manipulation. Fractal supplies 87K episodes (36%) captured on Google Robot at 3 FPS, offering 599 distinct task variations that enhance compositional generalization. Bridge V2 provides 53K episodes (22%) from WidowX platform at 5 FPS, covering extensive object and scene diversity crucial for cross-embodiment transfer. BC-Z contributes 52K episodes (22%) on Google Robot at 10

Table 4. **Three-Stage Training Configuration.** Hyperparameters for semantic latent token pretraining, VLM co-training, and end-to-end finetuning.

Hyperparameter	**Stage 1** **Latent Pretraining**	**Stage 2** **VLM Co-training**	**Stage 3** **Action Finetuning**
Batch Size	512	256	128
Learning Rate	1e-4	1e-4	2e-5
Optimizer	AdamW	AdamW	AdamW
Weight Decay	0.01	0.01	0.01
LR Schedule	Cosine	Cosine	Cosine
Warmup Steps	2,000	5,000	1,000
Training Steps	50K	100K	20K–40K
Gradient Accumulation	1	2	1
Trainable Modules	VQVAE (both stages)	VLM (Full) + Projector	VLM (LoRA) + Decoder
Frozen Modules	-	Visual Encoders	Visual Encoders
Loss Components	$\mathcal{L}_{\text{LAT}}$	$\mathcal{L}_{\text{trace}} + \mathcal{L}_{\text{latent}}$	$\mathcal{L}_{\text{flow}} + \lambda_{\text{VLM}}\mathcal{L}_{\text{VLM}}$

FPS, featuring high-quality teleoperation demonstrations for pick-and-place primitives. The resulting mixture contains approximately 240K episodes with robot platform distribution of 20% Franka, 58% Google Robot, and 22% WidowX—ensuring our latent action tokenizer learns generalizable visuomotor primitives grounded in diverse manipulation semantics.

C.2. Trace Annotation Pipeline

To obtain dense spatial annotations for TraceX-240K, we develop an automated trace extraction pipeline that ensures both coverage and precision. For DROID trajectories, we leverage the camera intrinsics and extrinsics provided in the updated dataset release to directly project 3D end-effector positions from robot state into 2D image coordinates, yielding ground-truth trace annotations. For the remaining datasets (Fractal, Bridge V2, BC-Z) lacking such calibration data, we design an automated annotation pipeline combining vision-language model inference with optical flow tracking to ensure both coverage and temporal consistency.

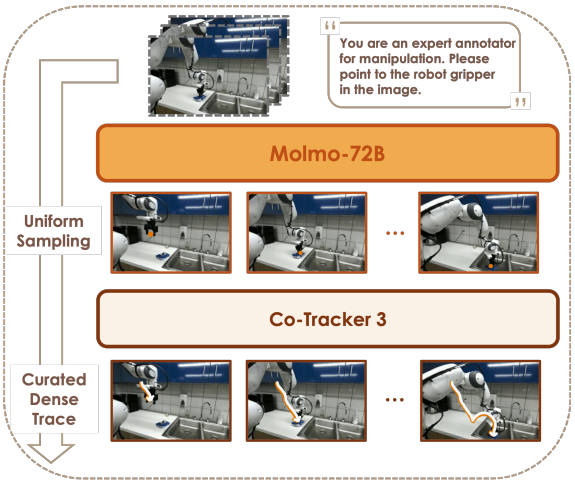

Figure 8. **Trace Annotation Pipeline.** Automated trace extraction combining Molmo-72B for keyframe annotation and CoTracker for dense temporal interpolation, generating spatially-aligned supervision for latent action pretraining.

Keyframe Annotation. Applying Molmo-72B [14] to every frame proves computationally prohibitive and yields inconsistent predictions due to temporal jitter and occlusion-induced localization errors. We instead adopt a sparse keyframe strategy: uniformly sample 8-10 keyframes per trajectory and prompt Molmo to identify end-effector positions as normalized 2D coordinates $p_i = (u_i, v_i) \in [0, 1]^2$. This reduces annotation cost by 10-15$\times$ while providing semantic anchors at critical manipulation phases (grasp initiation, transport, placement).

Temporal Interpolation and Refinement. Between keyframes, we apply CoTracker [24] for dense point tracking, propagating gripper locations across all intermediate frames to generate continuous trace sequences. We fuse CoTracker's dense predictions with Molmo's sparse keyframe anchors through temporal-distance-weighted blending, ensuring the final traces $\tau = (p_1, \ldots, p_L)$ remain semantically aligned with task-critical waypoints while maintaining smooth, physically-plausible motion. This hybrid pipeline yields temporally-coherent supervision covering complete manipulation episodes without manual annotation, with consistency filtering between Molmo and CoTracker rejecting approximately 10% of trajectories exhibiting irrecoverable tracking failures.

Figure 9. **TraceX-240K Dataset Construction.** Our curated dataset contains 240K trace-annotated robot trajectories from several widely-used robotics datasets, providing diverse manipulation primitives across multiple embodiments with automated dense spatial annotations.

D. Experiment Settings

D.1. Simulation Environments

We evaluate SemanticVLA on two complementary simulation benchmarks that probe different aspects of generalization: LIBERO for language-conditioned task diversity and SimplerEnv for cross-domain visual robustness.

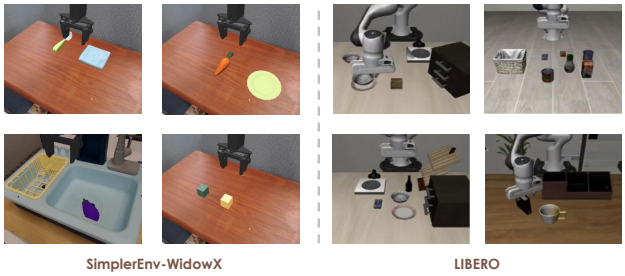

Figure 10. **Two complementary Simulations.** We conduct two complementary simulations including LIBERO and SimplerEnv.

LIBERO. We assess manipulation capabilities across four task suites on the Franka Panda arm [37]: *Spatial* (spatial reasoning with position references), *Object* (object attribute variations), *Goal* (goal specification diversity), and *Long* (compositional long-horizon tasks). Following established protocols [28], we finetune independently per suite for 30K steps with batch size 128 and action chunk size $H = 12$. Each suite contains 10 tasks, and we evaluate over 50 rollouts per task with randomized initial states. Success is determined by task-specific programmatic checks (e.g., object placement within tolerance thresholds, sequential subtask completion).

SimplerEnv. We evaluate cross-domain transfer on the WidowX manipulation suite [33], which tests robustness to visual appearance shifts through controlled lighting and background perturbations. We train on Bridge dataset demonstrations [17] and evaluate across four tasks: *Put Spoon*, *Put Carrot*, *Stack Block*, and *Put Eggplant*. Each task is assessed over 24 episodes with systematic visual variations. The WidowX platform's different morphology and camera viewpoint from training data enables stress-testing of learned visuomotor primitives. Following prior work [9, 50], we finetune for 40K steps with batch size 128 and action chunk size $H = 10$. Success requires precise end-effector positioning and stable grasp maintenance, evaluated through simulator state checks.

D.2. Real-World Environment

We conduct evaluations on a physical robot platform to validate the performance on tasks requiring genuine visual reasoning beyond memorized patterns.

Hardware Setup. Our platform comprises a Franka Research 3 robot arm (7-DOF, maximum reach 855mm) equipped with a Franka Hand parallel-jaw gripper. Visual observations are captured through two Intel RealSense D435 RGB-D cameras: one *exterior* camera mounted on an adjustable stand providing third-person view of the workspace, and one *wrist* camera affixed near the end-effector for close-up manipulation feedback. The workspace is a 120cm $\times$ 80cm tabletop with controlled lighting to minimize shadows while preserving realistic visual complexity. The system runs on a single NVIDIA RTX 3090 GPU.

Task Design. We evaluate two complementary categories, each containing 2 tasks with 5 variants per task, totaling 20 task variants with 20 rollouts each:

- *Long-horizon compositional tasks* require maintaining coherent plans across sequential subtasks without explicit subgoal annotations. *Table sorting* involves categorizing objects (vegetables, fruits, or meat) into designated regions based on semantic understanding. *Food preparing* encompasses multi-step meal assembly such as sandwich construction, requiring precise sequencing of ingredient placement and utensil manipulation.
- *Reasoning-intensive tasks* demand explicit visual reasoning integrated with manipulation. *Math calculation*

Figure 11. **Real-World Robot Platform.** Franka Research 3 manipulation system with dual Intel RealSense D435 cameras. Our experiments involve diverse everyday objects (vegetables, fruits, utensils, blocks) in realistic tabletop scenarios for long-horizon compositional and reasoning-intensive manipulation tasks.

interprets whiteboard arithmetic (e.g., "5+3") to place blocks at computed positions, testing VLM's ability to perform symbolic reasoning and spatial grounding simultaneously. *Word spelling* composes words from letter blocks based on whiteboard specifications, requiring integrated scene understanding, linguistic processing, and fine-grained pick-and-place control.

Each task variant modifies object positions, scene layouts, or semantic targets (e.g., different arithmetic problems, alternative word spellings) to probe generalization beyond training distributions. Success criteria are manually verified: for compositional tasks, all subtasks must complete in correct sequence; for reasoning tasks, final object configurations must match computed targets. We set a maximum episode length of 800 timesteps, marking trials as failures if tasks remain incomplete.

Training Protocol. For real-world deployment, we collect 50 demonstrations per task via kinesthetic teaching. Following Table 4, we finetune the complete SemanticVLA pipeline for 20K steps with batch size 128 and action chunk size $H = 10$. The VLM employs LoRA adaptation (rank=32) with learning rate 2e-5, while the flow decoder trains with learning rate 1e-4. We apply weak VLM regularization weight $\lambda_{\text{VLM}} = 0.1$ to preserve dual-path reasoning established during pretraining. Data augmentation includes random cropping and color jittering to improve robustness to lighting variations. This real-world setup validates three critical properties: (1) *sim-to-real transfer* via TraceX-240K pretraining generalizing to physical robot dynamics, (2) *genuine reasoning* through tasks requiring compositional understanding beyond pattern matching, and (3) *sample efficiency* by achieving robust performance with limited demonstrations per task.

E. Generalization Evaluation Setting

E.1. Instruction Rephrasing in Simulation

To assess whether SemanticVLA genuinely understands language instructions rather than memorizing linguistic-action correlations, we systematically evaluate robustness under instruction variations that preserve task semantics while altering surface forms. Following recent robustness protocols [18, 19, 50], we design rephrasing strategies across LIBERO and SimplerEnv benchmarks, categorizing variations into two levels: lexical substitution and semantic-level rephrasing.

Level 1: Lexical Substitution. This category employs direct synonym replacement while maintaining sentence structure, testing robustness to surface-level linguistic variations:

- *Verb variation*: "pick up" → "grasp", "place" → "put"
- *Noun variation*: "plate" → "dish", "compartment" → "section"
- *Color adjective*: "yellow and white" → "bright yellow", "red" → "dark red"

Level 2: Semantic-Level Rephrasing. This category replaces direct object references with descriptive attributes or relational properties, requiring genuine semantic grounding:

- *Color attributes*: "carrot" → "orange vegetable", "eggplant" → "purple vegetable"
- *Shape attributes*: "can" → "cylinder object", "bowl" → "round container"
- *Object attributes*: "alphabet soup" → "canned food", "moka pot" → "coffee maker"
- *Spatial relations*: "bottom drawer" → "lower compartment", "back compartment" → "rear section"
- *Functional properties*: "caddy" → "storage container",

Table 5. **Instruction Rephrasing Taxonomy.** Representative examples from LIBERO and SimplerEnv preserving task semantics and success criteria.

Level	Category	Original Instruction	Rephrased Variant
Lexical Substitution	Verb variation	Pick up the book and place it in the back compartment of the caddy	Grasp the book and put it in the rear section of the caddy
	Noun variation	Put the spoon on the plate	Put the spoon on the dish
	Color adjective	Put the yellow and white mug in the microwave	Put the bright yellow mug in the microwave
Semantic Rephrasing	Color attribute	Put the carrot on the plate	Put the orange vegetable on the plate
	Shape attribute	Pick the can from the basket	Pick the cylinder object from the basket
	Object attribute	Put the alphabet soup in the basket	Put the canned food in the basket
	Functional property	Put the yellow and white mug in the microwave and close it	Put the yellow and white mug in the heating appliance and close it
	Spatial relation	Put the black bowl in the bottom drawer of the cabinet	Put the black bowl in the lower compartment of the cabinet
	Negation	Put both moka pots on the stove	Put the coffee makers that are not in the sink on the stove

"microwave" → "heating appliance"

- *Negation*: "put both moka pots on the stove" → "put the coffee makers that are not in the sink on the stove"

All rephrased instructions are generated using GPT-4 with manual verification to ensure semantic equivalence and correct object references. For LIBERO, we generate 5 rephrased variants per task across all four suites, yielding 200 test cases (10 tasks × 4 suites × 5 variants). For SimplerEnv, we generate 5 rephrased variants per task across 4 WidowX tasks, evaluated over 24 episodes each. Table 5 provides representative examples across both levels and subcategories. All variants maintain identical task success criteria, isolating linguistic robustness from task difficulty.

E.2. Perturbations in Real-World Deployment

Beyond simulation robustness, we assess SemanticVLA's resilience to perceptual and task variations encountered in physical environments. Our real-world evaluation systematically introduces controlled perturbations across visual appearance and task configuration while maintaining core task specifications. This validates whether the dual-path architecture's visual grounding (via latent action tokens) and spatial reasoning (via explicit traces) remain stable under distributional shifts characteristic of unstructured environments.

Visual Generalization. We evaluate robustness to appearance variations through lighting and background changes. For lighting variations, we introduce colored spotlights (red, blue, green) and laser projections that create additional visual patterns on the workspace, testing whether latent tokens maintain stable visual grounding under novel illumination conditions that deviate from training distributions. For background changes, we replace the standard woodgrain tabletop with alternative surfaces including patterned tablecloths, textured mats, and high-contrast backgrounds, probing whether explicit trace reasoning—relying on spatial structure rather than background appearance—provides complementary robustness to latent tokens' visual features.

Task Generalization. We test generalization to novel object instances and cluttered workspace configurations. For novel objects, we substitute targets with unseen instances within the same functional category: different vegetable types for table sorting (bell peppers, zucchini), alternative utensils for food preparation (forks, knives), and varied geometric shapes for reasoning tasks (cylinders, spheres). Objects maintain similar affordances but differ in visual appearance and precise geometry, testing whether latent action tokens encode manipulation primitives generalizable across instance variations. For workspace clutter, we introduce 2-6 distractor objects with randomized positions, including similar objects (multiple blocks when task specifies one), partial occlusions (transparent containers), and visual clutter (task-irrelevant items). For reasoning tasks, we add salient but irrelevant visual stimuli (e.g., posters with unrelated numbers during math calculation), testing whether trace reasoning correctly attends to task-relevant spatial regions.

Language Perturbations. We apply instruction rephrasing through lexical substitution and semantic reformulation, similarly with Section E.1. Examples include: "Sort vegetables by type" → "Organize produce into categories"; "Put meat in brown cylinder" → "Place protein items in the container"; "Spell the word on whiteboard" → "Form the word using available letters"; "Answer math problem and

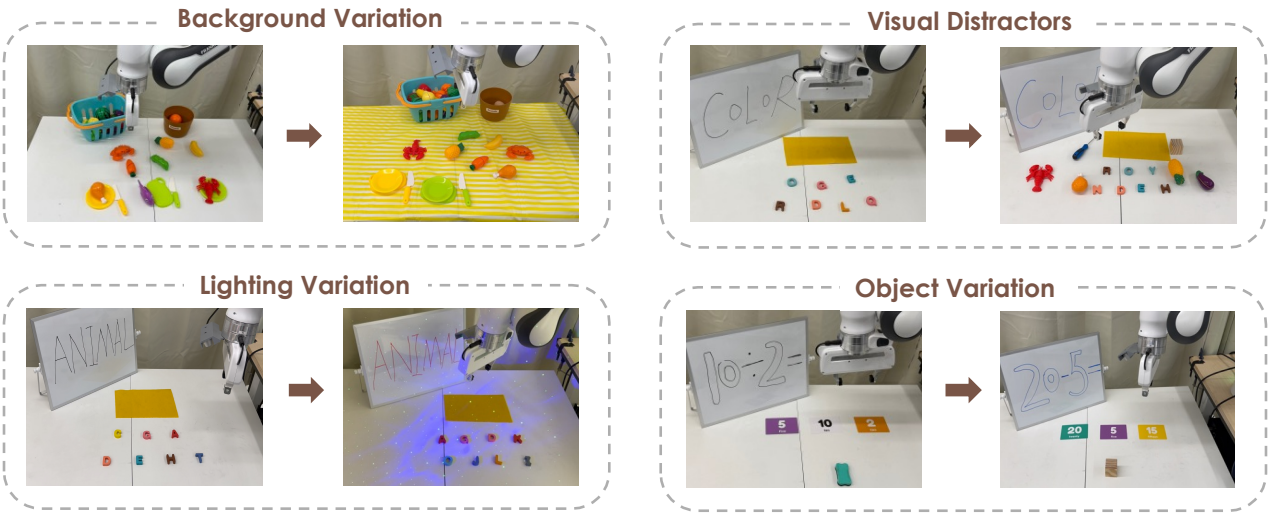

Figure 12. **Real-World Visual Perturbations.** Performance under background variation, visual distractors, lighting variation, and object variation across long-horizon sorting and reasoning-intensive manipulation tasks.

place accordingly" → "Calculate the result shown and position the object". These variations test whether the VLM genuinely reasons about task semantics rather than pattern-matching specific phrasings.

Each perturbation type is evaluated across all 10 real-world tasks (2 long-horizon, 2 reasoning-intensive, each with 5 variants) with 20 rollouts per configuration. Success rates are reported per perturbation category. Figure 7 illustrates representative examples showing how dual-path reasoning maintains robustness: explicit traces provide geometric anchors invariant to appearance changes, while latent tokens adaptively ground spatial plans in cluttered visual contexts.

F. Additional Experiment Results

While prior work evaluates trace quality primarily through downstream task success [31], we provide direct quantitative assessment to demonstrate that our dual-path architecture produces more accurate spatial reasoning. By measuring trace prediction accuracy under open-vocabulary instructions, we validate that explicit trace reasoning preserves VLM's native spatial grounding capabilities while achieving superior coordinate-level precision compared to baseline approaches.

Evaluation Protocol. We assess trace accuracy on our real-world Franka manipulation tasks, where test episodes use instruction rephrasing unseen during training to probe open-vocabulary spatial grounding. For each predicted trace $\hat{\tau} = (\hat{p}_1, \ldots, \hat{p}_L)$, we compute pixel-level deviations from manually annotated ground truth waypoints $\tau = (p_1, \ldots, p_L)$ by measuring Euclidean distance in the exterior camera view at 256×256 resolution. We report mean pixel error alongside accuracy thresholds at 10px, 30px, and 60px tolerance, reflecting increasing levels of spatial precision required for successful manipulation. Ground truth traces are extracted from collected real-robot demonstration trajectories, providing reliable spatial reference independent of task completion metrics.

Results and Analysis. Table 6 presents trace accuracy across our real-world evaluation suite. SemanticVLA achieves a mean error of 16.8 pixels with 68.9% of waypoints falling within 30-pixel tolerance, substantially outperforming MolmoAct and Magma which exhibit mean errors of 31.6 and 39.2 pixels respectively, with accuracy below 40% at the same threshold. This improvement validates that our trace-guided latent pretraining provides stronger geometric priors for spatial reasoning, enabling more precise coordinate generation through the VLM's language interface.

The ablation study reveals two complementary benefits of our dual-path design. Removing co-training between trace and latent tokens increases mean error to 18.9 pixels, demonstrating that joint optimization stabilizes trace prediction by leveraging latent representations as auxiliary spatial anchors—the discrete latent codes, pretrained on pure geometric patterns, act as consistency constraints that regularize coordinate generation against VLM's autoregressive sampling noise. Conversely, removing latent action planning while maintaining co-trained traces yields minimal accuracy degradation but dramatic task success drops from 62.3% to 48.0%, confirming that latent tokens primarily contribute through execution-level compensation rather than improving trace precision itself.

Task category breakdown shows that long-horizon tasks

Table 6. **Trace prediction accuracy and task success rates on real-world Franka experiments.** SemanticVLA achieves superior trace accuracy through trace-guided latent pretraining. Co-training between trace and latent tokens stabilizes coordinate prediction, while latent action planning compensates for remaining imprecision during execution.

Method	Mean Error (px) ↓	Acc@10px ↑	Acc@30px ↑	Acc@60px ↑	Task Success (%) ↑
MolmoAct [31]	31.6	18.3	38.7	81.3	40.3
Magma [49]	39.2	14.5	28.5	72.6	35.9
SemanticVLA (Full)	**16.8**	**32.7**	**68.9**	**96.3**	**62.3**
w/o Co-training	18.9	26.4	61.2	94.8	54.7 (−7.6)
w/o Latent Action Planning	17.3	30.2	65.4	95.8	48.0 (−14.3)
Task category breakdown (Full model):					
Long-horizon (Avg)	15.3	36.8	72.1	97.2	73.0
Food Preparing	17.1	31.4	68.9	96.5	77
Desktop Sorting	13.5	42.2	75.3	97.9	69
Reasoning-intensive (Avg)	18.3	28.6	65.7	95.4	51.5
Math Calculation	16.9	33.1	69.2	96.8	58
Word Spelling	19.7	24.1	62.2	94.0	45

achieve higher trace accuracy with mean error of 15.3 pixels, as larger manipulation primitives provide clearer visual targets for spatial grounding. Reasoning-intensive tasks prove more challenging at 18.3 pixels mean error, where the VLM must integrate symbolic reasoning with spatial localization—yet these scenarios demonstrate the largest task success improvements when latent action tokens stabilize against coordinate-level noise.

G. Qualitative Analysis

To assess SemanticVLA's reasoning capabilities and robustness boundaries, we present representative rollouts from real-world deployment across reasoning-intensive manipulation scenarios. Figures 13-16 demonstrate the system's core strength: handling tasks that demand genuine compositional reasoning—visual interpretation, symbolic calculation, linguistic understanding, and spatial planning—rather than memorized action patterns.

Reasoning-intensive tasks. Word spelling (Figure 13) and math calculation (Figure 14) require the VLM to perform explicit reasoning before manipulation. Successful cases (top rows, green boxes) show robust performance across lighting variations: the system correctly identifies letters to spell "RED" and "DOG", and accurately computes arithmetic results to place objects at correct positions. These successes validate that our dual-path architecture preserves VLM's native reasoning capabilities—the model genuinely "understands" whiteboard content rather than pattern-matching visual features to action sequences. However, failure cases (bottom rows, red boxes) reveal brittleness under extreme visual corruptions. Severe lighting interference causes visual grounding errors where the system misidentifies letters or places objects at incorrect spatial locations despite potentially correct arithmetic reasoning. These failures expose that while trace coordinates provide interpretable spatial plans, execution remains vulnerable when perceptual inputs severely degrade both VLM spatial grounding and latent action features simultaneously.

Long-horizon compositional tasks. Table sorting (Figure 15) and food preparation (Figure 16) demand sustained reasoning across sequential subtasks without explicit subgoal annotations. Successful cases demonstrate stable semantic categorization: the system correctly distinguishes vegetables from meat across workspace clutter, and executes multi-step sandwich assembly maintaining coherent sequencing despite layout variations. Critically, these tasks require the model to maintain task-level semantic understanding throughout extended manipulation sequences—sorting all vegetables before switching categories, or completing ingredient placement in logical order. Failure cases reveal reasoning fragility under object configuration shifts: Figure 15 top row shows category discrimination failure where the system incorrectly picks chicken drumstick when instructed to sort vegetables, while Figure 16 bottom row demonstrates placement errors induced by distractor container repositioning. These failures suggest that while our approach handles moderate workspace variations, substantial scene layout changes can disrupt the learned correspondence between semantic categories and visual grounding.

The qualitative evidence confirms SemanticVLA's advantage on reasoning-intensive manipulation, where explicit trace reasoning surfaces spatial planning through VLM's compositional understanding, while latent tokens provide execution-level robustness through visually-grounded primitives. However, the failure modes underscore that extreme perceptual corruption or ambiguous scene configurations remain challenging when neither pathway can reliably establish semantic grounding. Future work addressing these limitations could explore more robust visual encoders, adaptive perception preprocessing, or hierarchical reasoning structures that enable recovery from intermediate failures.

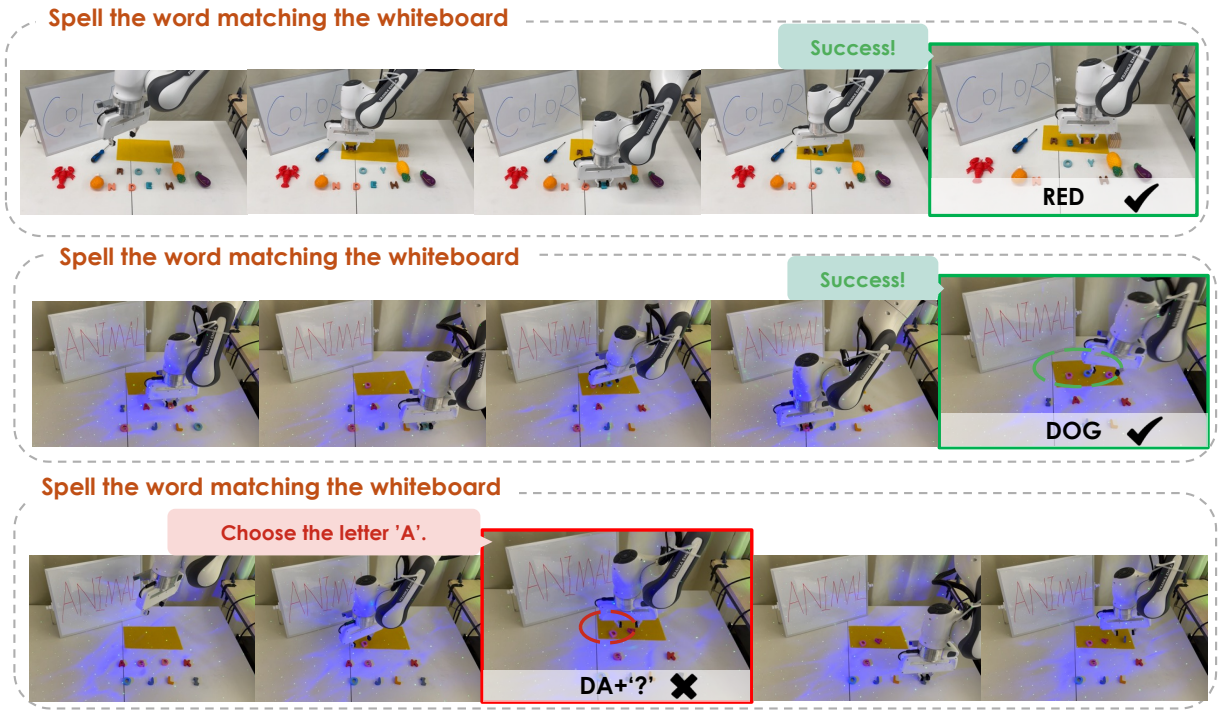

Figure 13. **Word spelling task case study.** Successful cases (top two rows, green) show robust letter identification under lighting variations. Failure case (bottom, red) reveals visual grounding errors under severe illumination interference.

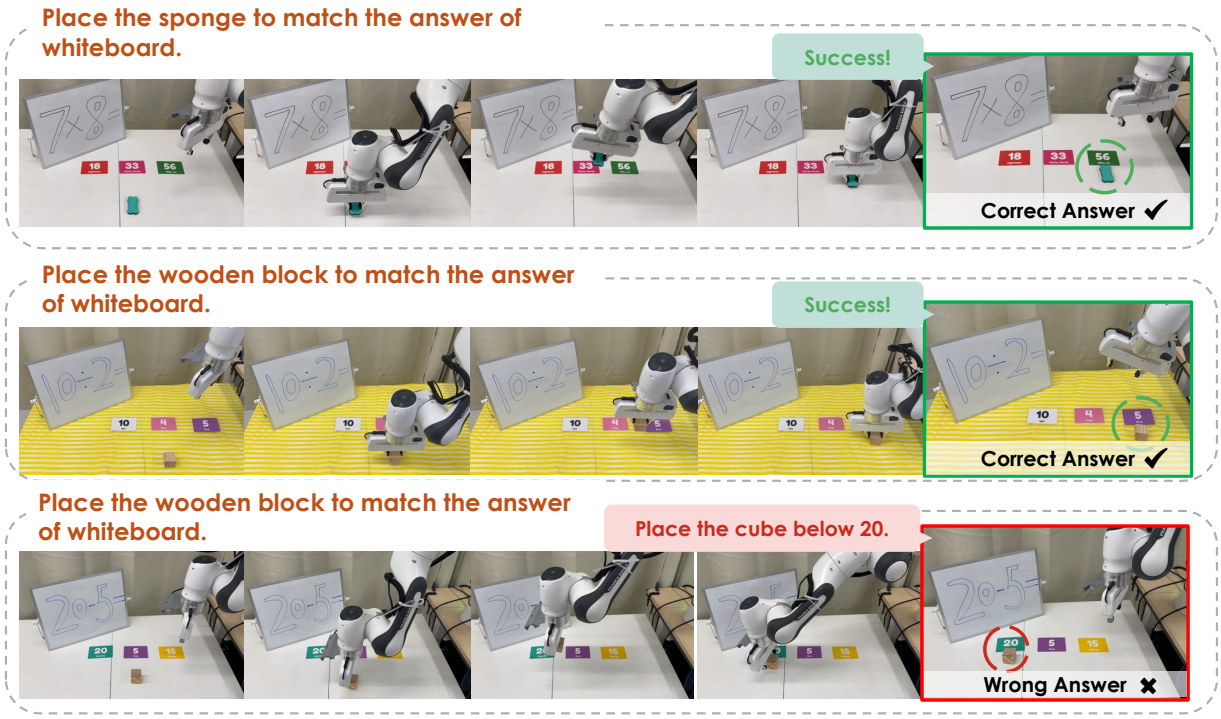

Figure 14. **Math calculation task case study.** Successful cases (top two rows, green) demonstrate genuine reasoning and accurate spatial placement. Failure case (bottom, red) shows spatial reasoning degradation under extreme visual corruptions.

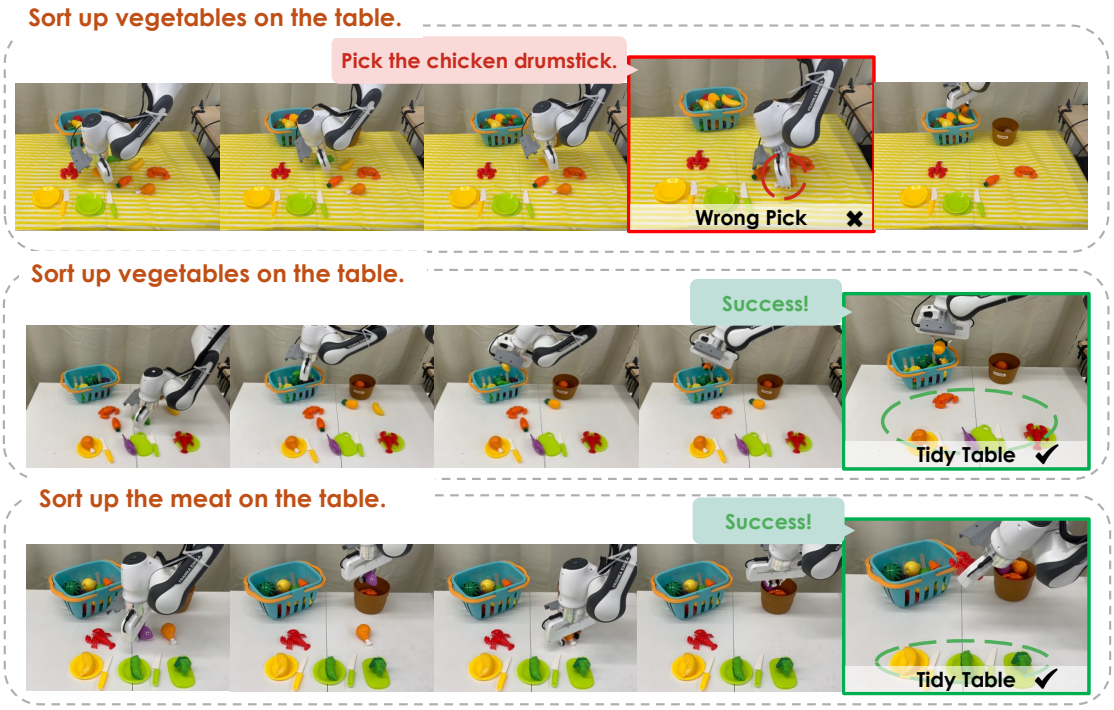

Figure 15. **Long-horizon sorting task case study.** Top row (red) shows category discrimination failure. Middle and bottom rows (green) demonstrate sustained common reasoning for multi-step object categorization despite workspace clutter.

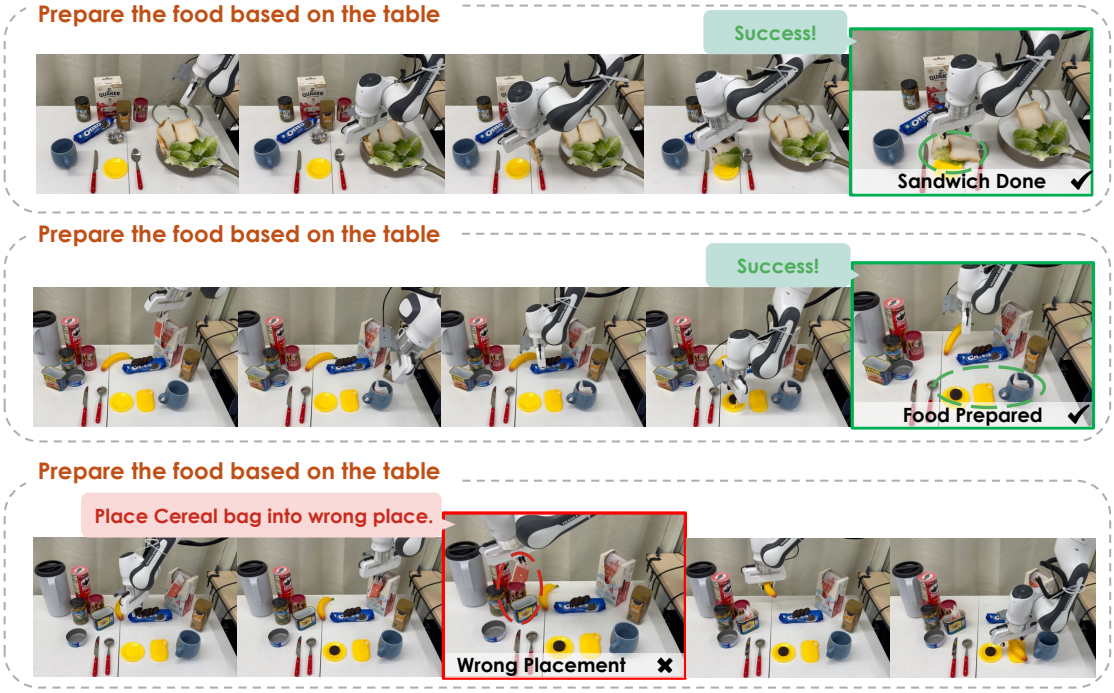

Figure 16. **Food preparation task case study.** Top two rows (green) show successful multi-step sandwich assembly across layout variations. Bottom row (red) shows placement errors induced by distractor container repositioning, exposing reasoning fragility under object configuration shifts.

H. Limitations & Future Work

Limitations. While SemanticVLA demonstrates strong performance on reasoning-intensive manipulation, several aspects merit consideration. Our explicit trace reasoning operates on 2D image coordinates, inherently lacking depth information for tasks requiring precise 3D spatial understanding such as insertion or fine-grained assembly. When multiple objects overlap or precise pose alignment is required, 2D coordinates cannot capture depth ordering or 6-DOF pose constraints, limiting spatial reasoning beyond planar trajectories—though the dual-path architecture partially compensates through visually-grounded latent tokens that implicitly encode depth cues. Additionally, while our compact trace representation achieves substantially higher inference efficiency than verbose natural language reasoning chains, explicit coordinate generation still introduces computational overhead compared to direct latent action prediction, as the VLM must autoregressively generate coordinate sequences. Although our decoupling framework amortizes this cost across multiple execution steps, inference efficiency remains a trade-off for interpretable spatial planning.

Future Work. Several promising directions could address these limitations and extend our framework's capabilities. Extending trace representations to 3D space—such as predicting depth or generating 3D waypoint clouds—could enhance spatial reasoning for manipulation requiring precise pose control in complex scenes. More efficient action decoding through adaptive trace discretization or hierarchical reasoning could optimize inference by reserving fine-grained waypoints only for complex manipulation phases. Additionally, incorporating reinforcement fine-tuning could improve robustness through trial-and-error learning, while hybrid architectures that selectively trigger explicit language reasoning for ambiguous scenarios could balance trace-based efficiency with adaptive deliberation. More broadly, we envision continued exploration of how System 1 (fast, implicit) and System 2 (slow, deliberate) reasoning can be effectively integrated in embodied AI—preserving VLM's compositional understanding while maintaining action experts' precise control. Our dual-path paradigm combining explicit trace reasoning with implicit latent action planning offers a principled framework that we hope will inspire the community to rethink how vision-language models should interface with robotic control, treating VLMs as reasoners rather than action memorizers.

References

[1] Figure AI. Helix, 2024. 2

[2] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. arXiv preprint arXiv:2403.01823, 2024. 3

[3] 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. 2, 6

[4] 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. 2, 5, 6, 7

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817, 2022. 1, 2, 5, 6

[6] 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. 1

[7] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. arXiv preprint arXiv:2503.06669, 2025. 2

[8] 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. 3, 5, 6, 7

[9] Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. arXiv preprint arXiv:2510.13778, 2025. 2, 4

[10] Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. arXiv preprint arXiv:2412.04445, 2024. 2

[11] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In Proceedings of Robotics: Science and Systems (RSS), 2023. 5

[12] Open X-Embodiment Collaboration, Abby O'Neill, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023. 2, 5

[13] Zichen Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. Advances in Neural Information Processing Systems, 37:33933–33961, 2024. 3

[14] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli

VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. arXiv preprint arXiv:2409.17146, 2024. 3

[15] 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. 2

[16] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In Proceedings of the 32nd ACM International Conference on Multimedia, pages 11198–11201, 2024. 1

[17] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. arXiv preprint arXiv:2109.13396, 2021. 2, 5, 4

[18] Irving Fang, Juexiao Zhang, Shengbang Tong, and Chen Feng. From intention to execution: Probing the generalization boundaries of vision-language-action models. arXiv preprint arXiv:2506.09930, 2025. 2, 7, 5

[19] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. arXiv preprint arXiv:2510.13626, 2025. 2, 7, 5

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. ICLR, 1(2):3, 2022. 5

[21] 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. 6

[22] 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. 2

[23] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In Conference on Robot Learning, pages 991–1002. PMLR, 2022. 2, 5

[24] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6013–6022, 2025. 5, 3

[25] 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 Forty-first International Conference on Machine Learning, 2024. 5

[26] 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. arXiv preprint arXiv:2403.12945, 2024. 2, 5

[27] 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. 1, 2, 6, 7

[28] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645, 2025. 6, 4

[29] Longxin Kou, Fei Ni, Yan Zheng, Jinyi Liu, Yifu Yuan, Zibin Dong, and Jianye Hao. Kisa: A unified keyframe identifier and skill annotator for long-horizon robotics demonstrations. In Forty-first International Conference on Machine Learning, 2024. 5

[30] Longxin Kou, Fei Ni, Yan Zheng, Peilong Han, Jinyi Liu, Haiqin Cui, Rui Liu, and Jianye Hao. Roboannotatorx: A comprehensive and universal annotation framework for accurate understanding of long-horizon robot demonstration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10353–10363, 2025. 1

[31] 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, 2025. 3, 4, 5, 6, 7, 8

[32] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. arXiv preprint arXiv:2403.03181, 2024. 3

[33] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. arXiv preprint arXiv:2405.05941, 2024. 6, 4

[34] 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, 2025. 3

[35] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. arXiv preprint arXiv:2508.05635, 2025. 3

[36] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. arXiv preprint arXiv:2505.11917, 2025. 2

[37] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems, 36, 2024. 6, 4

[38] Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. Quest: Self-supervised skill abstractions for learning continuous control. Advances in Neural Information Processing Systems, 37:4062–4089, 2024. 3

[39] Fei Ni, Jianye Hao, Yao Mu, Yifu Yuan, Yan Zheng, Bin Wang, and Zhixuan Liang. Metadiffuser: Diffusion model as conditional planner for offline meta-rl. In International Conference on Machine Learning, pages 26087–26105. PMLR, 2023. 1, 2

[40] Fei Ni, Jianye Hao, Shiguang Wu, Longxin Kou, Jiashun Liu, Yan Zheng, Bin Wang, and Yuzheng Zhuang. Generate subgoal images before act: Unlocking the chain-of-thought reasoning in diffusion model for robot manipulation with multimodal prompts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13991–14000, 2024. 3

[41] Fei Ni, Jianye Hao, Shiguang Wu, Longxin Kou, Yifu Yuan, Zibin Dong, Jinyi Liu, MingZhi Li, Yuzheng Zhuang, and Yan Zheng. Peria: Perceive, reason, imagine, act via holistic language and vision planning for manipulation. Advances in Neural Information Processing Systems, 37:17541–17571, 2024. 3

[42] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In Proceedings of Robotics: Science and Systems, Delft, Netherlands, 2024. 6

[43] 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. 4

[44] 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. 2, 6

[45] Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, et al. Eo-1: Interleaved vision-text-action pretraining for general robot control. arXiv preprint arXiv:2508.21112, 2025. 2

[46] 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. 6

[47] Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. Advances in neural information processing systems, 35:22955–22968, 2022. 3

[48] Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. arXiv preprint arXiv:2507.01016, 2025. 2, 3, 6

[49] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. arXiv preprint arXiv:2502.13130, 2025. 3, 6, 7, 8

[50] Shuai Yang, Hao Li, Yilun Chen, Bin Wang, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Instructvla: Vision-language-action instruction tuning from understanding to manipulation. arXiv preprint arXiv:2507.17520, 2025. 2, 4, 5

[51] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. arXiv preprint arXiv:2410.11758, 2024. 2, 3

[52] Yifu Yuan, Haiqin Cui, Yaoting Huang, Yibin Chen, Fei Ni, Zibin Dong, Pengyi Li, Yan Zheng, and Jianye Hao. Embodied-r1: Reinforced embodied reasoning for general robotic manipulation, 2025. 3

[53] Zichen Zhang, Yunshuang Li, Osbert Bastani, Abhishek Gupta, Dinesh Jayaraman, Yecheng Jason Ma, and Luca Weihs. Universal visual decomposer: Long-horizon manipulation made easy. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 6973–6980. IEEE, 2024. 5

[54] 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 Proceedings of the Computer Vision and Pattern Recognition Conference, pages 1702–1713, 2025. 3, 6

[55] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. arXiv preprint arXiv:2304.13705, 2023. 3

[56] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé 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. 3, 6

[57] Zhongyi Zhou, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. Vision-language-action model with open-world embodied reasoning from pretrained knowledge. arXiv preprint arXiv:2505.21906, 2025. 2

[58] Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. arXiv preprint arXiv:2502.14420, 2025. 2