GE-ACT 2.0: PRETRAINING AND SCALING A WORLD–ACTION MODEL FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

World–action models (WAMs) commonly adapt video generators pretrained on natural video, leaving open how their manipulation capability scales when they are pretrained from scratch on manipulation data. To study this question, we introduce Genie Envisioner Act 2.0 (GE-Act 2.0), a world–action model that combines a control-oriented autoencoder (CoAE), a single-step visual planner (SVP), and an inverse dynamics model (IDM). CoAE learns compact representations for robot control, while the SVP and IDM are pretrained from scratch on manipulation data. Knowledge-aligned selective optimization (KASO) jointly trains the SVP and IDM using generated futures compatible with recorded actions. These components generate a 52-action chunk in 104 ms on a single RTX 5090, supporting real-time control at 30 Hz. We study zero-shot out-of-distribution (OOD) generalization without per-task fine-tuning on 100 real-robot tasks across 20 manipulation skill groups and two embodiments. Scaling co-training data from 300 to 30,000 hours substantially improves success rates on both G1-OP and G2-90D, despite the latter contributing less than 2% of the training mixture. Beyond total data scale, our skill-level coverage analysis reveals a strong correlation between skill-specific training hours and zero-shot OOD success. We further study fine-grained instruction following, achieving $\geq$90% Follow Score for object identity, color, shape, and position. On RoboTwin-Clean2Rand and GenieSim-Instruction, GE-Act 2.0 outperforms strong recent baselines including $\pi_{0.5}$, further demonstrating its generalization ability. We will release pretrained checkpoints and code.

1 INTRODUCTION

Scaling pretraining has enabled broad capabilities in language and vision, motivating similar efforts in robot manipulation. Vision–language–action (VLA) policies map observations and language instructions directly to actions (Brohan et al., 2022; Black et al., 2024), while world–action models (WAMs) predict visual futures conditioned on these inputs and use the predictions to guide action generation (Wang et al., 2026a; Liao et al., 2025; Ye et al., 2026b; Wang et al., 2026b). Besides making the model's visual predictions inspectable, this formulation creates two routes to broader pretraining data. Visual-generation pretraining can capture physical and semantic structure from action-free video at a scale that robot demonstrations alone cannot provide; an inverse dynamics model (IDM) can potentially learn from robot trajectories that lack instruction or success annotations, including failed attempts and deployment rollouts.

However, existing WAM research typically builds on pretrained video generators (Wan Team et al., 2025; NVIDIA et al., 2025a) and focuses on coupling them with action models (Liao et al., 2025; Ye et al., 2026b), while pretraining and scaling the WAM as a whole remain less explored. This leaves two system-level questions unresolved: how to pretrain visual generation and inverse dynamics on their respective data before connecting them, and how WAM capability scales with manipulation data when all trainable visual-generation and action components are initialized from scratch, so that measured capability reflects the manipulation data rather than an externally pretrained generator.

We introduce Genie Envisioner Act 2.0 (GE-Act 2.0), a world–action system that connects separately pretrained visual prediction and inverse dynamics. A control-oriented autoencoder (CoAE) learns compact visual representations that retain information for future prediction and robot control. A single-step visual planner (SVP) predicts a complete visual future in one differentiable pass, allowing

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.

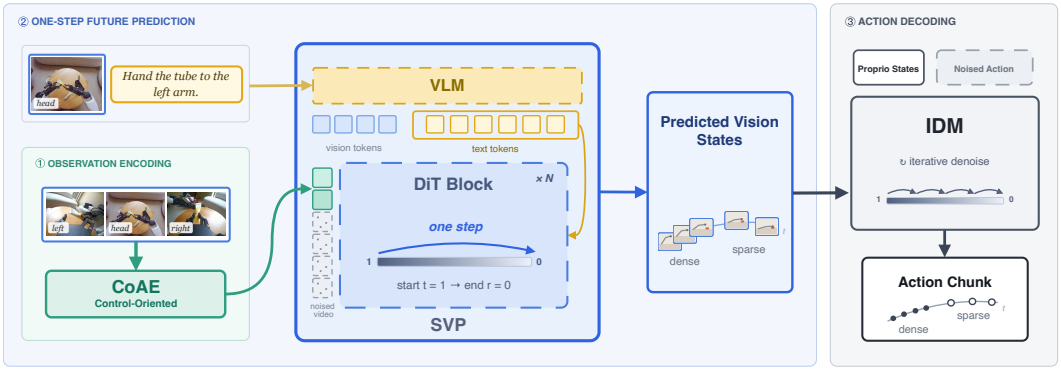

Figure 1: GE-Act 2.0. CoAE encodes multi-view observations; the SVP predicts future visual states in one flow-generator pass; and the IDM produces dense and sparse actions. Only the dense action chunk is executed.

the visual generator and IDM to be pretrained from scratch on complementary manipulation data before joint training. During co-training, a plausible predicted future may depict a different behavior from the recorded action, creating a supervision mismatch we call the validity gap. Knowledge-aligned selective optimization (KASO) addresses this mismatch by selecting futures that the active IDM judges compatible with the recorded action. Together, compact visual latents and one-pass future prediction support a full observation-to-action pipeline that generates 52 executable actions in 104 ms on a single RTX 5090.

To examine how this pretraining translates into manipulation capability, we evaluate what GE-Act 2.0 can do without downstream adaptation. Evaluations that fine-tune policies on target tasks or embodiments combine capabilities acquired during pretraining with those added downstream (Zhang et al., 2026; Kim et al., 2024b; Black et al., 2024). We instead evaluate pretrained checkpoints directly under zero-shot out-of-distribution (OOD) conditions, keeping the protocol fixed across training-data scales. The evaluation covers 100 tasks across 20 skill groups and two embodiments, together with tests of fine-grained instruction following.

Under this fixed evaluation protocol, increasing co-training data from 300 to 30,000 hours yields broad improvements in zero-shot OOD manipulation across both embodiments, including G2-90D, which contributes less than 2% of the mixture. Our skill-level coverage analysis reveals a strong correlation between skill-specific training hours and success, suggesting that skill coverage, alongside total data scale, is a strong predictor of which skills a pretrained model acquires. Fine-grained instruction-following tests further demonstrate language grounding in cluttered scenes, while GE-Act 2.0 outperforms strong baselines including $\pi_{0.5}$ on RoboTwin-Clean2Rand and GenieSim-Instruction. Appendix A reviews related work.

Contributions.

- **A practical world–action pretraining system.** GE-Act 2.0 combines control-oriented visual compression through CoAE, single-step future prediction through SVP, and action-compatible co-training through KASO to learn from complementary manipulation data and support real-time chunked control. We plan to **publicly release the pretrained checkpoints**[1] together with code and related materials to support further research.

- **An empirical study of pretrained manipulation capability.** We characterize zero-shot OOD skill scaling, the relationship between training coverage and success, and instruction following across real-robot evaluations on two embodiments, complemented by standardized simulation comparisons.

[1]Anonymous project homepage: `https://anonymous.4open.science/w/supplementary-website-769F/`.

2 GE-ACT 2.0: A SYSTEM FOR WORLD–ACTION PRETRAINING

GE-Act 2.0 connects a control-oriented autoencoder (CoAE), a single-step visual planner (SVP), and an inverse dynamics model (IDM) (Figure 1). CoAE encodes the current multi-view observations; the SVP predicts future visual latents conditioned on the scene and instruction; and the IDM converts current and predicted states, together with proprioception, into actions. CoAE is trained first and frozen. The SVP and IDM then learn separately from instruction–video and observation–action data, respectively, before fully labeled demonstrations connect them through KASO co-training. This permits pretraining on videos without actions and robot trajectories without language or success annotations. At deployment, compact CoAE latents, one-pass visual prediction, and five-step IDM decoding let the full pipeline produce 52 executable actions from three-camera observations in 104 ms on a single RTX 5090; each chunk spans 1.7 seconds at 30 Hz, supporting real-time control through action chunking.

2.1 COMPACT VISUAL REPRESENTATION

CoAE provides a compact, decodable visual interface for future prediction and action recovery (Figure 2). CoAE applies 64×64 spatial downsampling to produce compact visual latents. Training combines pixel, perceptual, and adversarial reconstruction losses with feature alignment to frozen SigLIP 2, V-JEPA 2.1, and DINOv3 teachers (Tschannen et al., 2025; Mur-Labadia et al., 2026; Siméoni et al., 2025). The reconstruction decoder preserves visual decodability, while alignment supervises semantic and spatiotemporal information. Architecture, initialization, and full objectives are in Appendix B; Section 5.1 evaluates the effects of feature alignment on action prediction and semantic retention.

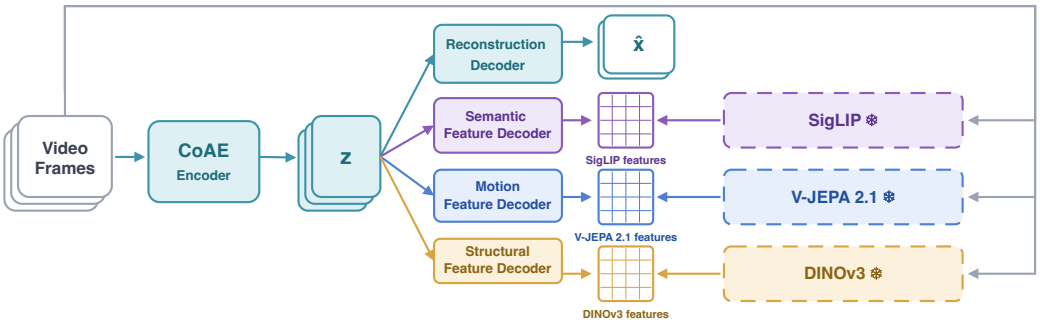

Figure 2: CoAE architecture and training objectives. A reconstruction decoder provides pixel-level supervision, while three alignment heads match language-aligned, spatiotemporal, and dense visual features from frozen visual teachers.

2.2 SINGLE-STEP FUTURE PREDICTION

The SVP generates a complete visual future in one differentiable pass, allowing an action loss to update the visual generator during co-training. We randomly initialize a multi-view diffusion transformer (DiT) (HaCohen et al., 2024) and pretrain it on manipulation video using a conditional MeanFlow objective (Geng et al., 2025; 2026). At inference, observed-frame latents remain fixed, while future-frame latents are initialized with Gaussian noise. A single denoising step transforms this noise into predicted future latents:

$$\hat{z}^{(i)} = z_1^{(i)} - u_\theta^{(i)}(z_1, r = 0, t = 1), \qquad i \in \mathcal{F}. \tag{1}$$

Here, z_1 contains observed latents and future-frame noise, $\mathcal{F}$ indexes future frames, and u_θ is the learned mean-velocity field. Conditional paths and training objectives are detailed in Appendix C.3.

Following Act2Goal (Zhou et al., 2025), the SVP predicts four dense frames over the 1.73-second action horizon and two sparse frames over the remaining clip, capturing near-term interaction and longer-term intent. A frozen Qwen3.5 VLM (Qwen Team, 2026) encodes the head view and instruction; its image-conditioned text features condition the generator. Schedules and conditioning details are in Appendices C.1 and C.2.

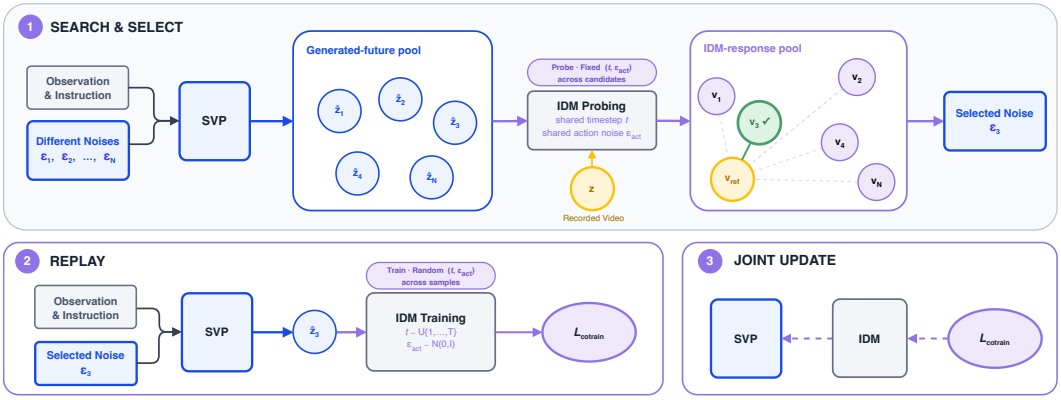

Figure 3: The KASO pipeline. Candidate visual predictions are sampled from the SVP and scored without gradients; selected noises are then replayed exactly so that the joint action loss updates both the IDM and the SVP.

2.3 Action-Compatible Co-Training

Our IDM is a flow-matching model conditioned on recorded visual frames and proprioceptive states, generating action sequences from noise. It predicts dense executable actions over the near-term horizon and sparse auxiliary actions over the longer horizon. We randomly initialize and pretrain the IDM using recorded future states, then connect it to the SVP through co-training, replacing recorded futures with generated predictions so that action supervision can update both components.

Replacing recorded future states with generated predictions introduces a potential mismatch between the predicted future and the recorded action used for supervision. For example, a generated future may depict a valid grasp from one side of an object, while the recorded action approaches from another. We call this mismatch the validity gap. We design knowledge-aligned selective optimization (KASO) to address it by selecting generated futures that are compatible with the recorded action according to the current IDM, so that only action-consistent knowledge from the visual generator is aligned into the IDM.

As illustrated in Figure 3, KASO samples several future predictions and compares the IDM's response to each with its response to the recorded future. It selects the candidate with the closest response, then regenerates that candidate using the same noise so that the action loss can update both the SVP and IDM. Selection operates in action space, allowing visually different futures to support the same behavior. Co-training also retains the visual-generation and recorded-pair IDM objectives:

$$\mathcal{L}_{\text{KASO}} = \mathcal{L}_{\text{SVP}} + \mathcal{L}_{\text{IDM}} + \mathcal{L}^{\text{sel}}_{\text{E2E}}. \tag{2}$$

The first term supervises visual generation, while the latter two supervise actions conditioned on recorded and selected generated futures, respectively. The toy study and matched robot ablation evaluate these choices in Section 5.2. Full objectives and selection details are provided in Appendix B, with pseudocode in Appendix C.4.

3 Zero-Shot OOD Evaluation on Real Robots

We evaluate GE-Act 2.0 under zero-shot out-of-distribution (OOD) conditions, deploying pretrained checkpoints without evaluation-task fine-tuning or per-task checkpoint selection. Test scenes, backgrounds, lighting configurations, and object instances are excluded from pretraining and co-training, so the suite measures how skills covered in training generalize to unseen environments and objects. The atomic-task evaluation covers 100 tasks in 20 skill groups on G1-OP and G2-90D and examines co-training data scaling and skill coverage (Sections 3.1 and 3.2). The instruction-following evaluation tests fine-grained grounding and instruction conflicts (Section 3.3).

3.1 Scaling with Co-Training Data

We train four models with nested co-training pools of 300, 1,200, 5,000, and 30,000 hours that preserve the mixture composition as closely as possible. All four start from the same pretrained

Table 1: Skill-group zero-shot OOD success rate (%) across co-training data scales for G1-OP (more than 50% of the co-training mixture) and G2-90D (less than 2%). Each entry averages the group's tasks (task counts in parentheses; 10 trials per task); shading scales with SR. Task-level results appear in Table 9.

	G1-OP				G2-90D			
Skill group	0.3k h	1.2k h	5k h	30k h	0.3k h	1.2k h	5k h	30k h
Pick (13)	40.8	53.1	63.1	80.0	48.5	70.0	67.7	66.2
Place (6)	65.0	73.3	85.0	88.3	58.3	56.7	68.3	80.0
Wipe (3)	30.0	33.3	43.3	76.7	0.0	0.0	20.0	56.7
Stack (7)	21.4	22.9	25.7	41.4	22.9	22.9	25.7	27.1
Pass (5)	34.0	52.0	60.0	70.0	8.0	18.0	22.0	34.0
Fold (4)	5.0	0.0	10.0	15.0	7.5	5.0	7.5	2.5
Straighten (5)	0.0	0.0	0.0	30.0	0.0	0.0	2.0	24.0
Pour (5)	12.0	24.0	30.0	54.0	6.0	10.0	8.0	20.0
Push (8)	15.0	21.2	21.2	45.0	0.0	10.0	8.8	16.2
Insert (6)	3.3	5.0	6.7	38.3	13.3	16.7	23.3	35.0
Close (3)	23.3	46.7	56.7	86.7	6.7	50.0	43.3	53.3
Separate (3)	0.0	0.0	0.0	43.3	0.0	13.3	10.0	16.7
Press (4)	0.0	0.0	5.0	17.5	0.0	0.0	2.5	7.5
Pull (6)	1.7	5.0	6.7	23.3	0.0	0.0	5.0	13.3
Hang (4)	5.0	0.0	2.5	15.0	0.0	7.5	12.5	17.5
Tear (2)	0.0	0.0	0.0	5.0	0.0	0.0	0.0	5.0
Flip (6)	10.0	20.0	21.7	28.3	0.0	10.0	15.0	20.0
Sweep (3)	0.0	0.0	0.0	3.3	0.0	0.0	0.0	6.7
Zip (3)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Shake/Stir (4)	0.0	0.0	5.0	12.5	0.0	17.5	20.0	32.5
Suite mean (100)	**17.1**	**22.6**	**27.3**	**44.1**	**13.4**	**21.0**	**23.5**	**31.1**

SVP and IDM and differ only in the fully labeled co-training pool; they share the same KASO recipe (Appendix E). Each is evaluated as a single checkpoint on all 100 tasks with fixed deployment settings, 10 trials per task, pre-specified terminal success criteria, and standardized resets. We report success rate (SR) at task, skill-group, and suite levels; Appendix F provides the taxonomy, protocol, and raw results. Table 1 reports SR for all 20 skill groups on both embodiments; Figure 9 plots the same trajectories, and the complete G1-OP task wall appears in Appendix G.

Across the four increasing data scales, suite-level SR rises from 17.1% to 22.6%, 27.3%, and 44.1% on G1-OP, and from 13.4% to 21.0%, 23.5%, and 31.1% on G2-90D. Both embodiments continue to improve substantially from 5,000 to 30,000 hours, with no observed saturation within the measured range. Improvements span 19 of 20 G1-OP skill groups and 18 of 20 G2-90D groups. The exceptions are Zip, with no successes across embodiments and scales, and Fold on G2-90D. Across all four scales, 87 and 78 task trajectories, respectively, are non-decreasing, indicating broad capability growth.

Scaling on the data-scarce embodiment. G2-90D gains 17.7 percentage points despite contributing less than 2% of the training mixture. Five of the six skill groups with fewer than two hours of G2-90D-specific data at the largest scale improve from zero to nonzero success, consistent with transfer from the shared multi-embodiment corpus. Appendix G provides representative rollouts and hardware-dependent comparisons.

3.2 Skill Coverage and Zero-Shot Success

Apparent motor difficulty alone does not explain the skill-level outcomes: demanding skills such as Stack, Pass, and Straighten improve with scale, while the similar motions of Wipe and Sweep yield markedly different success rates. We therefore examine how these outcomes relate to skill-specific training coverage. We audit coverage across all 20 skill groups by mapping step-captioned instruction–action segments from the multi-embodiment corpus to the evaluation taxonomy and converting annotated frame spans to hours. We then relate total training hours to the G1-OP 30,000-hour model's success using log hours and logit SR.

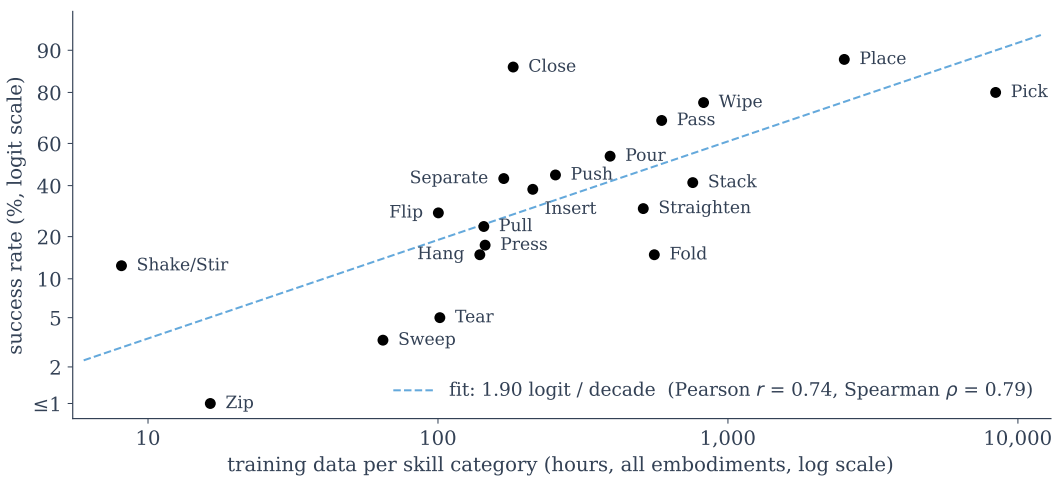

Figure 4: Skill-group training coverage and G1-OP zero-shot OOD success for the 30,000-hour model. Points represent all 20 skill groups. Axes show training hours across embodiments on a logarithmic scale and success rate on a logit scale.

Skill-specific training hours strongly correlate with zero-shot OOD success (Pearson $r = 0.74$; Spearman $\rho = 0.79$), and the fitted line rises by 1.90 logit units per decade of training hours. Substantial coverage accompanies meaningful success even for demanding skills: Pass, Straighten, and Stack receive 590.8, 510.5, and 756.2 training hours and reach 70.0%, 30.0%, and 41.4% SR, respectively. Among skills with similar motions, Wipe receives 824.1 hours and reaches 76.7% SR, whereas Sweep receives 64.6 hours and reaches 3.3%. Together with the scaling results, this analysis highlights skill coverage alongside total data scale as a direction for expanding the pretrained manipulation repertoire.

3.3 Instruction Following in Complex Scenes

We evaluate instruction grounding and execution using the final full-scale checkpoint on G1-OP without further adaptation. We use the familiar Pick and Place primitives to limit confounding from motor-skill difficulty. The suite tests object identity, color, size, shape, position, and order through 33 Pick and 26 Place instructions, each repeated five times (295 rollouts). Each scene permits multiple actions but only one instruction-specified target or destination. Follow Score requires contacting that target without contacting distractors; SR additionally requires completing the action. Figure 5 reports both metrics; Appendix F.3 provides the full protocol and instruction-level results.

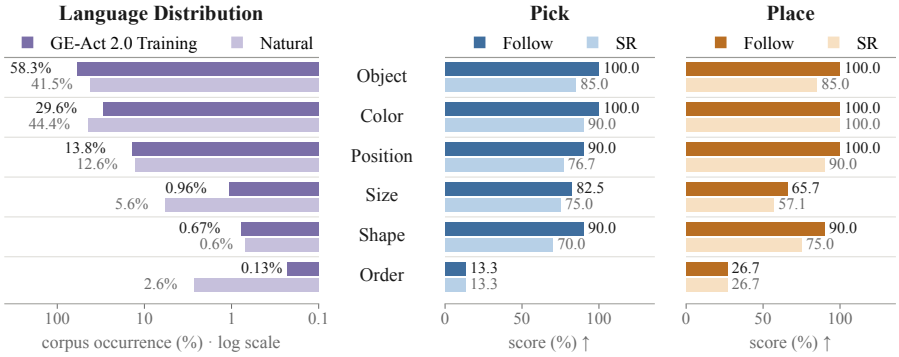

Figure 5: Fine-grained instruction following and language distribution across six grounding dimensions. Left: occurrence in 95,010 human-written RefCOCOg expressions (Natural) and the GE-Act 2.0 training corpus, shown on a logarithmic scale. Middle and right: Follow Score and full-task SR for Pick and Place, respectively. Dimensions are ordered by their occurrence in the GE-Act 2.0 training corpus; descriptor categories are counted independently and may overlap.

Follow Score reaches at least 90.0% for both Pick and Place on object identity, color, position, and shape, while size is less consistent and order remains the weakest dimension. Overall Follow Score is

83.1%, compared with 72.9% full-task SR, indicating that some failures occur after correct target grounding. Size, shape, and order references are rare in the training corpus (0.96%, 0.67%, and 0.13%, respectively), broadly mirroring the long-tailed distribution in RefCOCOg (Mao et al., 2016), a dataset of human-written expressions referring to objects in images. Order has the lowest Follow Score, consistent with its lowest representation in the training data.

Instruction-conflict stress tests. Qualitative stress tests examine behavioral and semantic conflicts: GE-Act 2.0 follows a changed instruction even when its hand is already close to the previously targeted object, and follows unusual object–destination instructions despite conventional scene associations. These cases probe whether language can override visually suggested actions. Appendix G.1 provides scenarios and rollouts.

4 OOD Generalization in Simulation

Our real-robot study characterizes how capability changes with training data under a fixed protocol. To position GE-Act 2.0 against other models, we use three simulation benchmarks that separate in-distribution training from OOD evaluation. We fine-tune GE-Act 2.0 on each benchmark's designated demonstrations and compare with baselines on held-out conditions without further adaptation, following the evaluation principle of Qwen-RobotManip (Yuan et al., 2026a).

RoboTwin Clean-to-Random. This benchmark tests generalization from controlled manipulation scenes to randomized environments (Chen et al., 2025b). We fine-tune on Easy/Clean demonstrations and evaluate individual perturbations and their combined Hard setting (Table 2).

GenieSim-Instruction. This benchmark tests ten instruction-following tasks involving attribute grounding, common-sense reference, and logical composition (Yin et al., 2026). We fine-tune on released demonstrations and evaluate OOD generalization in held-out scenes (Table 3).

LIBERO-Plus. LIBERO-Plus tests OOD robustness by applying seven types of perturbations to standard LIBERO tasks (Fei et al., 2025). We fine-tune on standard LIBERO demonstrations and evaluate all 10,030 perturbed instances without further adaptation (Table 4).

Table 2: RoboTwin Clean-to-Random success rate (%, ↑) after Easy/Clean adaptation. Baselines: StarVLA (StarVLA Community, 2026), GR00T N1.7 (NVIDIA et al., 2025b), and $\pi_{0.5}$ (Physical Intelligence et al., 2025). Column-best results are bolded.

Model	Easy	Background	Light	Clutter	Height	Hard
StarVLA	58.10	27.10	50.90	24.20	48.40	10.60
GR00T-N1.7	43.60	40.40	41.90	27.10	39.00	20.70
$\pi_{0.5}$	73.10	67.00	**69.20**	57.90	67.60	47.90
GE-Act 2.0	**76.71**	**70.28**	65.92	**66.52**	**70.39**	**60.52**

Table 3: OOD GenieSim-Instruction results (official normalized score, ↑). Additional baselines: π_0 (Black et al., 2024), LingBot-VA (Li et al., 2026), LingBot-VLA 2.0 (Wu et al., 2026), and ACoT-VLA (Zhong et al., 2026). Row-best results are bolded.

Task	π_0	GR00T-N1.7	LingBot-VA	LingBot-VLA 2.0	$\pi_{0.5}$	ACoT-VLA	**GE-Act 2.0**
pick_billiards_color	0.50	0.73	0.28	0.88	0.83	0.91	**0.95**
pick_block_color	0.49	0.80	0.22	0.80	0.87	**0.96**	0.92
pick_block_number	0.40	0.61	0.18	0.72	**0.82**	0.76	0.71
pick_block_shape	0.20	0.44	0.10	0.43	**0.55**	0.52	0.52
pick_block_size	0.34	0.62	0.08	0.68	0.79	**0.83**	0.76
pick_common_sense	0.19	0.68	0.08	0.63	0.59	0.66	**0.74**
pick_follow_logic_or	0.39	0.72	0.28	0.69	0.81	0.79	**0.89**
pick_object_type	0.42	0.72	0.13	**0.82**	0.80	0.80	0.78
pick_specific_object	0.37	0.69	0.23	0.87	0.81	0.83	**0.89**
straighten_object	0.38	0.45	0.50	0.56	**0.59**	0.51	0.54
Average	0.368	0.646	0.205	0.708	0.746	0.757	**0.770**

Table 4: LIBERO-Plus success rate (%, ↑). Overall weights perturbation axes by instance count. Baselines: Fast-WAM (Yuan et al., 2026b), StarVLA (StarVLA Community, 2026), and $\pi_{0.5}$ (Physical Intelligence et al., 2025). Column-best results are bolded.

Model	Camera	Robot	Language	Light	Background	Noise	Layout	Overall
Fast-WAM	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
StarVLA	52.5	49.8	**88.5**	95.7	**95.7**	73.0	76.9	74.1
$\pi_{0.5}$	78.4	**73.6**	80.8	**96.2**	94.1	89.0	**84.5**	**84.4**
GE-Act 2.0	**94.1**	50.7	81.4	94.0	60.5	**95.5**	83.1	80.4

GE-Act 2.0 leads in five of six RoboTwin conditions, reaching 60.52% on Hard versus 47.90% for $\pi_{0.5}$. On GenieSim-Instruction, it achieves the highest mean normalized score, 0.770 versus 0.757 for ACoT-VLA, and leads on four tasks. LIBERO-Plus shows a mixed robustness profile: GE-Act 2.0 achieves 80.4% overall, above StarVLA but below $\pi_{0.5}$, leading on camera and sensor-noise perturbations, with robot-state and background shifts the largest remaining gap.

5 Component Analysis and Ablations

5.1 CoAE: Compact Representation Diagnostics

We evaluate CoAE with two probes, comparing it with its three visual teachers and DC-AE (Chen et al., 2025a), which shares its spatial downsampling factor. For each probe, we freeze every encoder and train the same probe architecture with the same data splits and optimization (Appendix B.3).

Action recovery. An inverse-dynamics probe reconstructs actions from current and future representations, testing whether control-relevant information is preserved; we report mean absolute error (MAE). CoAE obtains 0.01673 action MAE, 13–31% higher than V-JEPA 2.1 and DINOv3 while using one-sixteenth as many tokens per frame, and outperforms DC-AE and SigLIP (Table 5).

Instruction grounding. A VLM-conditioned probe classifies GenieSim-Instruction observation–instruction pairs as matched (original instruction) or mismatched (instruction from another episode); caption-match error is the percentage of misclassified pairs. CoAE achieves the lowest error, 2.05%, compared with 2.71% for the next-best DC-AE. CoAE thus combines compact, decodable latents with lower errors on both probes than DC-AE at the same spatial downsampling factor.

Table 5: Controlled probes of five frozen visual representations. Spatial downsampling reports the height × width factors; channels denotes feature dimension. Lower is better for both probe metrics.

Representation	Spatial downsampling	Channels	Pixel decoder	Action MAE ↓	Caption-match error (%) ↓
CoAE	64×64	512	✓	0.01673	**2.05**
DC-AE	64×64	128	✓	0.02573	2.71
DINOv3	16×16	1024	✗	**0.01273**	2.75
V-JEPA 2.1	16×16	1024	✗	0.01487	4.15
SigLIP	32×32	2048	✗	0.03138	3.28

5.2 KASO: Action-Compatible Co-Training

Controlled toy system. We build a two-stage toy system: a flow-matching generator, analogous to the SVP, produces a visual state, and an MLP regressor, analogous to the IDM, maps it to an action. Recorded states form four Gaussian modes on the plane $z_3 = 0$ and map to two action modes near ± 1 (Appendix E.7). All strategies share initialization and optimization.

Decoupled Pretraining recovers all four video modes but yields a diffuse action density (Figure 6). Without co-training, its IDM sees only recorded states and cannot handle the small deviations that generation inevitably introduces. Each module works well in isolation; the failure lies at their interface, which is the gap KASO targets. E2E and E2E+PT pair each generated future with a recorded action drawn independently of it (Section 2.3), so every generated mode receives targets near both $+1$ and -1. Squared-error regression resolves this conflict by predicting their mean near zero, collapsing the action density. E2E additionally loses the video modes, which E2E+PT preserves through its retained pretraining losses. KASO supervises the IDM only on generated futures ranked compatible with the recorded action, so each mode keeps a consistent target, and it recovers both

the four video modes and the two action modes. Appendices E.7 and C.5 provide full details and additional analysis.

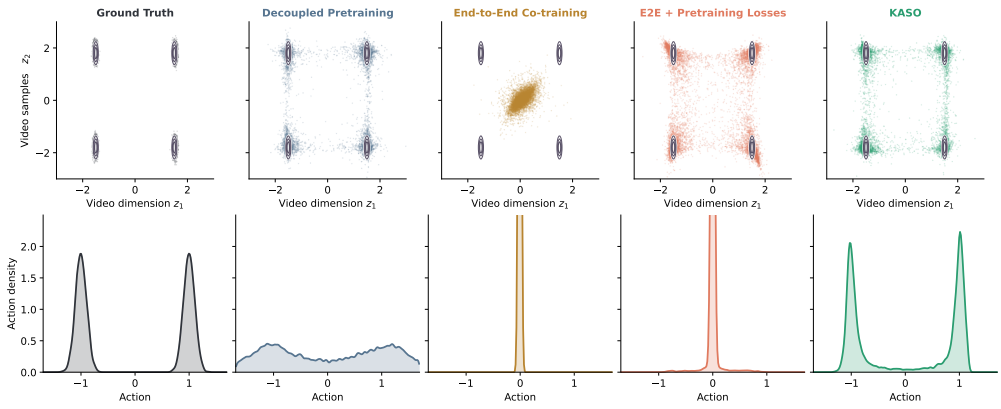

Figure 6: Controlled toy comparison. Top: generated video samples with ground-truth contours. Bottom: induced action densities. All strategies use seed 0 and one generated sample per evaluation input.

Controlled real-robot ablation. We compare E2E, E2E+PT, and KASO from the same pretrained checkpoints and the same 300-hour alignment mixture, including 30 hours from G2-90D. Evaluation on G2-90D uses unseen environments and object instances without evaluation-specific fine-tuning: 25 single-object picking trials with a small cube, which is relatively hard to grasp, placed at varied positions, and 10 trials per target in a scene containing a sponge, red apple, mouse, and remote control. There, Follow Score requires contacting the instructed target and no distractor; pick success additionally requires completing the pickup.

Table 6: Controlled real-robot ablation of retaining pretraining losses and action-compatible selection (%). Macro averages weight the four targets equally.

(a) Follow score: four-object scene (10 trials per target)

Method	Sponge	Red apple	Mouse	Remote	Macro avg.
E2E	90	100	90	70	87.5
E2E+PT	100	100	90	90	95
KASO	100	100	90	90	95

(b) Pick success

	Single-object scene (25 trials)	Four-object scene (10 trials per target)				
Method	Small cube	Sponge	Red apple	Mouse	Remote	Macro avg.
E2E	12	30	60	20	0	27.5
E2E+PT	12	40	40	10	0	22.5
KASO	40	60	60	20	10	37.5

KASO and E2E+PT achieve the same 95% four-object Follow Score, but KASO increases macro-average pick success from 22.5% to 37.5%, improving every tested target. Single-object success rises from 12% to 40%. KASO also exceeds E2E on all three aggregate metrics (Table 6), suggesting improved action execution beyond correct target selection.

6 Conclusion and Limitations

GE-Act 2.0 is a world–action pretraining system whose co-training method, KASO, supervises the IDM only on generated futures compatible with recorded actions. Zero-shot OOD evaluation on 100 real-robot tasks across two embodiments shows broad gains with increasing co-training data, while skill-specific coverage is strongly associated with success. Future work includes egocentric-video pretraining at scale and integrating reasoning, long-horizon planning, and memory into this low-level policy.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experiments, assist with method implementation, create or modify scientific figures, summarise existing literature, and draft or edit portions of the manuscript for clarity. We did not use these tools to develop theoretical models, formulate or prove mathematical claims, propose or refine hypotheses, clean or reformat datasets, generate synthetic data, or translate the manuscript. We manually checked LLM-generated code for correctness, verified literature summaries against the cited sources, and reviewed all AI-generated text and figures. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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.

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.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.15391.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025a. arXiv:2410.10733.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. ABot-M0.5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation, 2025b. URL `https://arxiv.org/abs/2506.18088`.

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. IGOR: Image-GOal representations are the atomic control units for foundation models in embodied AI. *arXiv preprint arXiv:2411.00785*, 2024.

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025c. arXiv:2412.04445.

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

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, 2025. URL https://arxiv.org/abs/2510.13626.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.12557.

Generalist AI Team. GEN-0: Embodied foundation models that scale with physical interaction. https://generalistai.com/blog/nov-04-2025-GEN-0, November 2025. Blog post.

Generalist AI Team. GEN-1: Scaling embodied foundation models to mastery. https://generalistai.com/blog/apr-02-2026-GEN-1, April 2026. Blog post.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2512.02012.

Alexi Gladstone, Heng Ji, and Yilun Du. Explorative modeling: Unlocking a third pretraining axis and end-to-end generation. *arXiv preprint arXiv:2607.27372*, 2026.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. LTX-Video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2412.14803.

Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, Yue Liao, Peng Gao, Hongsheng Li, Maoqing Yao, and Guanghui Ren. EnerVerse: Envisioning embodied future space for robotics manipulation. *arXiv preprint arXiv:2501.01895*, 2025.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, and Guanghui Ren. EnerVerse-AC: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations (ICLR)*, 2024a. arXiv:2310.02279.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024b.

Seungku Kim, Suhyeok Jang, Byungjun Yoon, Dongyoung Kim, John Won, and Jinwoo Shin. RoboCurate: Harnessing diversity with action-verified neural trajectory for robot learning. *arXiv preprint arXiv:2602.18742*, 2026.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026. LingBot-VA.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie Envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.02747.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023b. arXiv:2209.03003.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. InstaFlow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.06380.

Yi Liu, Sukai Wang, Dafeng Wei, Xiaowei Cai, Linqing Zhong, Jiange Yang, Guanghui Ren, Jinyu Zhang, Maoqing Yao, Chuankang Li, Xindong He, Liliang Chen, and Jianlan Luo. Unified embodied VLM reasoning with robotic action via autoregressive discretized pre-training. *arXiv preprint arXiv:2512.24125*, 2025.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. URL `https://openaccess.thecvf.com/content_cvpr_2016/html/Mao_Generation_and_Comprehension_CVPR_2016_paper.html`.

Motubrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, Louis Liu, Mengchen Cai, Rongxu Cui, Ruowen Zhao, Runqing Wang, Shuhe Huang, Yao Feng, Yinze Rong, Zeyuan Wang, and Jun Zhu. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025a.

NVIDIA, 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*, 2025b. GR00T N1.7 release: https://github.com/NVIDIA/Isaac-GR00T.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

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

Boxiang Qiu, Liliang Chen, Yue Liao, Nan Wang, Lintao Wang, Jiayi Luo, Wenzhi Zhao, Shengcong Chen, Di Chen, Ye Li, Chen Gao, Shuicheng Yan, Si Liu, Maoqing Yao, and Guanghui Ren. GE-Sim 2.0: A roadmap towards comprehensive closed-loop video world simulators for robotic manipulation. *arXiv preprint arXiv:2605.27491*, 2026.

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

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2202.00512.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, Chen Gao, Wei Wu, Xihui Liu, Dhruv Shah, Zhaoxiang Zhang, Zhibo Chen, Jun Zhu, Yonghong Tian, Tat-Seng Chua, Wenwu Zhu, and Yong Li. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023. arXiv:2303.01469.

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

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

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

Siyin Wang, Junhao Shi, Zhaoyang Fu, Xinzhe He, Feihong Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, Mike Zheng Shou, Xuanjing Huang, Xipeng Qiu, and Yu-Gang Jiang. World action models: The next frontier in embodied AI. *arXiv preprint arXiv:2605.12090*, 2026a.

Yuran Wang, Siqiao Huang, Mingleyang Li, et al. OpenWAM: An open, modular exploration towards systematic world-action model pretraining, 2026b. URL https://arxiv.org/abs/2609.07398.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems (RSS)*, 2024. arXiv:2401.00025.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024a.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2024b. arXiv:2312.13139.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. DayDreamer: World models for physical robot learning. In *Conference on Robot Learning*, 2022.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From foundation to application: Improving VLA models in practice. *arXiv preprint arXiv:2607.06403*, 2026. LingBot-VLA 2.0 technical report.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2408.12528.

Jingfeng Yao, Yuda Song, Yucong Zhou, and Xinggang Wang. Towards scalable pre-training of visual tokenizers for generation. *arXiv preprint arXiv:2512.13687*, 2025a.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b. arXiv:2501.01423.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. GigaWorld-Policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.11758.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b.

Chenghao Yin, Da Huang, Di Yang, Jichao Wang, Nanshu Zhao, Chen Xu, Wenjun Sun, Linjie Hou, Zhijun Li, Junhui Wu, et al. Genie Sim 3.0: A high-fidelity comprehensive simulation platform for humanoid robot, 2026. URL https://arxiv.org/abs/2601.02078.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b. arXiv:2311.18828.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.06940.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, Jie Zhang, Jingyang Fan, Gengze Zhou, Qihang Peng, Chenxu Lv, Xiaoyue Chen, An Yang, Fei Huang, Junyang Lin, Dayiheng Liu, Jingren Zhou, Chenfei Wu, and Xiong-Hui Chen. Qwen-RobotManip technical report: Alignment unlocks scale for robotic manipulation foundation models, 2026a. URL https://arxiv.org/abs/2606.17846.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination?, 2026b. URL https://arxiv.org/abs/2603.16666.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. arXiv:2303.15343.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, Jiaming Zhou, Yishu Shen, Yudong Jin, Fangyi Xu, Shuailei Ma, Jiaqi Liao, Guanxing Lu, Zifan Shi, Yongkun Wen, Yujie Zhao, Weixuan Tang, Xinyang Wang, Chaojian Li, Jiapeng Zhu, Ka Leong Cheng, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026. LingBot-VA 2.0.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Maoqing Yao, Si Liu, and Guanghui Ren. ACoT-VLA: Action chain-of-thought for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2601.11404.

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*, 2025.

Pengfei Zhou, Shengcong Chen, Di Chen, et al. τ_0-WM: A unified video-action world model for robotic manipulation. *arXiv preprint arXiv:2606.01027*, 2026.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2406.14540.

A Related Work

A.1 Generalist Policies and World–Action Models

Generalist robot policies scale language-conditioned control across broad task collections (Brohan et al., 2022; Black et al., 2024; Physical Intelligence et al., 2025; Octo Model Team et al., 2024; Kim et al., 2024b; Liu et al., 2025; Generalist AI Team, 2025; 2026). World–action models connect visual prediction to control through several interfaces (Wang et al., 2026a). Decoupled two-stage designs first generate future visual states and then infer actions with a separate action model (Jang et al., 2025); joint designs predict video and actions in a shared backbone (Wu et al., 2024b; Cheang et al., 2024; Motubrain Team et al., 2026; Ye et al., 2026b); and feature-coupled designs decode actions from internal representations of a generative backbone (Liao et al., 2025; Hu et al., 2025). GE-Act 2.0 uses an explicit completed future as the interface to a separately pretrained IDM. This differs from the parallel action branch of Genie Envisioner (Liao et al., 2025) and allows the generator and IDM to pretrain on distinct data before end-to-end co-training.

A.2 Representations for Robot Video and Control

Most latent video generators operate in autoencoder spaces developed for content reconstruction (Wan Team et al., 2025; NVIDIA et al., 2025a). Robot learning has also explored representations designed around motion or action: Genie discovers discrete latent actions from video (Bruce et al., 2024); LAPA and Moto connect learned motion tokens to robot actions (Ye et al., 2025; Chen et al., 2025c); and IGOR compresses visual change into a shared latent action space (Chen et al., 2024). Other work uses predictive visual features directly for control, including video-pretrained backbones (Wu et al., 2024b; Cheang et al., 2024), diffusion features (Hu et al., 2025), and point trajectories (Wen et al., 2024). In image generation, representation alignment methods such as REPA and VA-VAE show that matching a tokenizer's latents or a generator's intermediate features to frozen semantic encoders can improve generation (Yu et al., 2025; Yao et al., 2025b). CoAE addresses both concerns: it provides a compact latent space with a reconstruction decoder for generation, is aligned to multiple pretrained visual teachers, and is evaluated by how well a fixed-capacity action probe can recover actions from its encoded video windows.

A.3 Fast Video Generation for World Models

Robot world models often adapt a video generator pretrained on natural video, then retain multi-step diffusion or compress the sampler through distillation (Jang et al., 2025; Guo et al., 2025; Huang et al., 2025; Jiang et al., 2025; Qiu et al., 2026; Zhu et al., 2025; Li et al., 2026; Zhang et al., 2026; Chen et al., 2026; NVIDIA et al., 2025a). One- and few-step generation has a broader lineage in images: progressive and distribution-matching distillation compress a pretrained teacher (Salimans & Ho, 2022; Yin et al., 2024b;a); consistency models enforce agreement along the probability-flow trajectory (Song et al., 2023; Luo et al., 2023; Kim et al., 2024a); and rectified or shortcut flows learn paths intended for small integration budgets (Lipman et al., 2023; Liu et al., 2023b; 2024; Frans et al., 2025). MeanFlow and improved MeanFlow instead train an average-velocity field that can traverse the complete noise interval in one network forward pass without first distilling a multi-step teacher (Geng et al., 2025; 2026). Seaweed-APT demonstrates one-step natural-video generation through adversarial post-training (Lin et al., 2025). The closest methodological starting points for our single-step visual planner are MeanFlow and improved MeanFlow; our focus is their conditional extension to multi-view future visual states and their use as a differentiable train–deployment interface for an action model.

A.4 Connecting Generated Futures to Actions

World models used only for planning, latent imagination, or representation extraction need not expose an action model to generated future visual-state samples (Hafner et al., 2018; 2019; Wu et al., 2022; Hu et al., 2025; Qiu et al., 2026). World–action training creates a stronger coupling. Teacher forcing trains the action model on recorded future visual states, leaving a generated-versus-recorded input gap at deployment, as in LingBot-VA (Li et al., 2026; Zhang et al., 2026); other systems feed generated future visual states but detach the sampling path or rely on robustness to absorb generation

errors (Chen et al., 2026). These approaches primarily address whether generated video is used and whether action gradients reach the video generator. KASO targets a different mismatch: even a plausible visual prediction may represent a different outcome mode from the recorded action used as its label. It samples several candidates, ranks them with the active IDM against the recorded future visual states, and applies generated-visual-state action supervision only to the compatible candidates (Section 2.3). We also note that a similar best-of-K selection rule has appeared independently in concurrent work on general generative modeling (Gladstone et al., 2026), where training on the best of K candidate matches lets predictions commit to modes rather than blur them; KASO instead judges compatibility in action space with the active IDM and uses the selection to align the visual planner with the IDM.

A.5 Evaluating Pretrained Robot Models

Robot models are commonly evaluated after task-specific fine-tuning (Liao et al., 2025; Li et al., 2026; Zhang et al., 2026; Ye et al., 2026a; Zhou et al., 2026; Chen et al., 2026; Wu et al., 2024b; Cheang et al., 2024), while fewer systems report zero-shot results (Zhou et al., 2026; Physical Intelligence et al., 2025; Generalist AI Team, 2025; 2026). Fine-tuned performance is important, but it combines the quality of the pretrained model with the amount and coverage of adaptation data. We therefore report no-fine-tuning evaluation as the primary measure of pretrained capability and state which visual factors are excluded and which forms of task adaptation are absent (Section 3). Simulation and world-model benchmarks (Liu et al., 2023a; James et al., 2020; Shang et al., 2026) provide horizontal reference points rather than direct evidence for the capability acquired by our manipulation-video pretraining.

B Detailed Representation and Co-Training Design

B.1 Compact Visual Representation for World–Action Pretraining

The first system requirement is a visual space compact enough for repeated future prediction while retaining the state changes needed for action recovery. Most WAMs inherit a latent space from a pretrained video generator (Jang et al., 2025; Guo et al., 2025; Li et al., 2026; Zhang et al., 2026; Chen et al., 2026). These spaces typically use modest spatial compression, leaving a large token burden inside the control loop, and are optimized for generic reconstruction rather than manipulation-specific visual planning.

GE-Act 2.0 uses a control-oriented autoencoder (CoAE) as its shared visual interface. CoAE combines aggressive spatial compression with pixel reconstruction and alignment to complementary frozen visual teachers; the same latents are consumed by the SVP and IDM. Its role is therefore practical and measurable: reduce the cost of future prediction while exposing information useful for control and instruction grounding. We describe the training objective and then test these two properties with frozen-representation probes.

B.1.1 Representation Design and Training

CoAE is a framewise 2D autoencoder designed to produce a short spatial token sequence. Given an input frame o, its encoder produces $z = \mathcal{E}(o)$. Whereas commonly used video autoencoders downsample each spatial axis by $8\times$ or $16\times$ (Wan Team et al., 2025; NVIDIA et al., 2025a), CoAE uses a $64\times$ spatial downsampling factor. At the 256×384 frame resolution used by the SVP (Section 2.2), this maps each input frame to a 4×6 grid of 512-channel latents, corresponding to 24 tokens per frame. We initialize CoAE from the 128-channel DC-AE (Chen et al., 2025a) by transferring its compatible weights, while expanding the latent width to 512 channels and randomly initializing the newly introduced parameters.

During autoencoder training, a decoder reconstructs the input as $\hat{o} = \mathcal{D}(z)$. We supervise this reconstruction with pixel, perceptual (LPIPS), and adversarial losses:

$$\mathcal{L}_{\mathrm{rec}} = \left\|\hat{o} - o\right\|_2^2 + \lambda_{\mathrm{perc}}\,\mathcal{L}_{\mathrm{perc}}\big(\hat{o}, o\big) + \lambda_{\mathrm{gan}}\,\mathcal{L}_{\mathrm{gan}}\big(\hat{o}\big). \tag{3}$$

Reconstruction alone does not directly supervise the semantic and spatiotemporal information relevant to action control. Motivated by evidence that representation objectives can improve generative models

trained over learned visual tokens (Yao et al., 2025b;a), we align z with three frozen visual teachers. Separate alignment heads g_k map z toward the final-layer representations $G_k(o)$ of the SigLIP 2 vision encoder used by Qwen3.5, V-JEPA 2.1, and DINOv3 (Zhai et al., 2023; Tschannen et al., 2025; Mur-Labadia et al., 2026; Siméoni et al., 2025). These teachers provide complementary language-aligned semantics, spatiotemporal structure, and dense visual features, respectively. The alignment objective is

$$\mathcal{L}_{\text{align}} = \sum_{k=1}^{3} \lambda_k \left(1 - \cos\big(g_k(z), G_k(o)\big)\right). \tag{4}$$

The complete training objective is $\mathcal{L}_{\text{CoAE}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{align}}$. Figure 2 summarizes the architecture and training objectives.

We evaluate this representation independently in Section 5.1 before interpreting the complete system's scaling results.

B.2 Action-Compatible World–Action Co-Training

The second system requirement appears when the separately pretrained SVP and IDM are connected. The IDM must learn from generated futures, but a stochastic future need not depict the same valid behavior as the recorded action paired with its context. This creates a video–action mismatch that we call the validity gap (Section 2.3). GE-Act 2.0 uses Knowledge-Aligned Selective Optimization (KASO) as its co-training rule: candidate futures are compared in the IDM's action space, and only action-compatible candidates receive the generated-future action loss.

B.2.1 Inverse Dynamics and Separate Pretraining

The IDM translates visual transitions into actions. We denote the CoAE latents of the current multi-view observation by o, the future visual latents by z, and the current proprioceptive state by s. The IDM attends to the future visual latents through cross-attention; the future visual-state sequence spans the dense and sparse portions of the temporal schedule in Section 2.2. Conditioned on the future visual-state sequence, the IDM jointly produces a dense control-rate action chunk and sparse actions over the longer horizon. At deployment, the controller executes only the dense action chunk. The sparse actions are auxiliary training targets that encourage the IDM to preserve longer-horizon intent. With $a_t = t\,\varepsilon + (1-t)\,a$ denoting a noised action sequence at flow time t, its velocity head v_ϕ is trained by

$$\ell_{\text{FM}}\big(a;\, o,\, s,\, z\big) \;=\; \mathbb{E}_{t,\varepsilon}\Big\| v_\phi\big(a_t, t;\, o, s, z\big) - \big(\varepsilon - a\big)\Big\|^2. \tag{5}$$

During separate IDM pretraining, a and the recorded future visual latents z come from the same physical rollout, so they are compatible by construction. During co-training, a remains the action stored in that rollout, whereas the IDM receives generated future visual latents $\hat{z}$ from the SVP.

B.2.2 Connecting the Pretrained Components

GE-Act 2.0 is pretrained in three steps. CoAE is trained first and then frozen, fixing the shared visual latent space. In that space, the SVP is pretrained on instruction–video data and the IDM is pretrained separately on paired recorded future–action data, so neither component requires the other's annotations. KASO co-training then connects the two pretrained checkpoints through generated futures while retaining their original objectives.

Using these pretrained components, our one-step generator allows the SVP and the IDM to be co-trained directly through a differentiable generator–IDM interface. Let G_θ denote the one-step generation map in Eq. (1). The generated future and End-to-End Co-training (E2E) objective are

$$\hat{z} = G_\theta(o, c, \xi), \qquad \mathcal{L}_{\text{E2E}} = \ell_{\text{FM}}(a; o, s, \hat{z}). \tag{6}$$

The action gradient passes through the generated video to update both components. However, optimizing only $\mathcal{L}_{\text{E2E}}$ can overwrite capabilities acquired during separate pretraining. A similar effect has been reported in VLA training, where optimizing only the action objective can degrade pretrained VLM capabilities, motivating the continued use of VLM pretraining objectives during robot training (Driess et al., 2025). We therefore retain the SVP's video objective $\mathcal{L}_{\text{SVP}} = \mathcal{L}_u + \mathcal{L}_v$

and the IDM's recorded-video objective $\mathcal{L}_{\text{IDM}} = \ell_{\text{FM}}(a; o, s, z)$ during co-training. We refer to this variant as E2E with pretraining losses (E2E+PT). Suppressing fixed loss coefficients,

$$\mathcal{L}_{\text{E2E+PT}} = \mathcal{L}_{\text{E2E}} + \mathcal{L}_{\text{SVP}} + \mathcal{L}_{\text{IDM}}. \tag{7}$$

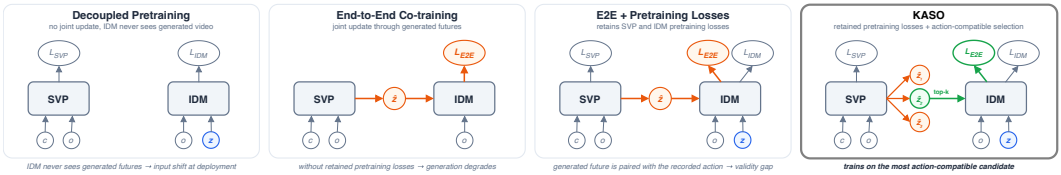

Figure 7: Comparison of Decoupled Pretraining, End-to-End Co-training, E2E + Pretraining Losses, and KASO. Decoupled Pretraining is the pretraining stage itself: the SVP and the IDM are trained on their own objectives and never exchange generated futures. The four strategies differ in whether the IDM receives generated future visual states, whether the pretraining losses for the SVP and IDM remain active, and whether generated candidates are filtered for action compatibility.

The first three panels of Figure 7 compare Decoupled Pretraining, E2E, and E2E+PT. E2E+PT retains both pretraining losses, but it does not eliminate conflicting supervision during co-training.

B.2.3 Why Generated Futures Can Conflict with Recorded Actions

The remaining conflict arises within the generated co-training pair itself: the IDM conditions on the generated $\hat{z}$, while its target remains the recorded action a. Although a and the recorded z come from the same rollout, $\hat{z}$ is sampled separately and need not depict a future consistent with a.

Manipulation is inherently multimodal: given the same observation o and instruction c, the task can still be completed in several different ways. The robot may approach from the left or the right, grasp now or a moment later, or choose among multiple objects that satisfy the instruction. We refer to each behaviorally distinct realization as an *outcome mode*. Under behavior cloning, each demonstration provides action supervision only for the outcome mode realized by its recorded episode, whereas stochastic generation may sample any outcome mode consistent with the same observation and instruction. Consequently, (a, z) share an outcome mode by construction, while an independently generated $\hat{z}$ and the recorded a need not.

Formally, the dataset provides a joint sample, whereas the generated training pair combines two conditional marginals,

$$\begin{aligned} &(z, a) \sim p_{\text{data}}(z, a \mid o, s, c), && && \text{recorded in the same rollout,} \\ &\hat{z} \sim p_\theta(z \mid o, c), \quad a \sim p_{\text{data}}(a \mid o, s, c), && \hat{z} \perp a \mid (o, s, c), && \text{paired during co-training.} \end{aligned} \tag{8}$$

A generated pair is action-compatible when $\hat{z}$ and a instantiate the same outcome mode; otherwise the IDM is asked to predict one mode's action while conditioning on another mode's future visual states. We call this video–action mismatch **the validity gap**. The conditional independence in Eq. (8) distinguishes the gap from generation error: even if $p_\theta(z \mid o, c)$ matches the true future marginal, better generation makes each sampled $\hat{z}$ more plausible but does not couple its outcome mode to the recorded action.

The controlled toy analysis of this mismatch is reported with the component studies in Section 5.2.

B.2.4 KASO: Knowledge-Aligned Selective Optimization

KASO implements an online top-k procedure for selecting action-compatible future visual-state sequences during co-training, as illustrated in Figure 7. At a high level, for each context, the SVP samples N candidate future videos. The active IDM translates each candidate into an action response and compares it with its response to the recorded video. KASO selects the top-k candidates with the smallest discrepancy; only those candidates are replayed with gradients and receive the generated-video action loss (Figure 3). The comparison is deliberately made in action space rather than pixel

space: two visually different future visual-state sequences may support the same action, for example when they differ only in task-irrelevant background motion, whereas a visually convincing prediction from another mode does not. In the toy system of Figure 6, compatibility-based selection suppresses mismatched cross-mode pairs, allowing KASO to retain all four video modes and recover the two action modes near ± 1.

Relative to E2E+PT in Eq. (7), KASO changes only the generated-video action branch by applying it to the selected top-k candidates. Comparing E2E+PT with KASO therefore isolates selection, while comparing E2E with E2E+PT isolates the two retained pretraining losses.

To score the candidates, KASO probes the IDM at a fixed high-noise point t_p on its action-flow path. It draws one probe noise ε and constructs $\tilde{a} = t_p\varepsilon + (1 - t_p)a$; the same $\tilde{a}$ is then evaluated under the recorded future visual states and under every generated candidate. Sharing the action input and the active IDM parameters leaves the future visual conditioning as the only changing input. For the set $\mathcal{D}$ of dense action tokens, the candidate energy is

$$E_n = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \left\| v_{\phi,i}\big(\tilde{a}, t_p;\, o, s, \hat{z}_n\big) - v_{\phi,i}\big(\tilde{a}, t_p;\, o, s, z\big) \right\|_2^2. \tag{9}$$

We average only over dense tokens, which directly cover the action-execution horizon depicted by the near-horizon frames. Probing at high noise is what makes this energy informative. At low noise, $\tilde{a} \approx a$ and the flow target $\varepsilon - a$ is largely predictable from the action input alone, so the velocity depends only weakly on the future visual conditioning and all candidates receive similar energies. At high noise, $\tilde{a}$ carries almost no information about a, so the velocity must be inferred from $(o, s, \hat{z}_n)$, and Eq. (9) reduces to the disagreement between the action the IDM reads from the candidate future and the action it reads from the recorded future. A low energy therefore means that the current IDM cannot behaviorally distinguish the candidate from the recorded future, which is the action-compatibility notion introduced above.

Let $\mathcal{S}_k$ contain the indices of the k candidates with the lowest energies. The selected E2E loss is

$$\mathcal{L}_{\mathrm{E2E}}^{\mathrm{sel}} = \frac{1}{k} \sum_{n \in \mathcal{S}_k} \ell_{\mathrm{FM}}(a; o, s, \hat{z}_n). \tag{10}$$

The full KASO objective retains the same two pretraining losses:

$$\mathcal{L}_{\mathrm{KASO}} = \mathcal{L}_{\mathrm{E2E}}^{\mathrm{sel}} + \mathcal{L}_{\mathrm{SVP}} + \mathcal{L}_{\mathrm{IDM}}. \tag{11}$$

We use $k = 1$ in our implementation.

Candidate generation and scoring are performed without gradients, while the generation noise for every candidate is retained. After selection, the SVP replays the selected noises with gradients enabled; because generation is a deterministic one-step map of each noise, every replay reconstructs exactly the visual prediction that was scored. The IDM then applies its standard flow-matching objective with a freshly sampled action noise and flow time, and the gradient passes through both the IDM and the SVP via Eq. (1). The update also includes $\mathcal{L}_{\mathrm{SVP}}$ on the recorded video and $\mathcal{L}_{\mathrm{IDM}}$ on recorded action–video pairs. Thus, KASO adapts the IDM to generated inputs, shapes the SVP's flow generator with the action objective, and continues both pretrained objectives in the same stage.

Selection is recomputed online at every optimizer step using the current SVP and active IDM, so the selected examples track both models as they change. It also relies on a stochastic video generator: without diversity across noise draws, the candidate pool would contain no alternative mode to select. This online resampling distinguishes KASO from offline data-curation pipelines that generate and filter a fixed synthetic dataset before policy training (Kim et al., 2026). Group size N, probe level t_p, loss coefficients, and the recorded/generated mixing ratio are given in Appendix E.

Full pseudocode is provided in Algorithm 1 in Appendix C.4.

The controlled toy study and matched real-robot ablation are reported in Section 5.2.

B.3 CoAE Probe Implementation Details

Action recovery. We freeze each encoder and train the same two-layer inverse-dynamics probe on 400 samples from a cluttered shelf organization task, pairing dense and sparse frames with recorded actions. We evaluate action MAE on 100 held-out samples using matched splits and optimization across representations.

Instruction grounding. The caption-matching probe uses 5,000 GenieSim-Instruction episodes distributed uniformly across the ten benchmark tasks. The complete task list appears in Table 3. A positive pair contains an observation and its original instruction, whereas a negative pair uses an instruction sampled from another episode; the two classes are sampled equally. For each frozen encoder, an eight-layer network following our video-generator architecture (Section 2.2) fuses the latent tokens and candidate instruction while retaining VLM conditioning. A two-layer MLP classifies the average-pooled hidden states. We optimize the probe with binary cross-entropy under the same protocol for all representations, and report caption-match error, the complement of classification accuracy.

C Visual Planner Details and Co-Training Algorithm

C.1 Multi-Scale Visual Prediction

A useful visual prediction must explain both how the scene changes next and where the task is headed. Dense near-term targets train the SVP to resolve fine motion and interaction dynamics, while sparse far-horizon targets require it to preserve task intent and model longer-term scene evolution. Training at both scales encourages the SVP to connect near-term interaction dynamics with longer-term scene evolution and task intent. The same two scales give the IDM fine temporal resolution over the imminent action chunk and sufficient reach to determine task-level intent. Predicting every frame at control rate over the full horizon, however, would spend most of the token budget on the far horizon, where fine resolution is unnecessary. We therefore adopt the multi-scale temporal design of Act2Goal (Zhou et al., 2025). From the current frame f_0, the SVP predicts N_d *dense* frames uniformly strided across the action-execution horizon H, and N_s *sparse* frames partitioning the remainder of the clip up to its end frame $f_0 + T$,

$$\mathcal{F} \;=\; \{\, f_0 \,\} \;\cup\; \underbrace{\left\{\, f_0 + j\,\tfrac{H}{N_d} \,\right\}_{j=1}^{N_d}}_{\text{dense: control}} \;\cup\; \underbrace{\left\{\, f_0 + H + \left\lfloor k\,\tfrac{T-H}{N_s} \right\rceil \right\}_{k=1}^{N_s}}_{\text{sparse: intent}}. \tag{12}$$

The sparse partition is arithmetic and includes the clip end $f_0 + T$ at $k = N_s$, with the reach T set per clip. Default values of (N_d, N_s, H) are given in Appendix E.

C.2 Toward Unified Understanding and Generation

Multimodal modeling is increasingly moving toward systems that connect understanding and generation instead of treating them as isolated capabilities (Xie et al., 2025; Wu et al., 2024a). This connection is particularly important for a visual planner: an instruction such as "place it behind the red bowl" cannot be resolved from language alone, because its referents and spatial relations depend on the current scene. A common video-generation design, exemplified by Wan (Wan Team et al., 2025), represents language with a text-only encoder and leaves this scene-specific grounding to the generative backbone.

To provide scene-grounded conditioning, the SVP uses a frozen vision–language model (VLM) as its multimodal understanding component. The VLM jointly processes the current head-view observation o_{head} and instruction c, producing hidden states at each layer,

$$\big(h_0, h_1, \ldots, h_L\big) = \mathrm{VLM}\big(o_{\text{head}}, c\big), \qquad h_\ell \in \mathbb{R}^{S\times d}. \tag{13}$$

We instantiate this component with Qwen3.5 (Qwen Team, 2026). Only the text-span states $h_\ell[\mathcal{T}]$ are passed to the flow generator, but they have been contextualized by the accompanying image. This keeps the cross-attention sequence text-sized while grounding its content in the observed scene.

Different VLM layers retain different mixtures of lexical, spatial, and visual information. We therefore fuse their text-span states with an input-dependent gate. Writing g for the learned gating network and Pool for masked token pooling,

$$\alpha_\ell = \mathrm{softmax}_\ell\big(g\big(\mathrm{Pool}\big(h_\ell[\mathcal{T}]\big)\big)\big), \qquad c_{\text{vlm}} = \mathrm{LN}\left(\sum_{\ell=0}^{L} \alpha_\ell h_\ell[\mathcal{T}]\right). \tag{14}$$

Every DiT block cross-attends to c_{vlm}, shared across generated views. The two modules retain specialized backbones, but form a single visual-planning pathway: the VLM resolves what the instruction refers to, and the generative DiT predicts how that grounded scene should evolve. The IDM receives neither c nor c_{vlm} directly, so language affects action through the generated future visual states and does not become a prerequisite for IDM pretraining.

C.3 Conditional MeanFlow for One-Step Visual Planning

The SVP formulates temporally structured, conditional multi-view visual prediction as a one-step MeanFlow problem (Geng et al., 2025; 2026). The resulting objective maps Gaussian noise to a complete future visual-latent sequence in one differentiable forward pass. For recorded future visual latents z and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, the linear trajectory is $z_t = (1-t)z + t\epsilon$, with velocity $v_c = \epsilon - z$, and its average velocity over an interval $[r, t]$ is

$$u(z_t, r, t) := \frac{1}{t-r} \int_r^t v(z_\tau, \tau)\, \mathrm{d}\tau, \tag{15}$$

which directly relates two points on the same trajectory through $z_r = z_t - (t-r)u(z_t, r, t)$. With $(r, t) = (0, 1)$, one forward evaluation of the learned field maps noise to data. Differentiating Eq. (15) with respect to t gives

$$u(z_t, r, t) = v(z_t, t) - (t-r)\frac{\mathrm{d}}{\mathrm{d}t} u(z_t, r, t), \tag{16}$$

where the total derivative is

$$\frac{\mathrm{d}}{\mathrm{d}t} u_\theta = \partial_z u_\theta \cdot \frac{\mathrm{d}z_t}{\mathrm{d}t} + \partial_t u_\theta. \tag{17}$$

We optimize this identity with a shared DiT trunk, a mean-velocity head u_θ, and an auxiliary instantaneous-velocity head v_θ. The auxiliary prediction supplies the state tangent in the total derivative, and a single Jacobian–vector product (JVP) yields the compound prediction

$$V_\theta = u_\theta + (t-r)\ \mathrm{sg}\left[\frac{\mathrm{d}u_\theta}{\mathrm{d}t}\right], \tag{18}$$

where $\mathrm{sg}[\cdot]$ stops gradients through the JVP correction. The prediction V_θ is regressed against v_c, while v_θ receives an auxiliary flow-matching loss.

To construct the conditional multi-view trajectory, we keep observed frames fixed and evolve only future frames. Each training frame is encoded by CoAE into a latent $z^{(i)}$ and assigned a binary mask

$$m_i = \begin{cases} 0, & \text{conditioning frame,} \\ 1, & \text{future frame to predict.} \end{cases} \tag{19}$$

Suppressing the view axis, the conditional path is

$$z_t^{(i)} = (1-m_i)z^{(i)} + m_i\big((1-t)z^{(i)} + t\epsilon^{(i)}\big), \qquad v_c^{(i)} = m_i\big(\epsilon^{(i)} - z^{(i)}\big). \tag{20}$$

Thus, conditioning coordinates remain clean and have zero velocity throughout the path.

For the derivative to remain consistent with this partial trajectory, we mask the predicted state tangent and the per-token interval endpoints,

$$\frac{\mathrm{d}z_t}{\mathrm{d}t} = m \odot v_\theta(z_t, t), \qquad (r_i, t_i) = (m_i r, m_i t), \tag{21}$$

so conditioning tokens contribute neither an advection tangent nor a time derivative. The masked derivative is computed in one JVP computation. Both heads are supervised only on prediction coordinates,

$$\mathcal{L}_u = \big\| m \odot (V_\theta - v_c) \big\|^2, \qquad \mathcal{L}_v = \big\| m \odot (v_\theta - v_c) \big\|^2. \tag{22}$$

Sampling distributions for (r, t), dense/sparse weighting, and stabilization details are given in Appendix E.

The resulting one-step prediction map is given in Eq. (1) in the main paper.

C.4 KASO Co-Training Pseudocode

Algorithm 1 One co-training step with KASO. Setting $k = N$ recovers E2E+PT; removing in addition $\mathcal{L}_{\text{SVP}}$ and $\mathcal{L}_{\text{IDM}}$ recovers E2E.

Require: SVP flow-generator parameters θ, IDM parameters ϕ
Require: Current observation latents o, proprioceptive state s, instruction c, recorded future visual latents z, recorded action a
Require: Group size N, selected count k, probe noise level t_p

// Pretraining losses on recorded data
Sample video flow times and noise; compute $\mathcal{L}_{\text{SVP}} = \mathcal{L}_u + \mathcal{L}_v$ *Eq.* (22)
$\mathcal{L}_{\text{IDM}} \leftarrow \ell_{\text{FM}}(a;\, o, s, z)$ *Eq.* (5)

// Candidate generation without gradients
for $n = 1, \ldots, N$ **do**
  Draw and retain generation noise ξ_n
  $\hat{z}_n \leftarrow$ one-step generation from (o, c, ξ_n) *Eq.* (1)
end for

// Compatibility probe with the active IDM, no gradients
Draw shared probe noise ε; $\tilde{a} \leftarrow t_p\varepsilon + (1 - t_p)a$
$v_{\text{ref}} \leftarrow v_\phi(\tilde{a}, t_p;\, o, s, z)$
for $n = 1, \ldots, N$ **do**
  $E_n \leftarrow \|v_\phi(\tilde{a}, t_p;\, o, s, \hat{z}_n) - v_{\text{ref}}\|^2_{\text{dense}}$ *Eq.* (9)
end for

// Top-k selection and differentiable replay
$\mathcal{S}_k \leftarrow$ indices of the k smallest energies in $\{E_n\}_{n=1}^N$
for $n \in \mathcal{S}_k$ **do**
  $\hat{z}_n \leftarrow$ replay ξ_n with gradients
end for

// Joint update
$\mathcal{L}^{\text{sel}}_{\text{E2E}} \leftarrow \frac{1}{k}\sum_{n\in\mathcal{S}_k} \ell_{\text{FM}}(a;\, o, s, \hat{z}_n)$ *Eq.* (10)
$\mathcal{L}_{\text{KASO}} \leftarrow \lambda_{\text{E2E}}\mathcal{L}^{\text{sel}}_{\text{E2E}} + \lambda_{\text{SVP}}\mathcal{L}_{\text{SVP}} + \lambda_{\text{IDM}}\mathcal{L}_{\text{IDM}}$ *Eq.* (2)
$\theta, \phi \leftarrow \theta, \phi - \alpha\nabla_{\theta,\phi}\mathcal{L}_{\text{KASO}}$

C.5 Analysis of the Controlled Toy System

To isolate the consequences of the validity gap, we construct a toy two-stage system (details in Appendix E.7) and train it with each strategy in Figure 7, starting from identical initializations. Future visual states are drawn from an equal-weight mixture of four Gaussian outcome modes centered at $(\pm 1.5, \pm 1.8, 0)$, with every recorded sample lying exactly on the $z_3 = 0$ plane. Actions follow the deterministic mapping $a = f(z) = z_1 z_2/(1.5 \times 1.8)$, which maps the four video modes to two action modes near $a = \pm 1$. Consequently, an IDM trained only on recorded pairs is unconstrained away from the ground-truth plane when a generator produces residual z_3 deviations. To expose this off-manifold failure mode, every arm uses the same IDM input normalization $(1.5, 1.8, 0.2)$; the third-coordinate scaling leaves all recorded inputs unchanged but magnifies generated off-plane deviations equally across methods. Full controls and training details are given in Appendix E.7. We use a flow-matching generator as the first stage of this system, corresponding to the SVP in the full WAM, and an action regressor as the second stage, serving as a simplified IDM.

Figure 6 shows the learned video distribution and its induced action density for seed 0. Ground truth contains four video modes and two action modes near ± 1. For this displayed run, Decoupled Pretraining covers all four video modes, but its action density is diffuse: its IDM is trained only on recorded states and is therefore unconstrained on generated states outside that support. Because generated futures and recorded actions are paired independently in Eq. (8), both E2E and E2E+PT

associate each generated outcome mode with equally weighted action targets near ± 1. Under squared-error regression, the IDM regressor in both methods minimizes this conflicting supervision by predicting their mean at zero. Without the retained pretraining losses, E2E also loses the four video modes retained by E2E+PT. A generative IDM may preserve both action modes, but the same pairing still removes their dependence on the outcome mode shown by the generated video. Our proposed method, KASO (Section 2.3), couples the SVP and the IDM through action-compatible pairs and recovers both the multimodal video distribution and the bimodal action distribution, so subsequent post-training begins with a more diverse action distribution, which is particularly important for methods such as reinforcement learning. The toy KASO arm selects one of eight candidates during training only; all arms use one generated sample per evaluation input. Moreover, at seed 0 the Decoupled arm has the lowest IDM error on recorded pairs and the lowest video sliced-Wasserstein distance among the learned arms, localizing its failure to the generated-video action interface rather than either pretrained module.

D Training Data and Mixture Composition

GE-Act 2.0 uses three stage-specific data mixtures. SVP pretraining uses instruction–video data and can incorporate manipulation recordings without robot action labels, including egocentric video. IDM pretraining uses action-labeled robot trajectories, including demonstrations, failure trajectories, and deployment data. Co-training with KASO connects the two components through instruction–video–action data while retaining their separately pretrained objectives (Section 2.3).

The data span G1-OP, G2-OP, G2-90D, simulation, open-source robot datasets, egocentric and human manipulation video, and rollout and failure data. Their proportions differ across SVP pretraining, IDM pretraining, and co-training, as summarized in Figure 8. G1-OP contributes more than 50% of the co-training mixture, whereas G2-90D contributes less than 2%. This imbalance motivates reporting atomic-skill results separately for the two embodiments in Section 3.1. The three stages contain 39,000, 32,000, and 30,000 hours, respectively.

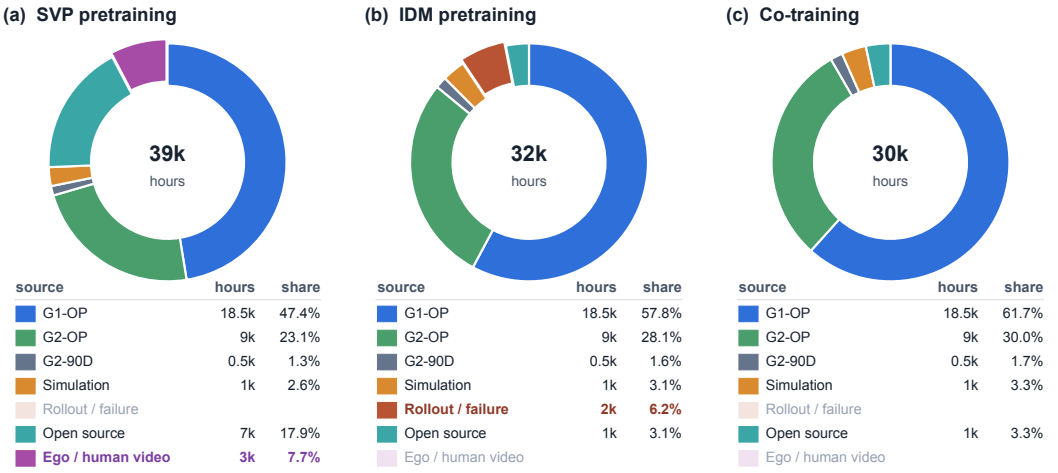

Figure 8: Stage-specific training-data mixtures. The panels show the 39,000-hour SVP pretraining mixture, the 32,000-hour IDM pretraining mixture, and the 30,000-hour co-training mixture. Slice angle denotes the share of data hours; colors and source order are held fixed across panels. Violet and coral highlights emphasize the action-free egocentric/human video used for SVP pretraining and the rollout/failure trajectories used for IDM pretraining, respectively. The keys report source hours and percentages.

E Training Recipes and Hyperparameters

Unless stated otherwise, all stages use bf16 mixed precision with DeepSpeed ZeRO stage 2, AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$ and weight decay 10^{-5}, gradient-norm clipping at 1.0, a constant learning

rate after a linear warmup, and no classifier-free guidance or caption dropout. Batch sizes are per GPU.

E.1 ARCHITECTURES

Table 7 summarizes the configuration of each trained component.

Table 7: Trained components. Parameter counts are read from the final co-trained checkpoint; the frozen Qwen3.5-2B VLM (2.72 B parameters) is excluded.

Component	Configuration
CoAE	Framewise deep-compression autoencoder (DC-AE layout) with $64\times$ spatial downsampling and 512 latent channels; a 256×384 frame gives a 4×6 grid of 24 tokens. Latents are standardized per channel with corpus statistics.
SVP flow generator	2.51 B parameters. A 36-block diffusion transformer of width 2048 (32 heads): 20 shared blocks followed by 8 blocks for the mean-velocity head u_θ and 8 parallel blocks for the instantaneous-velocity head v_θ. Every block cross-attends to the fused VLM states; every third block attends across views. Conditioning enters through adaptive layer normalization as the sum of sinusoidal embeddings of t and of r and a learned view-identity embedding. Per view the input is one conditioning frame plus six predicted frames, i.e., $7 \times 24 = 168$ tokens.
VLM (frozen)	Qwen3.5-2B. The prompt holds the current head-view frame (96 vision tokens) and the instruction in a fixed window of 192 tokens; the text-span states of all 25 layers are fused by the layer gate of Eq. (14).
IDM	0.56 B parameters. 28 transformer blocks of width 1152 (16 heads) with self-attention over action tokens and cross-attention to the visual latent tokens of the current and predicted frames. The flow time and a learned embodiment embedding enter through adaptive layer normalization; actions are 32-dimensional joint-space vectors, zero-padded across embodiments, and the proprioceptive state enters through a small MLP.

E.2 TEMPORAL SCHEDULE AND SAMPLE CONSTRUCTION

Frames and actions. The defaults of Eq. (12) are $(N_d, N_s, H) = (4, 2, 52)$ frames at 30 Hz, to which all sources are aligned, so H covers 1.73 s. The clip end $f_0 + T$ is the end of the current step-caption segment, with a small jitter and an occasional extension to the end of the following segment. Each sample carries 52 dense actions at the control rate and two sparse actions at the sparse frames; actions are absolute joint positions standardized per source, and sparse action tokens receive a loss weight of 0.1. Frames are resized to 256×384.

E.3 SVP PRETRAINING

The flow time is $t = s(\tau)$ with $\tau \sim \sigma(\mathcal{N}(0, 1))$ and the shift $s(\tau) = 2.7\tau/(1 + 1.7\tau)$; the endpoint is $r = s(\rho\tau)$ with $\rho \sim \mathcal{U}(0, 1)$. In 75% of the samples $r = t$, which reduces the objective to flow matching; the rest train the interval $[r, t]$. The conditioning frame's latent is mixed with Gaussian noise at a level drawn uniformly in $[0, 0.1]$. The Jacobian–vector product of Eq. (18) is computed exactly with forward-mode autodiff, and the detached correction $(t - r)\,\mathrm{d}u_\theta/\mathrm{d}t$ is clamped element-wise to $[-5, 5]$; the time embedders are zero-initialized so that the correction starts at zero. The losses of Eq. (22) are averaged over latent elements with dense and sparse frames weighted equally, and each per-sample loss is normalized as $\mathcal{L}/\,\mathrm{sg}[\mathcal{L} + 0.1]$ separately for the two heads. In pretraining only, 5% of the samples replace the instruction with a clean goal frame and learned null-text tokens.

The SVP is trained from random initialization with learning rate 10^{-4} (20,000-step warmup) and a per-GPU batch of 8 and later 12, with the data sources of Figure 8 introduced progressively; the learning rate is decayed linearly to zero at the end of pretraining.

E.4 IDM PRETRAINING

The IDM is trained on recorded futures with Eq. (5), a logit-normal flow time $t \sim \sigma(\mathcal{N}(0,1))$ with unit loss weight, and light Gaussian blur and noise on the visual latent tokens. It is trained with a per-GPU batch of 16 and learning rate 10^{-4} with a 5,000-step warmup on the 32,000-hour mixture of Figure 8. The controlled ablation of Section 5.2 uses the same IDM checkpoint.

E.5 KASO CO-TRAINING

Fused update. Each optimizer step draws one batch of instruction–video–action samples and accumulates three micro-steps on it: $\mathcal{L}_{\mathrm{SVP}}$ on the recorded video, $\mathcal{L}_{\mathrm{IDM}}$ on the recorded future, and $\mathcal{L}_{\mathrm{E2E}}^{\mathrm{sel}}$ on the selected generated future, with coefficients $(\lambda_{\mathrm{SVP}}, \lambda_{\mathrm{IDM}}, \lambda_{\mathrm{E2E}}) = (1, 0.1, 1)$ and the accumulated gradient divided by three. Every sample thus contributes to the recorded and the generated action loss once per update (a 1:1 mixing ratio). The flow generator and the IDM are fully trainable; the VLM and CoAE stay frozen.

Selection. The group size is $N = 4$ with $k = 1$ and the probe level is $t_p = 0.95$; the energy of Eq. (9) is averaged over the 52 dense action tokens with the CoAE-encoded recorded future as reference. Selection is a plain best-of-N choice: the single lowest-energy candidate receives the generated-future action loss with unit weight, and the remaining candidates are discarded. The selected loss uses a freshly drawn flow time and action noise.

Runs. The models at all data scales start from the same pretrained SVP and IDM and use learning rate 10^{-4} with a 5,000-step warmup and a per-GPU batch of 16 on nested slices of the co-training corpus; the model at the largest scale uses the full corpus of Figure 8. Each scaling model is trained for one epoch over its pool, so the number of optimizer updates grows with the data, as in standard data-scaling protocols.

For the controlled ablation of Section 5.2, all arms use the 300-hour mixture, the same pretrained components, and a per-GPU batch of 8. KASO was trained with learning rate 10^{-4} and a 1,000-step warmup; E2E+PT uses the same fused update with selection disabled ($k = N$), and E2E uses $\mathcal{L}_{\mathrm{E2E}}$ alone.

E.6 DEPLOYMENT

At deployment the policy receives the three camera views at 256×384, the proprioceptive state, and the instruction. The SVP produces the future latents in one forward pass at $(r, t) = (0, 1)$, and the IDM integrates its action flow with five Euler steps without guidance. The first 30 actions of the dense chunk are executed at the control rate before the next prediction. All results in Section 3 use the last checkpoint of the corresponding run with these fixed settings and no per-task checkpoint selection; the reset and scoring protocol is given in Appendix F.

E.7 TOY TWO-STAGE SYSTEM

Data-generating process. The toy experiment uses an unconditional two-stage mapping $\epsilon \mapsto \hat{z} \mapsto \hat{a}$ with $z \in \mathbb{R}^3$ and $a \in \mathbb{R}$. The (z_1, z_2) marginal is an equal-weight mixture of four Gaussians with means $\mu_{ij} = (1.5i, 1.8j)$ for $i, j \in \{-1, 1\}$ and per-axis standard deviations 0.04 and 0.18. The third coordinate is exactly zero for every recorded sample. We verify at runtime that the largest empirical deviation of a component weight from 0.25 is below 0.02. The action is deterministic, $a = z_1 z_2/(1.5 \times 1.8)$, so the four video modes map to two balanced action modes at $a = \pm 1$.

Networks and initialization. The generator A is a time-conditioned velocity field implemented as a tanh MLP with widths 4–24–24–3 and input $[x, t]$. Its hidden layers use Xavier-uniform initialization with gain 0.65, its output layer uses gain 0.12, and all biases are zero. The deterministic IDM proxy B is a tanh MLP with widths 3–24–24–1; its hidden and output Xavier gains are 0.80 and 0.50, respectively, with zero biases. Before entering B, each video coordinate is divided by $(1.5, 1.8, s_z)$ with $s_z = 0.2$. All four arms start from identical copies of A and B; a runtime check confirms that their step-0 outputs agree exactly. The point-estimate IDM exposes conditional-mean regression directly while retaining the video–action compatibility problem studied in the full system.

Shared optimization. For flow matching, we draw $t \sim \mathcal{U}(0,1)$ and use the linear bridge $x_t = (1-t)\epsilon + tz$ with velocity target $z - \epsilon$. Generated samples are obtained with 48 left-endpoint Euler steps, evaluating the velocity at $t_i = i/48$. Every arm uses AdamW for 7,000 steps with learning rate 2×10^{-3}, weight decay 10^{-5}, batch size 256, and global gradient-norm clipping at 5. The learning rate is halved every 650 steps starting at step 1,900. The pipeline weight ρ_s is zero through step 300, increases linearly to one over the next 300 steps, and remains one thereafter.

Let $\mathcal{L}_{\mathrm{FM}}$ denote the video flow-matching loss, $\mathcal{L}_{\mathrm{real}} = \mathrm{MSE}(B(z), a)$ the IDM loss on recorded pairs, and $\mathcal{L}_{\mathrm{pipe}} = \mathrm{MSE}(B(A(\epsilon)), a)$ the generated-video action loss. The four objectives are

$$\begin{aligned} \text{Decoupled Pretraining:} &\quad \mathcal{L}_{\mathrm{FM}} + \mathcal{L}_{\mathrm{real}}, \\ \text{End-to-End Co-training:} &\quad \rho_s \, \mathcal{L}_{\mathrm{pipe}}, \\ \text{E2E + Pretraining Losses:} &\quad 0.3 \, \mathcal{L}_{\mathrm{FM}} + 0.1 \, \mathcal{L}_{\mathrm{real}} + \rho_s \, \mathcal{L}_{\mathrm{pipe}}, \\ \text{KASO:} &\quad 0.3 \, \mathcal{L}_{\mathrm{FM}} + 0.1 \, \mathcal{L}_{\mathrm{real}} + \rho_s \, \mathcal{L}_{\mathrm{sel}}^{(8)}, \end{aligned}$$

where $\mathcal{L}_{\mathrm{sel}}^{(8)}$ is the pipeline loss restricted to the selected candidate, defined below. Decoupled Pretraining keeps unit weights on both pretraining losses, whereas the connected arms reduce them to 0.3 and 0.1; this weighting favors the baseline.

Selection and replay. For each batch row, the toy KASO arm draws eight generator noises and integrates all eight candidates without gradients. It scores candidate k by $\big(B(\hat{z}_k) - a\big)^2$, selects the minimum-energy candidate, and then re-integrates from that candidate's retained noise with gradients enabled. Only this replayed trajectory contributes $\mathcal{L}_{\mathrm{sel}}^{(8)}$. We verify that the replayed trajectory matches the selected no-gradient trajectory exactly. The eight-way selection is training-only: evaluation draws one generator sample per input for every arm.

Evaluation and stress-test controls. Figure 6 uses seed 0 and evaluates all arms on the same fixed batch of 12,000 Gaussian noises. Action densities are 220-bin normalized histograms on $[-1.7, 1.7]$, smoothed by a discrete Gaussian with standard deviation 1.6 bins (approximately 0.025 action units); all panels share the same vertical scale.

The choice $s_z = 0.2$, rather than the default value 1, creates a controlled off-manifold sensitivity stress test by amplifying the IDM's response to the third coordinate fivefold. It is shared by all four arms and does not change any recorded input because recorded data satisfy $z_3 = 0$. At seed 0, the IDM RMSE on recorded pairs is 0.0112 for both $s_z = 0.2$ and $s_z = 1.0$. Furthermore, the Decoupled arm has the lowest recorded-pair IDM error and the lowest video sliced-W_1 distance among the learned arms (0.068, versus 0.103 for KASO), but fails when its IDM is applied to generated off-manifold video.

The displayed figure is one run from a nontrivial seed spread. Across seeds 0–4, the maximum action-density height of Decoupled Pretraining relative to the ground-truth maximum is 0.240, 0.617, 0.647, 0.955, and 1.035, respectively. We therefore interpret Figure 6 as a controlled illustration of the off-manifold failure mechanism rather than as a seed-averaged performance benchmark.

F Full Results

F.1 Atomic-Skill Taxonomy and Evaluation Details

Our real-robot evaluation suite contains 100 atomic task instances in 20 skill groups (Table 8). Each task is evaluated on both G1-OP and G2-90D using the same instruction and an observable, instruction-specific terminal success criterion fixed before evaluation. For example, Pick requires a successful grasp and lift of the specified object, Place requires the object to reach the instructed receptacle, and Close requires the target to be fully closed. The analogous terminal condition is used for each remaining action type.

For every task, embodiment, and training-data scale, we run 10 trials under a standardized reset procedure. Within each task and embodiment, the reset configurations are matched across checkpoints; scenes, backgrounds, lighting, and test object instances are excluded from pretraining and co-training. We report empirical success rate (SR), i.e., the percentage of successful trials. The task names below

are compact descriptions; the evaluation uses the full instruction wording and the indicated arm configuration.

Table 8: Atomic-task taxonomy for the 100-task real-robot zero-shot OOD suite. Parenthetical counts sum to 100. Unless an arm or both arms are explicitly specified, the task is executed with the right arm.

Skill group	Atomic task instances
Pick (13)	Red apple; toothpaste; red cube; water bottle (two scene variants); marker pen from water cup; toothbrush; water cup; controller; towel; stapler; whisk; towel from rod.
Place (6)	Apple into container; blue cube into plate; orange into small bowl; marker pen into pen holder; red cube into paper cup; marker pen into open drawer.
Wipe (3)	Plate with rag; water stain with sponge; table with eraser.
Stack (7)	Red cube on yellow block; paper cup on paper cup; bowl on bowl; coaster on coaster; cola can on cola can; cutlery-box lid; trash-can lid.
Pass (5)	Remote control, left-to-right; orange ball, right-to-left; sponge eraser, left-to-right; Rubik's cube, right-to-left; cola can, left-to-right.
Fold (4)	Towel, one arm; towel, both arms; rag twice, both arms; unfold towel, both arms.
Straighten (5)	Water bottle (two scene variants); water cup (two scene variants); soda drink.
Pour (5)	Fruit pieces into plate; sesame into plate; sugar into glass cup; beans into water cup; small cubes into bowl.
Push (8)	Red block left; paper cup to center; black box left; kettle forward; book to table edge; tissue pack into tissue box; close drawer; roll bottle.
Insert (6)	Bread into bread machine; flower into vase; straw into cup; plug into power strip; remove book from shelf; place book into shelf.
Close (3)	Tissue-box lid; blue-box lid; kettle lid.
Separate (3)	Separate paper cups, both arms; uncap signature pen with the opposite arm holding the pen; unscrew bottle cap, both arms.
Press (4)	Lamp switch; bread-machine button; stapler; calculator button.
Pull (6)	Remove pen cap; pull tissue from bag; pull tray forward; pull cable from box; unplug charger while the opposite arm holds the power strip; open drawer.
Hang (4)	Remove towel from rod; hang towel on rod; hang rag on storage box, left arm; hang hat on coat rack.
Tear (2)	Open envelope, left arm; tear one sheet from paper-towel roll, both arms.
Flip (6)	Close book; open book; turn book face-up; flip plate; flip coaster; flip packaged tissues.
Sweep (3)	Sweep fruit cubes into dustpan with broom, both arms; sweep fruit cubes by hand; sweep fruit cubes into dustpan with brush while the opposite arm holds the dustpan.
Zip (3)	Zip file bag, both arms; unzip file bag, both arms; unzip pencil case while the opposite arm holds the case.
Shake/Stir (4)	Shake glass bottle; shake bottle to check remaining amount; stir coffee with chopstick; stir water with spoon.

F.2 Task-Level Data-Scaling Results

Table 9 reports the empirical SR for every task, data scale, and embodiment. G1-OP and G2-90D results are shown in separate four-column blocks, with every value computed over 10 trials. The columns correspond to the nested co-training pools used in Section 3.1. The group headings and task descriptions match the taxonomy in Table 8.

Between the 300- and 30,000-hour models, tasks with nonzero empirical SR increase from 39 to 76 on G1-OP and from 24 to 72 on G2-90D.

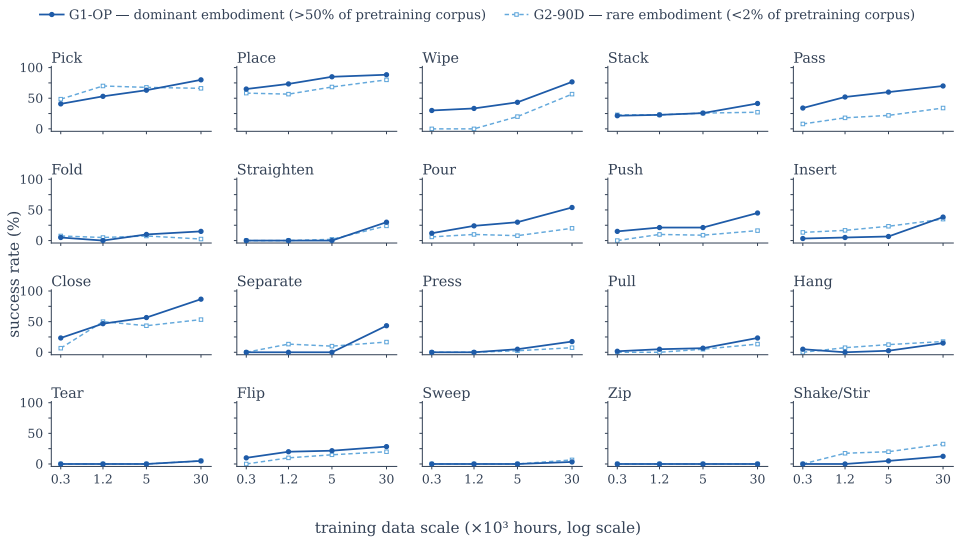

Figure 9: Skill-group zero-shot OOD scaling trajectories for G1-OP (more than 50% of the co-training mixture) and G2-90D (less than 2%). Each panel plots one group's mean SR (values in Table 1) across the 300-, 1,200-, 5,000-, and 30,000-hour models.

Table 9: Task-level zero-shot OOD success rate (SR, %) across data scales. Every G1-OP and G2-90D result is shown separately; each SR is computed over 10 trials, and all 100 tasks are included.

Task	G1-OP SR (%)				G2-90D SR (%)			
	0.3k h	1.2k h	5k h	30k h	0.3k h	1.2k h	5k h	30k h
Pick (13 tasks)								
pick up red apple	90	100	100	100	60	100	100	100
pick up toothpaste	0	20	40	80	30	40	60	60
pick up red cube	80	80	70	90	70	70	50	50
pick up water bottle	40	50	70	90	60	100	90	100
pick up water bottle (2)	50	70	90	100	100	100	100	100
pick pen from water cup	0	10	30	60	0	30	40	40
pick up toothbrush	30	30	40	50	0	40	30	30
pick up water cup	20	40	40	80	40	60	50	50
pick up controller	10	30	60	50	80	100	80	40
pick up towel	80	90	80	100	70	80	80	70
pick up stapler	70	100	80	90	80	80	70	80
pick up whisk	60	50	70	80	40	50	70	70
pick towel from rod	0	20	50	70	0	60	60	70
Place (6 tasks)								
put apple into container	100	100	100	100	100	100	90	90
put blue cube into plate	100	100	100	100	90	90	100	100
put orange into bowl	80	100	100	100	60	60	80	80
put pen into holder	20	20	40	50	0	40	30	70
put cube into paper cup	30	60	80	80	30	0	40	60
put pen into drawer	60	60	90	100	70	50	70	80
Wipe (3 tasks)								
wipe plate with rag	20	20	30	70	0	0	10	60
wipe stain with sponge	30	40	40	80	0	0	20	50
wipe table with eraser	40	40	60	80	0	0	30	60
Stack (7 tasks)								
stack cube on block	0	10	20	40	0	0	0	10
stack paper cups	0	0	0	30	0	0	0	10
stack bowls	70	80	80	90	90	90	100	90
stack coasters	80	70	80	100	70	70	80	70
stack cola cans	0	0	0	10	0	0	0	10
cover cutlery box lid	0	0	0	0	0	0	0	0
cover trash can lid	0	0	0	20	0	0	0	0
Pass (5 tasks)								
pass remote control	40	60	60	80	0	20	30	60
pass orange ball	70	70	80	100	0	0	0	0
pass sponge eraser	60	90	100	100	40	50	60	80
pass rubik's cube	0	0	0	0	0	0	0	0
pass cola can	0	40	60	70	0	20	20	30

Table 9: Task-level zero-shot OOD success rate across data scales (continued).

Task	G1-OP SR (%)				G2-90D SR (%)			
	0.3k h	1.2k h	5k h	30k h	0.3k h	1.2k h	5k h	30k h
Fold (4 tasks)								
fold towel (one arm)	0	0	40	50	30	20	30	10
fold towel (two arms)	0	0	0	10	0	0	0	0
fold rag twice	0	0	0	0	0	0	0	0
unfold folded towel	20	0	0	0	0	0	0	0
Straighten (5 tasks)								
straighten water bottle	0	0	0	10	0	0	0	10
straighten water bottle (2)	0	0	0	40	0	0	10	50
straighten water cup	0	0	0	60	0	0	0	40
straighten water cup (2)	0	0	0	30	0	0	0	0
straighten soda drink	0	0	0	10	0	0	0	20
Pour (5 tasks)								
pour fruits into plate	20	40	50	70	30	20	20	30
pour sesame into plate	30	40	30	60	0	0	10	40
pour sugar into cup	10	20	30	50	0	0	0	10
pour beans into cup	0	20	40	50	0	0	10	10
pour cubes into bowl	0	0	0	40	0	30	0	10
Push (8 tasks)								
push block to left	30	40	0	70	0	0	0	10
push cup to center	0	20	30	60	0	20	30	50
push box to left	40	40	60	70	0	0	10	20
push kettle forward	30	20	20	50	0	60	20	30
push book to edge	0	0	0	30	0	0	0	0
push tissue pack in	0	0	0	0	0	0	0	10
close the drawer	20	50	60	70	0	0	0	0
roll the bottle	0	0	0	10	0	0	10	10
Insert (6 tasks)								
insert bread into machine	0	0	0	50	30	30	40	50
insert flower into vase	0	0	0	40	0	0	20	30
insert straw into cup	20	30	40	70	30	40	30	40
insert plug into strip	0	0	0	0	0	0	0	0
take book from shelf	0	0	0	60	0	0	20	50
put book into shelf	0	0	0	10	20	30	30	40
Close (3 tasks)								
close tissue box lid	10	30	40	70	0	50	40	40
close blue box lid	30	50	40	90	0	70	60	80
close kettle lid	30	60	90	100	20	30	30	40
Separate (3 tasks)								
separate paper cups	0	0	0	20	0	20	20	20
uncap signature pen	0	0	0	80	0	10	10	30
unscrew bottle cap	0	0	0	30	0	10	0	0
Press (4 tasks)								
press lamp switch	0	0	0	0	0	0	0	0
press bread machine button	0	0	10	30	0	0	0	0
press stapler to staple	0	0	10	30	0	0	10	20
press calculator button	0	0	0	10	0	0	0	10
Pull (6 tasks)								
pull off pen cap	0	0	0	20	0	0	0	20
pull tissue from bag	10	30	30	80	0	0	30	40
pull tray to front	0	0	10	30	0	0	0	10
pull cable from box	0	0	0	0	0	0	0	0
unplug charger from strip	0	0	0	0	0	0	0	10
open the drawer	0	0	0	10	0	0	0	0
Hang (4 tasks)								
take towel off rod	0	0	10	60	0	0	30	50
hang towel on rod	0	0	0	0	0	0	0	0
hang rag on box	20	0	0	0	0	30	20	20
hang hat on rack	0	0	0	0	0	0	0	0
Tear (2 tasks)								
tear open envelope	0	0	0	10	0	0	0	10
tear paper towel off roll	0	0	0	0	0	0	0	0
Flip (6 tasks)								
close the open book	0	50	70	90	0	0	20	30
open the book	0	0	0	0	0	0	0	0
flip book to front	0	0	0	0	0	0	0	0
flip plate over	0	0	0	0	0	0	0	0
flip coaster to front	60	70	60	80	0	60	70	90
flip tissue pack over	0	0	0	0	0	0	0	0
Sweep (3 tasks)								
sweep cubes into dustpan	0	0	0	10	0	0	0	0
sweep cubes by hand	0	0	0	0	0	0	0	10

Table 9: Task-level zero-shot OOD success rate across data scales (continued).

Task	G1-OP SR (%)				G2-90D SR (%)			
	0.3k h	1.2k h	5k h	30k h	0.3k h	1.2k h	5k h	30k h
sweep cubes with brush	0	0	0	0	0	0	0	10
Zip (3 tasks)								
zip up file bag	0	0	0	0	0	0	0	0
unzip file bag	0	0	0	0	0	0	0	0
unzip pencil case	0	0	0	0	0	0	0	0
Shake/Stir (4 tasks)								
shake glass bottle	0	0	10	20	0	40	40	60
shake bottle to check	0	0	10	30	0	30	40	60
stir coffee with chopstick	0	0	0	0	0	0	0	0
stir water with spoon	0	0	0	0	0	0	0	10

F.3 Instruction-Following Prompts and Results

Protocol. The fine-grained grounding study of Section 3.3 evaluates the final checkpoint of the full KASO co-training run on G1-OP without any adaptation. The suite contains 59 instructions, 33 for Pick and 26 for Place, and every instruction is executed for five trials under the same reset procedure, giving 295 rollouts. A trial satisfies the Follow Score when the gripper contacts only the instruction-specified object (Pick) or the carried object contacts only the specified destination container (Place); SR additionally requires the complete Pick or Place behavior. The instruction-level Follow Score and SR reported in Table 10 are the fraction of the five trials that satisfy each criterion, so every value is a multiple of 20%. Averaged over all 295 rollouts, the Follow Score is 83.1% and the SR is 72.9%.

Scenes. Each of the five physical scene families in Figure 10 has a Pick and a Place variant, and each is arranged so that several actions are visually feasible while only the instruction-specified target or destination is correct. (i) *Object identity*: a cluttered tabletop of distinct household items (a badminton ball, a knife, a comb, a computer mouse, a television remote, a toy bear, a screwdriver, and a roll of tape) for Pick, and a white plate, a box, a bowl, and a cup as candidate destinations for Place. (ii) *Color*: fruits in four colors (red, orange, yellow, and green) for Pick, and plates in four colors (blue, pink, orange, and yellow) for Place. (iii) *Size*: blocks or cups in three sizes for Pick and containers in three sizes for Place, each evaluated in two layouts; the instructions use both extreme references (largest, smallest, larger) and the medium reference. (iv) *Shape*: rectangular, cylindrical, and round objects for Pick and containers of the same three shapes for Place. (v) *Position and order*: a row of identical cups (Pick) or containers (Place). The horizontal layout supports left/right references and the set-relative references *middle* and *second from the left/right*; the vertical layout supports near-side and far-end references.

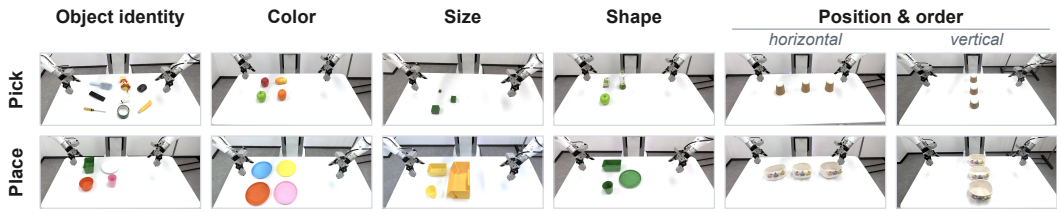

Figure 10: Controlled scenes for fine-grained instruction following, ordered as introduced in the text: object identity, color, size, shape, and position & order. Position and order share one setup, evaluated in a horizontal layout (left/right and ordinal references) and a vertical layout (near/far). Top row: Pick scenes; bottom row: Place scenes.

Dimension mapping. Each instruction is annotated with one raw subcategory, which maps to the six grounding dimensions as follows. Object identity and color are single subcategories. Size combines extreme references (largest, smallest, larger) and the medium reference. Position combines direct left/right references and near/far references. Order combines the middle reference and

ordinal references (second from the left or right). Table 10 lists the subcategory beside every instruction. Instructions are reproduced verbatim as issued to the policy, including minor typographical irregularities.

Table 10: Instruction-level results of the fine-grained grounding study (G1-OP, five trials per instruction). Follow and SR are percentages of the five trials.

Dimension	Instruction	Follow	SR
Pick (33 instructions)			
Object	pick up the white badminton ball with the right arm.	100	80
Object	pick up the yellow knife with the left arm.	100	80
Object	pick up the blue comb with the right arm.	100	80
Object	pick up the black mouse with the left arm.	100	100
Object	pick up the black tv controller with the right arm.	100	100
Object	pick up the toy bear with the left arm.	100	60
Object	pick up the yellow screwdriver with the right arm.	100	80
Object	pick up the green tape with the right arm.	100	100
Color	pick up the red fruit with the right arm.	100	60
Color	pick up the orange fruit with the right arm.	100	100
Color	pick up the yellow fruit with the right arm.	100	100
Color	pick up the green fruit with the right arm.	100	100
Size (extreme)	pick up the largest block with the right arm.	100	100
Size (extreme)	pick up the smallest block with the left arm.	100	100
Size (medium)	pick up the medium block with the right arm.	0	0
Size (extreme)	pick up the larger cup with the left arm.	100	80
Size (medium)	pick up the medium block with the left arm.	100	100
Size (extreme)	pick up the largest block with the right arm.	100	100
Size (extreme)	pick up the smallest block with the right arm.	100	80
Size (medium)	pick up the medium block with the right arm.	60	40
Shape	pick up the rectangular object with the right arm.	100	20
Shape	pick up the cylindrical object with right arm.	60	60
Shape	pick up the round object with right arm.	100	100
Shape	pick up the cylindrical object with left arm.	100	100
Position (left/right)	pick up the cup on the right side.	80	20
Order (ordinal)	Pick up the second cup from the right.	0	0
Order (middle)	Pick up the middle cup.	40	40
Order (ordinal)	Pick up the second cup from the left.	0	0
Position (left/right)	pick up the cup on the left side.	100	100
Position (left/right)	pick up the cup on the right of the table with right arm.	100	100
Position (left/right)	pick up the cup on the left of the table with left arm.	100	100
Position (near)	Pick up the cup on the near side of the table with the right arm.	60	40
Position (far)	pick up the farthest cup on the table with right arm.	100	100
Place (26 instructions)			
Object	place the object into the plate with the right arm.	100	100
Object	place the object into the box with the right arm.	100	100
Object	place the object into the bowl with the right arm.	100	80
Object	place the object into the cup with the right arm.	100	60
Color	place the object into the blue plate with the right arm.	100	100
Color	place the object into the pink plate with the right arm.	100	100
Color	place the object into the orange plate with the right arm.	100	100
Color	place the object into the yellow plate with the right arm.	100	100
Size (extreme)	place the object into the smallest container with left arm.	100	40
Size (extreme)	place the object into the largest container with right arm.	100	100
Size (medium)	place the object into the medium container with right arm.	0	0
Size (medium)	place the object into the medium container with left arm.	80	80
Size (extreme)	place the object into the smallest container with right arm	0	0
Size (extreme)	place the object into the largest container with right arm.	80	80
Size (medium)	place the object into the medium container with right arm.	100	100
Shape	place the object into the rectangular container with the right arm.	100	100
Shape	place the object into the cylindrical container with the right arm	80	40
Shape	place the object into the round container with the right arm	80	60
Shape	place the object into the round container with the left arm	100	100
Position (left/right)	place the object into the container on the right side with the right arm.	100	100

Table 10: Instruction-level results of the fine-grained grounding study (continued).

Dimension	Instruction	Follow	SR
Order (middle)	place the object into the middle container with the right arm.	20	20
Order (middle)	place the object into the middle container with the left arm.	40	40
Position (left/right)	place the object into the container on the left side with the left arm.	100	100
Position (far)	Place the object into the container at the far end of the table with the right arm.	100	100
Position (near)	Place the object into the container on the near side of the table with the right arm.	100	60
Order (ordinal)	place the object into the second container from the left.	20	20

Descriptor frequencies. The occurrence rates in Figure 5 are lexical matches over two corpora: the 95,010 human-written referring expressions of RefCOCOg (Mao et al., 2016), and the step-captioned Pick and Place instruction segments of the GE-Act 2.0 training corpus (approximately 3.7 million segments, simulation excluded). Table 11 gives the matching rule for every dimension together with the RefCOCOg counts and the resulting shares. Non-object descriptors are counted independently, so one expression can contribute to several dimensions and the shares are not a partition; object identity is defined as the residual expressions that match none of the other five rules in RefCOCOg, and as bare object noun phrases without a descriptor in the training corpus. Size is the one dimension for which simulation instructions are included in the training-corpus share, because superlative size references occur almost exclusively there; its 0.96% comprises 0.68% extreme and 0.28% medium references.

Table 11: Lexical matching rules and descriptor frequencies used in Figure 5. RefCOCOg counts are out of 95,010 expressions.

Dimension	Matching rule	RefCOCOg count	RefCOCOg (%)	GE corpus (%)
Object	Residual: no color, size, shape, position, or order descriptor	39,474	41.5	58.3
Color	Explicit color words; *orange* as a fruit name excluded	42,161	44.4	29.6
Size	*large*, *small*, *big*, *tiny*, *medium* and comparative/superlative forms	5,299	5.6	0.96
Shape	*round*, *square*, *rectangular*, *cylindrical*, *triangular*	574	0.6	0.67
Position	*left*, *right*, *near*, *far*; body-side phrases such as *right hand* excluded unless another spatial relation is present	11,927	12.6	13.8
Order	*middle*, *center*, ordinals, *leftmost*, *rightmost*; generic *between* excluded	2,440	2.6	0.13

G Additional Real-Robot Visualizations

Additional skill-level observations. Zip records no success in any of its 240 trials across both embodiments and all four scales. Each of its three tasks requires a precise pinch on a small slider and a steady pull along its track while the other arm stabilizes the item. The group receives only 16.4 hours across the corpus, the second-lowest coverage of any skill group. The largest G1-OP gains are Close (+63.3 points), Wipe (+46.7), Separate (+43.3), Pour (+42.0), and Pick (+39.2).

Data-scarce embodiment and hardware effects. Figure 12 shows representative G2-90D successes; within the skill-attributed G2-90D data at the largest scale, 10 of 20 groups contain fewer than five hours, including six with fewer than two and four with fewer than one. Pass reaches 34.0% SR with 3.9 hours of G2-90D data. Shake/Stir scores higher on G2-90D than on G1-OP. In rollout inspection, G1-OP more often drops cups during shaking when its gripper fails to maintain a stable hold on large containers, whereas G2-90D's wider gripper provides a more secure grasp. This failure pattern suggests that the absolute embodiment gap is partly attributable to end-effector geometry, but does not explain the within-G2-90D scaling trend, which compares models on the same hardware.

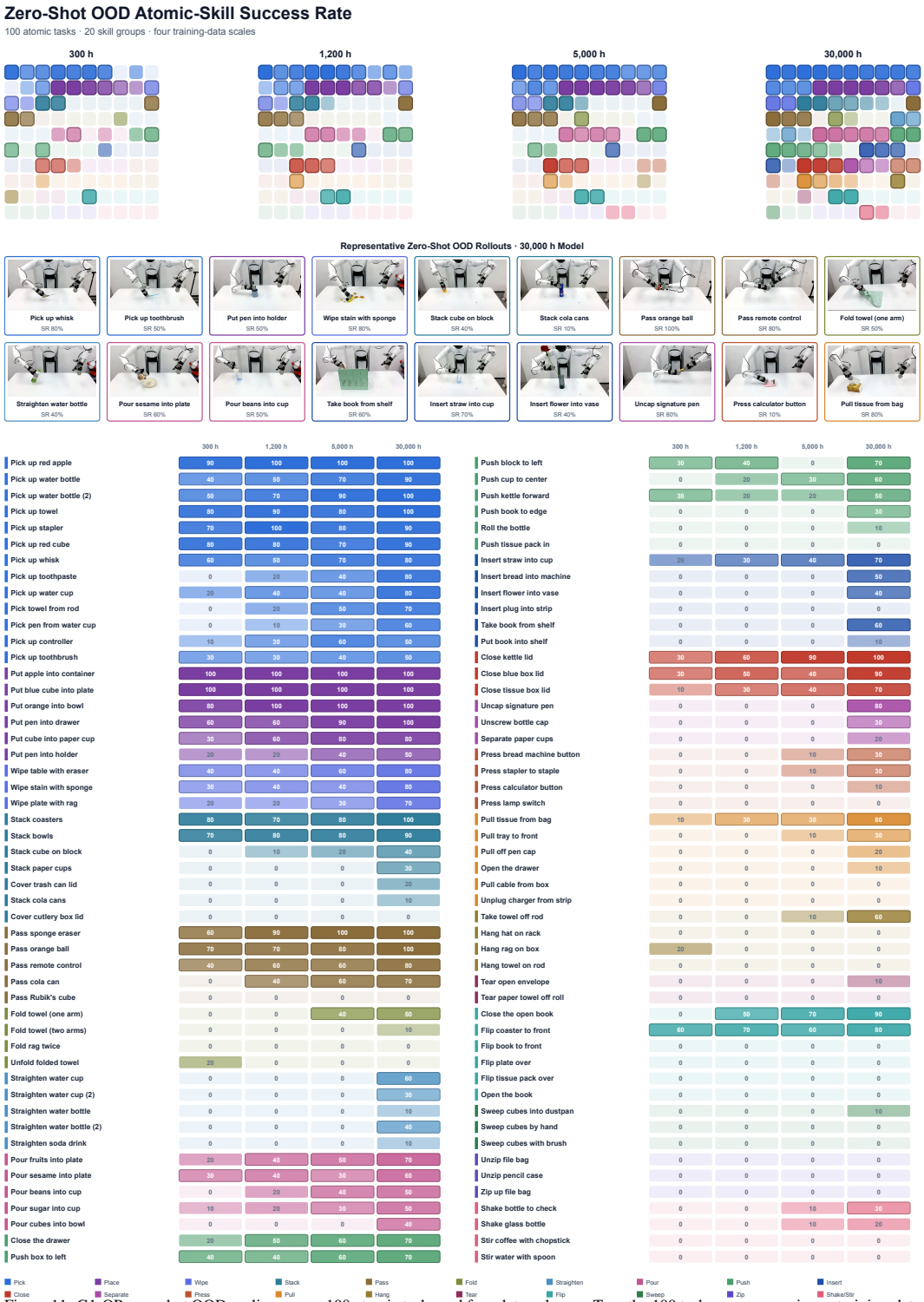

Figure 11: G1-OP zero-shot OOD scaling across 100 atomic tasks and four data volumes. Top: the 100 tasks as one mosaic per training-data scale, with tile intensity denoting per-task SR. Middle: representative zero-shot OOD rollouts of the 30,000-hour model. Bottom: per-task SR at every scale, grouped by skill; colors follow the skill-group legend.

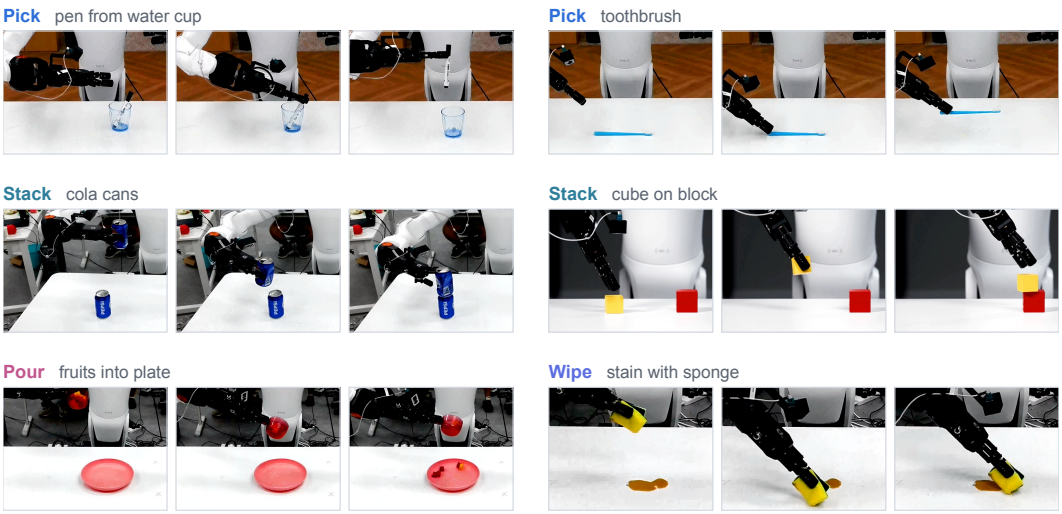

Figure 12: Representative successful zero-shot OOD rollouts on G2-90D across skill groups, each shown as three frames spanning the start, mid-execution, and outcome of the trial. Skill labels are colored by skill group, matching the palette of Figure 11.

G.1 Instruction Following under Behavioral and Semantic Conflicts

We next examine two harder forms of instruction conflict: behavioral conflict, where a revised command opposes a robot–scene state already committed toward the previous behavior, and semantic conflict, where the requested relation contradicts a conventional scene association. In preliminary experiments with earlier, weaker model variants, we observed a characteristic failure mode: once the end effector had approached an object or initiated an action, the policy often continued the visually implied behavior even after the instruction changed. A plausible explanation is a strong state–action prior in the training data: trajectories in which the gripper has reached object A typically continue by grasping A rather than redirecting to B. The behavioral-conflict tests therefore ask whether the active instruction can override this state-conditioned inertia.

Behavioral conflicts. We replace either the target or the commanded arm only after the end effector executing the original instruction is already near the target or has begun to grasp it, such that the evolved physical state favors completing the previous behavior.

- *Target switch.* The commanded object changes from A to B, and later from B to C, each time when the end effector is already close to, or beginning to grasp, the previous target (Figure 13a). Immediately after each switch, the robot retains brief residual motion toward the previous target, but subsequently redirects its trajectory and finally grasps C.
- *Arm switch.* The command changes from the right arm on the green cup to the left arm on the orange cup when the right end effector is already at the green cup (Figure 13b). The robot disengages the right arm and completes the new command with the left arm with little visible hesitation.

Semantic conflicts. We retain familiar objects and motor primitives but request a physically feasible relation that conflicts with the conventional object relations implied by the scene. This construction tests whether behavior is controlled by the explicit instruction or by familiar visual–action associations memorized from training. For example, we place a cup beside a shoe and a shoebox, then instruct the robot to put the cup into the shoebox (Figure 13c). The policy completes the unusual placement, following the explicit object–destination relation rather than defaulting to the more conventional scene association.

The demonstrations reveal distinct qualitative dynamics: arm designation can be revised with little hesitation, whereas a target change near contact requires recovery from the previous trajectory. Successful counter-conventional placement further provides qualitative evidence that an explicit object–destination command can override a familiar scene prior. These cases complement the static

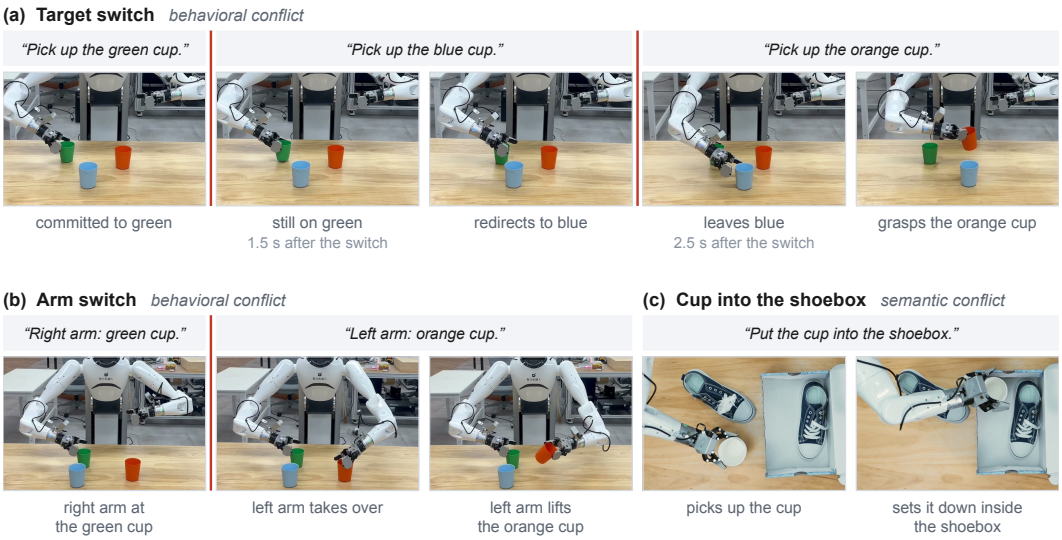

Figure 13: Instruction-conflict stress tests. Each panel shows the active instruction, frames from one rollout, and the outcome; red rules mark where the instruction changed. (a) Target switch: the commanded cup changes from green to blue, then to orange, while the gripper is already at the previous target; the robot redirects each time and grasps the final target. (b) Arm switch: the command moves from the right arm on the green cup to the left arm on the orange cup, and the left arm completes it. (c) Semantic conflict: the robot places a cup into a shoebox, following the explicit object–destination relation over the conventional scene association. Illustrative only; not part of the quantitative evaluation.

dimension-level scores by exposing how instruction-dependent control behaves under temporal and semantic conflict.