

GE-Act 2.0: Pretraining and Scaling a World-Action Model for Robotic Manipulation

AgiBot Research Team

<https://ge-act-v2.github.io/>

Abstract

World-action models predict future states to guide robot actions, enabling learning from both action-free video and action-labeled interaction. Yet most inherit pretrained video generators, leaving pretraining and scaling of world-action models underexplored. We introduce Genie Envioner Act 2.0 (GE-Act 2.0), a world-action model in which every trainable generative and action component is initialized from scratch on manipulation data. GE-Act 2.0 combines a control-oriented autoencoder (CoAE), a single-step visual planner (SVP), and an inverse dynamics model (IDM). CoAE retains action- and instruction-relevant information under aggressive compression, while SVP produces a completed future state in one differentiable pass, allowing visual planning and inverse dynamics to be pretrained separately on complementary data. These components are then jointly trained using knowledge-aligned selective optimization (KASO), which reduces mismatched supervision by selecting only predicted future states judged behaviorally compatible with the recorded action. We evaluate pretrained checkpoints directly, without per-task fine-tuning, on 100 tasks across 20 manipulation skill groups, with held-out scenes, backgrounds, lighting conditions, and object instances. Scaling co-training data from 300 to 30,000 hours raises success from 17.1% to 44.1% on G1-OP and from 13.4% to 31.1% on G2-90D. Despite comprising less than 2% of the co-training data, G2-90D improves by 17.7 points, suggesting useful cross-embodiment transfer from the broader corpus. The gains span 19/20 and 18/20 skill groups, respectively, and skill-specific coverage strongly correlates with zero-shot out-of-distribution (OOD) success (Pearson $r = 0.80$; Spearman $\rho = 0.85$). Under the same zero-shot OOD protocol, the model grounds object, color, shape, and position references in at least 90% of trials; qualitative stress tests further show that it follows explicit instructions even when they conflict with an already-committed behavior or a conventional scene association.

1 Introduction

Large-scale pretraining has driven successive breakthroughs in language, multimodal understanding, and visual generation. In robot manipulation, vision-language-action (VLA) policies map observations and instructions directly to motor commands at scale (Black et al., 2024; Brohan et al., 2022; Physical Intelligence et al., 2025), but their direct action objective does not explicitly require modeling the physical dynamics underlying interaction. World-action models (WAMs) are emerging as a new paradigm: they predict how a scene may unfold and use that prediction to support action generation (Chen et al., 2026; Huang et al., 2025; Li et al., 2026; Liao et al., 2025; Wang et al., 2026; Ye et al., 2026b; Zhang et al., 2026). 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 (NVIDIA et al., 2025a; Wan Team et al., 2025) and devotes substantial effort to coupling them with action models (Liao et al., 2025; Motubrain Team et al., 2026; Wang et al., 2026; Wu et al., 2024b; Ye et al., 2026b), rather than to pretraining and scaling the WAM as a whole. This leaves two system-level questions unresolved: how to pretrain visual generation and inverse dynamics on their

Every interaction is training data

One world-action model learns from action-free, instruction-free and fully labeled data, and scales with all of it.

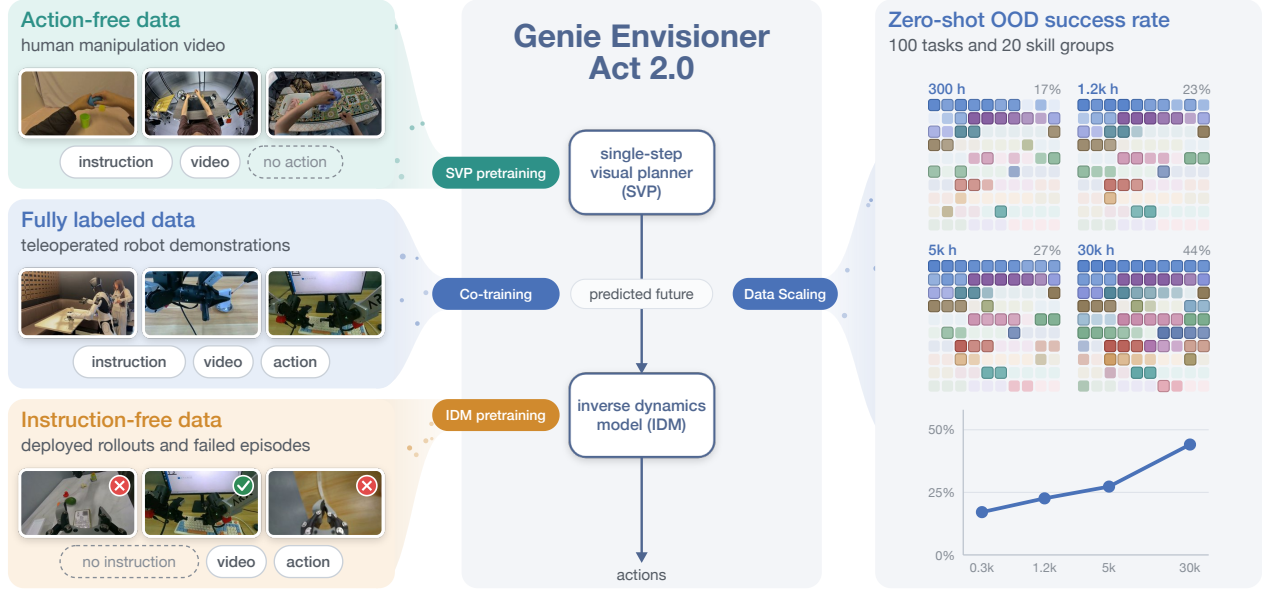


Figure 1: GE-Act 2.0 pretrains its visual planner on video without action labels and its inverse dynamics model on trajectories without language instructions, then co-trains the full system on fully labeled robot demonstrations. Scaling co-training data from 300 to 30,000 hours improves zero-shot OOD success across a 100-task real-robot suite. Each tile in the right panel is one task, coloured by skill group and shaded by success rate; the curve is the mean over all tasks.

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.

Genie Envisioner Act 2.0 (GE-Act 2.0) therefore adopts a simple decoupled two-stage architecture and focuses on three design problems for scalable WAM pretraining: constructing the latent space, predicting actionable future visual states in that space, and aligning those states with recorded action supervision.

Representation: A latent space for robot manipulation must make future prediction efficient while retaining the visual information needed for action prediction. Most systems inherit a space from a standard video autoencoder (NVIDIA et al., 2025a; Wan Team et al., 2025), optimized for reconstruction fidelity rather than control. We therefore introduce a control-oriented autoencoder (CoAE) that learns an aggressively compressed latent space through reconstruction and multi-teacher feature alignment, preserving the information needed for both visual prediction and action recovery (Section 3.2).

Generation: Visual generation and inverse dynamics can exploit different types of data. Visual-generation pretraining can use action-free video, whereas IDM pretraining can use instruction-free robot trajectories such as failed attempts and deployment rollouts, which are difficult to exploit under conventional policy-training objectives. However, most prior WAMs do not support standalone IDM pretraining on this broader interaction data (Hu et al., 2025; Liao et al., 2025; Ye et al., 2026b); in Section 3.3.1, we trace this limitation to the cost of multi-step generation. We therefore introduce a single-step visual planner (SVP), which produces a complete future in one differentiable flow-generator pass and allows the SVP and IDM to be pretrained separately on complementary data before being connected through end-to-end co-training (Section 3.3).

Alignment: Even when the robot sees the same scene and receives the same instruction, there may be several correct ways to complete the task. In imitation learning, a recorded demonstration contains only one of these valid behaviors, while an independently generated future may depict another. Pairing that future with the recorded action gives the IDM mismatched visual and action supervision, creating what we term **the validity gap**. Repeated

mismatches can erase distinct action modes and reduce the action diversity available for later exploration. KASO (knowledge-aligned selective optimization) samples multiple futures and co-trains only on candidates that the active IDM judges compatible with the recorded mode, preserving broader action-space coverage for subsequent post-training methods such as reinforcement learning (Section 3.4).

Evaluating what pretraining contributes is therefore the central empirical question for this design. Many evaluations of WAMs and VLA policies use downstream supervised fine-tuning (SFT) on the target tasks or embodiments (Black et al., 2024; Brohan et al., 2022; Cheang et al., 2024; Kim et al., 2024b; Li et al., 2026; Wu et al., 2024b; Ye et al., 2026a; Zhang et al., 2026; Zhou et al., 2026). Some further evaluate under the same lighting, backgrounds, and object sets used for SFT, making the test effectively in-domain. These protocols characterize the final policy, but entangle capability acquired during pretraining with capability added downstream. We instead evaluate GE-Act 2.0 directly under out-of-distribution (OOD) conditions, without task- or embodiment-specific SFT, and keep the protocol fixed across scaling checkpoints. This design measures the capability already present after pretraining and how it changes with training data.

Contributions.

- We introduce GE-Act 2.0, a world–action model designed for pretraining from scratch on manipulation data. Its control-oriented autoencoder provides a compact representation for visual prediction and action recovery, while its single-step visual planner allows visual planning and inverse dynamics to be pretrained separately on complementary data and subsequently connected through end-to-end co-training.
- We identify the validity gap, a supervision mismatch that arises when a predicted future depicts a behavior different from the recorded action, and propose KASO to co-train GE-Act 2.0 using action-compatible predicted futures.
- We demonstrate systematic scaling of GE-Act 2.0’s pretrained zero-shot OOD capability: increasing manipulation data produces broad gains across 100 real-robot tasks and two embodiments without task-specific fine-tuning. The results further reveal a strong relationship between skill coverage and success, as well as useful transfer to a sparsely represented embodiment. Under the same OOD protocol, complementary language experiments demonstrate fine-grained referential grounding and instruction following when explicit commands conflict with already-committed behaviors or conventional scene associations.

2 Related Work

2.1 Generalist Policies and World-Action Models

Generalist robot policies scale language-conditioned control across broad task collections (Black et al., 2024; Brohan et al., 2022; Generalist AI Team, 2025, 2026; Kim et al., 2024b; Liu et al., 2025; Octo Model Team et al., 2024; Physical Intelligence et al., 2025). World-action models connect visual prediction to control through several interfaces (Wang et al., 2026). 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 (Cheang et al., 2024; Motubrain Team et al., 2026; Wu et al., 2024b; Ye et al., 2026b); and feature-coupled designs decode actions from internal representations of a generative backbone (Hu et al., 2025; Liao 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 GE-Act 1.0 and allows the generator and IDM to pretrain on distinct data before end-to-end co-training.

2.2 Representations for Robot Video and Control

Most latent video generators operate in autoencoder spaces developed for content reconstruction (NVIDIA et al., 2025a; Wan Team et al., 2025). 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 (Chen et al., 2025c; Ye et al., 2025); 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 (Cheang et al., 2024; Wu et al., 2024b), 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 (Yao et al., 2025b; Yu et al., 2025). 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.

2.3 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 (Chen et al., 2026; Guo et al., 2025; Huang et al., 2025; Jang et al., 2025; Jiang et al., 2025; Li et al., 2026; NVIDIA et al., 2025a; Qiu et al., 2026; Zhang et al., 2026; Zhu et al., 2025). One- and few-step generation has a broader lineage in images: progressive and distribution-matching distillation compress a pretrained teacher (Salimans and Ho, 2022; Yin et al., 2024a,b); consistency models enforce agreement along the probability-flow trajectory (Kim et al., 2024a; Luo et al., 2023; Song et al., 2023); and rectified or shortcut flows learn paths intended for small integration budgets (Frans et al., 2025; Lipman et al., 2023; Liu et al., 2023b, 2024). 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.

2.4 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; Hu et al., 2025; Qiu et al., 2026; Wu et al., 2022). 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 3.4). 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.

2.5 Evaluating Pretrained Robot Models

Robot models are commonly evaluated after task-specific fine-tuning (Cheang et al., 2024; Chen et al., 2026; Li et al., 2026; Liao et al., 2025; Wu et al., 2024b; Ye et al., 2026a; Zhang et al., 2026; Zhou et al., 2026), while fewer systems report zero-shot results (Generalist AI Team, 2025, 2026; Physical Intelligence et al., 2025; Zhou et al., 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 5.1). Simulation and world-model benchmarks (James et al., 2020; Liu et al., 2023a; Shang et al., 2026) provide horizontal reference points rather than direct evidence for the capability acquired by our manipulation-video pretraining.

3 GE-Act 2.0

3.1 System Overview

GE-Act 2.0 is a world-action model designed for from-scratch pretraining and scaling with heterogeneous video and robot-interaction data. As summarized in Figure 2, it comprises a control-oriented autoencoder (CoAE), a single-step visual planner (SVP), and an inverse dynamics model (IDM). The SVP serves as the system’s generative world model. We train CoAE, and we randomly initialize and pretrain every other trainable component from scratch: the flow generator in the SVP, and the IDM.

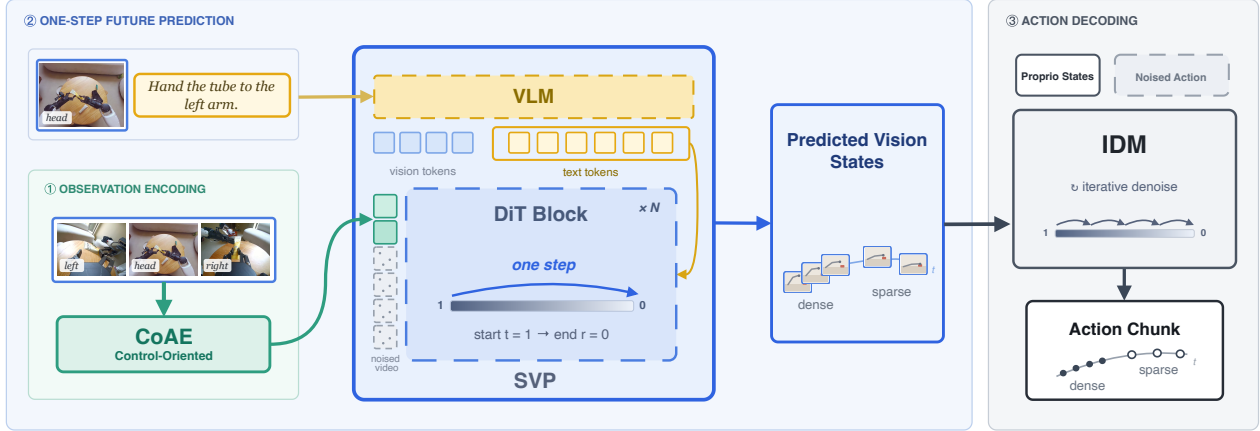


Figure 2: GE-Act 2.0 system architecture. The SVP grounds the instruction in the current observation and fully denoises multi-view future visual latents in a single flow-generator pass; the IDM combines current and predicted latents with proprioception to produce dense and sparse action sequences.

CoAE first encodes the current multi-view observations into compact visual latents. Conditioned on these latents and the instruction, the SVP grounds the instruction in the current scene and fully denoises dense and sparse future visual latents in a single flow-generator pass. The fully denoised future visual states are passed to the IDM, which combines current and predicted visual latents with proprioception to predict dense and sparse sequences of action; the controller executes the dense chunk, while the sparse sequence provides auxiliary far-horizon targets during training.

The remainder of this section develops these components and their interaction. Section 3.2 introduces CoAE and its compact visual space, Section 3.3 develops the SVP, and Section 3.4 connects generated visual futures to action learning through the IDM and KASO.

3.2 A Control-Oriented Autoencoder for World–Action Models

A latent space for a world-action model determines how efficiently and effectively a future can be predicted, as well as what information the action model can recover from that prediction. Yet most existing WAMs inherit this space from a pretrained video generator (Chen et al., 2026; Guo et al., 2025; Jang et al., 2025; Li et al., 2026; Zhang et al., 2026). This inheritance creates two mismatches. First, the modest spatial compression of standard video autoencoders leaves dense latent grids, imposing a large token burden on every future prediction inside the control loop. Second, these autoencoders are trained primarily to reconstruct generic images and videos, rather than explicitly designed to support downstream visual planning and action prediction.

Our control-oriented autoencoder (CoAE) addresses both limitations by learning an aggressively compressed, control-oriented latent space from manipulation video. Its training uses pixel reconstruction together with alignment to complementary visual teachers, and its latents are consumed directly by both the SVP and the inverse-dynamics model. We next describe its training objective and evaluate the resulting representation through action recovery and instruction grounding.

3.2.1 Beyond Reconstruction with Multi-Teacher Alignment

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 $16\times$ (NVIDIA et al., 2025a; Wan Team et al., 2025), CoAE uses a $64\times$ spatial downsampling factor. At the 256×384 frame resolution used by the SVP (Section 3.3), 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:

$$\mathcal{L}_{\text{rec}} = \|\hat{o} - o\|_2^2 + \lambda_{\text{perc}} \mathcal{L}_{\text{perc}}(\hat{o}, o) + \lambda_{\text{gan}} \mathcal{L}_{\text{gan}}(\hat{o}). \quad (1)$$

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., 2025a,b), 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 (Mur-Labadia et al., 2026; Siméoni et al., 2025; Tschannen et al., 2025; Zhai et al., 2023). These teachers provide complementary language-aligned semantics, spatiotemporal structure, and dense visual features, respectively. The alignment objective is

$$\mathcal{L}_{\text{align}} = \sum_{k=1}^3 \lambda_k (1 - \cos(g_k(z), G_k(o))). \quad (2)$$

The complete training objective is $\mathcal{L}_{\text{CoAE}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{align}}$. Figure 3 summarizes the architecture and training objectives.

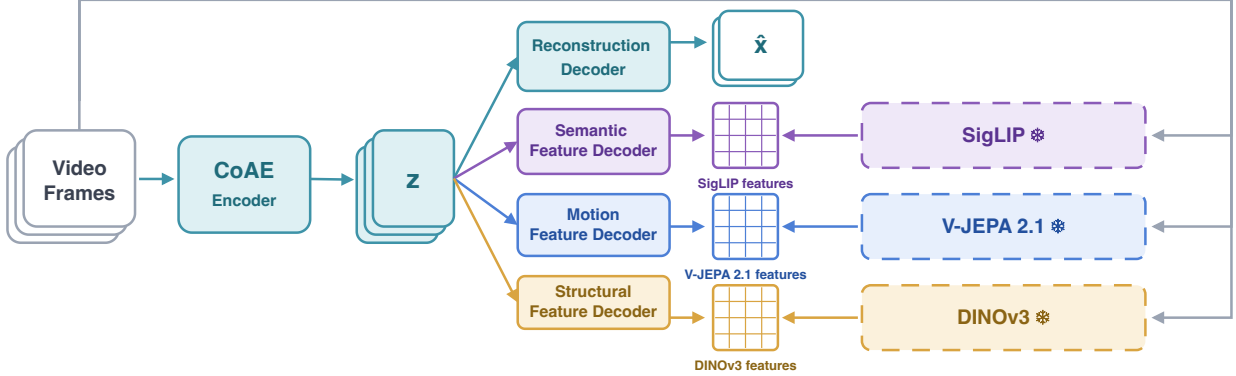


Figure 3: CoAE architecture and training objectives. At the 256×384 frame resolution used by the SVP, the encoder maps one input frame to a 4×6 grid of 512-channel latents, giving 24 tokens per frame. A reconstruction decoder provides pixel-level supervision, while three alignment heads match language-aligned, spatiotemporal, and dense visual features from frozen visual teachers.

3.2.2 Control and Instruction Grounding under Extreme Compression

We evaluate the compressed representation through action recovery and instruction grounding, comparing CoAE with its three visual teachers and DC-AE (Chen et al., 2025a), a deep compression autoencoder with the same $64 \times$ spatial downsampling factor as CoAE.

Action recovery. Action recovery is our primary probe because CoAE is the latent interface through which predicted visual changes inform the IDM. Unlike reconstruction metrics, which assess appearance fidelity, inverse-dynamics probing directly tests whether the compressed latent preserves the visual state changes needed to infer executed robot actions. We freeze each encoder and train the same two-layer inverse-dynamics probe (Section 3.4.1) on a set of 400 samples from a visually cluttered shelf task that requires the robot to categorize and organize many items. Each sample pairs dense and sparse frames with the corresponding recorded actions. We report action mean absolute error (MAE) on 100 held-out samples from the same task. All representations use the same train–test split and optimization protocol. Because every encoder is frozen and evaluated with the same two-layer probe, differences in MAE indicate how directly each representation exposes control-relevant information. As shown in Table 1, DINOv3 and V-JEPA 2.1 obtain the lowest error. CoAE follows at 0.01673, trailing them by only 13–31% while using one-sixteenth as many tokens per frame. In contrast, the errors of DC-AE and SigLIP are 54% and 88% higher than that of CoAE, respectively.

Table 1: Controlled probes of five frozen visual representations. Spatial downsampling is reported as the height $\times$ width factor relative to the input; for example, 64×64 means that one latent location spans a 64×64 -pixel input region. Channels gives the latent feature dimension. Lower is better for both probe metrics.

Representation	Spatial downsampling	Channels	Pixel decoder	Action MAE $\downarrow$	Caption-match error (%) $\downarrow$
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

Instruction grounding. As a complementary probe, instruction grounding tests whether the compact latent preserves how the current scene relates to the commanded behavior. We construct and evaluate the caption-matching probe using 5,000 GenieSim-Instruction episodes. The episodes are distributed uniformly across all ten benchmark tasks and instruction categories, providing balanced coverage of color, number, shape, size, object type, specific-object reference, common-sense reference, logical composition, and object straightening. The complete task list appears in Appendix A.2 and Table 4. 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 3.3) 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 and report classification accuracy under the same protocol for all representations. CoAE achieves 97.95% accuracy, ahead of every teacher and the reconstruction-only baseline. Its 2.05% error rate is 24% lower than the next-best DC-AE and roughly half the 4.15% error of V-JEPA 2.1.

Figure 4 summarizes the central result: our CoAE occupies a favorable compactness–informativeness operating point. On action recovery, it remains within 13–31% of DINOv3 and V-JEPA 2.1 while using one-sixteenth as many tokens per frame. Despite using the same $64 \times$ downsampling along each spatial axis, DC-AE performs worse on both probes, and CoAE has the lowest instruction-matching error of all five, showing that compression rate alone does not explain its result. CoAE and DC-AE are also the only two representations that keep a pixel decoder, so this operating point costs no decodability. We attribute this to manipulation-domain training combined with multi-teacher alignment, which supplies language-aligned semantics, spatiotemporal structure, and dense visual features.

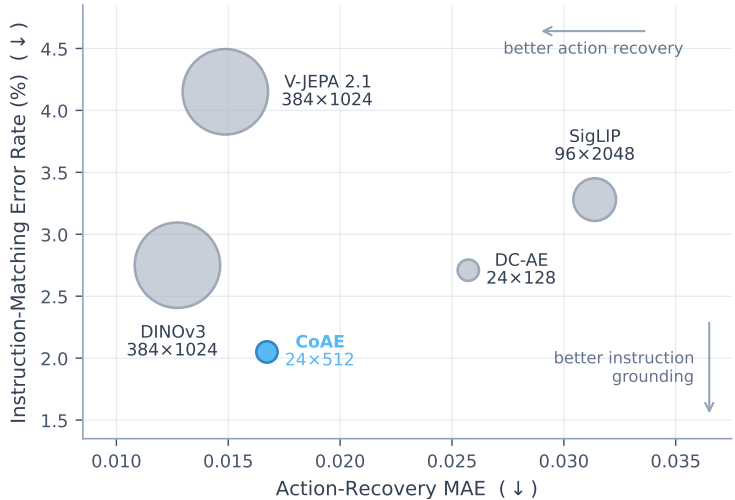


Figure 4: Action-recovery and instruction-grounding probes for five frozen visual representations. Lower is better on both axes. For each 256×384 input frame, labels report tokens per frame $\times$ channel dimension; marker area scales with tokens per frame.

3.3 A Single-Step Visual Planner as a World–Action Interface

The single-step visual planner (SVP) is the visual-generation component of GE-Act 2.0 and provides an explicit future-visual interface to a separately pretrained IDM. It comprises a multimodal understanding module and a conditional flow generator. The flow generator adopts the multi-view diffusion transformer (DiT) pattern of GE-Act 1.0 (HaCohen et al., 2024; Liao et al., 2025), interleaving per-view space–time processing with periodic cross-view attention. Given a current observation and instruction, the multimodal understanding module grounds the instruction in the observed scene, and the flow generator fully denoises the corresponding future visual latents in a single flow-generator pass. The

generated future is then passed to the IDM during co-training and deployment. To our knowledge, our SVP is the first native single-step robot world model pretrained from random initialization on manipulation video.

3.3.1 One-Step Generation Unlocks Underutilized Data

High-quality robot demonstrations remain scarce, making data a key bottleneck to scaling robot learning. Consequently, whether a system can exploit diverse data beyond successful, instruction-annotated trajectories is a central design question. An IDM can learn from robot trajectories with aligned observations and actions, including instruction-free play, failed attempts, and deployment rollouts, without requiring language or success annotations. Such data are difficult to exploit efficiently under the imitation-learning objectives used by VLAs and prior WAMs.

However, using these broader data within a WAM requires more than pretraining the IDM separately: the pretrained IDM must subsequently be connected to the SVP for end-to-end co-training. During separate pretraining, the IDM receives future observations from recorded trajectories. During co-training, the same input must instead come from the visual planner, and the action loss must propagate through the generated future to update both components. With a conventional multi-step video generator, however, a complete future is obtained only after K sequential denoising steps. Propagating action gradients back to the generator therefore requires retaining and differentiating through the entire K -step generation process, substantially increasing computation and memory. Systems such as VPP, DreamZero, and GE-Act 1.0 (Hu et al., 2025; Liao et al., 2025; Ye et al., 2026b) therefore avoid this cost by predicting actions from intermediate states of the generator. Their action components consequently remain tied to the visual generator and do not provide a separate training path for the broader interaction data described above.

Our single-step visual planner resolves this limitation by producing a complete future in one differentiable generator pass. The IDM can therefore first be pretrained on recorded robot trajectories and then connected to the SVP for end-to-end co-training. During co-training, the IDM predicts actions from futures generated by the SVP, and the action loss updates both the IDM and the SVP. At deployment, the IDM continues to predict actions from futures generated by the SVP.

3.3.2 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 $f_0 + T$,

$$\mathcal{F} = \{f_0\} \cup \underbrace{\left\{f_0 + j \frac{H}{N_d}\right\}_{j=1}^{N_d}}_{\text{dense: control}} \cup \underbrace{\left\{f_0 + H + \left\lfloor k \frac{T-H}{N_s} \right\rfloor\right\}_{k=1}^{N_s}}_{\text{sparse: intent}}. \quad (3)$$

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 B.

3.3.3 Toward Unified Understanding and Generation

Multimodal modeling is increasingly moving toward systems that connect understanding and generation instead of treating them as isolated capabilities (Wu et al., 2024a; Xie et al., 2025). 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,

$$(h_0, h_1, \dots, h_L) = \text{VLM}(o_{\text{head}}, c), \quad h_\ell \in \mathbb{R}^{S \times d}. \quad (4)$$

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,

$$\alpha_\ell = \text{softmax}_\ell(g(\text{Pool}(h_\ell[\mathcal{T}]))) , \quad c_{\text{vlm}} = \text{LN}\left(\sum_{\ell=0}^L \alpha_\ell h_\ell[\mathcal{T}]\right). \quad (5)$$

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.

3.3.4 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

$$u(z_t, r, t) := \frac{1}{t - r} \int_r^t v(z_\tau, \tau) d\tau, \quad (6)$$

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. (6) with respect to t gives

$$u(z_t, r, t) = v(z_t, t) - (t - r) \frac{d}{dt} u(z_t, r, t), \quad (7)$$

where the total derivative is

$$\frac{d}{dt} u_\theta = \partial_z u_\theta \cdot \frac{dz_t}{dt} + \partial_t u_\theta. \quad (8)$$

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

$$V_\theta = u_\theta + (t - r) \text{sg}\left[\frac{du_\theta}{dt}\right], \quad (9)$$

where $\text{sg}[\cdot]$ 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

$$m_i = \begin{cases} 0, & \text{conditioning frame,} \\ 1, & \text{future frame to predict.} \end{cases} \quad (10)$$

Suppressing the view axis, the conditional path is

$$z_t^{(i)} = (1 - m_i)z^{(i)} + m_i((1 - t)z^{(i)} + t\epsilon^{(i)}), \quad v_c^{(i)} = m_i(\epsilon^{(i)} - z^{(i)}). \quad (11)$$

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,

$$\frac{dz_t}{dt} = m \odot v_\theta(z_t, t), \quad (r_i, t_i) = (m_i r, m_i t), \quad (12)$$

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,

$$\mathcal{L}_u = \|m \odot (V_\theta - v_c)\|^2, \quad \mathcal{L}_v = \|m \odot (v_\theta - v_c)\|^2. \quad (13)$$

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

At inference, conditioning slots contain their clean latents and prediction slots contain pure noise. A single forward pass at $(r, t) = (0, 1)$ gives

$$\hat{z}^{(i)} = z_1^{(i)} - m_i u_\theta^{(i)}(z_1, r=0, t=1). \quad (14)$$

Conditioning frames pass through unchanged, while predicted frames are fully denoised in one forward pass. The resulting future visual latents are differentiable with respect to all trainable flow-generator parameters, providing the explicit world–action interface used in Section 3.4.

3.4 World-Action Alignment and KASO

The final part of pretraining first optimizes the IDM, then connects the separately optimized SVP and the IDM for co-training. Joint optimization introduces conflicting video and action supervision, including a video–action mismatch that we call the validity gap (Section 3.4.3). We address it with Knowledge-Aligned Selective Optimization (KASO; Section 3.4.4), which preserves action diversity in the resulting pretrained policy and gives subsequent post-training methods such as reinforcement learning broader coverage of the action space.

3.4.1 Inverse Dynamics Model

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 3.3.2. 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

$$\ell_{\text{FM}}(a; o, s, z) = \mathbb{E}_{t,\varepsilon} \left\| v_\phi(a_t, t; o, s, z) - (\varepsilon - a) \right\|^2. \quad (15)$$

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.

3.4.2 Knowledge Transfer with Conflicting Supervision

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. (14). The generated future and End-to-End Co-training (E2E) objective are

$$\hat{z} = G_\theta(o, c, \xi), \quad \mathcal{L}_{\text{E2E}} = \ell_{\text{FM}}(a; o, s, \hat{z}). \quad (16)$$

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,

$$\mathcal{L}_{\text{E2E+PT}} = \mathcal{L}_{\text{E2E}} + \mathcal{L}_{\text{SVP}} + \mathcal{L}_{\text{IDM}}. \quad (17)$$

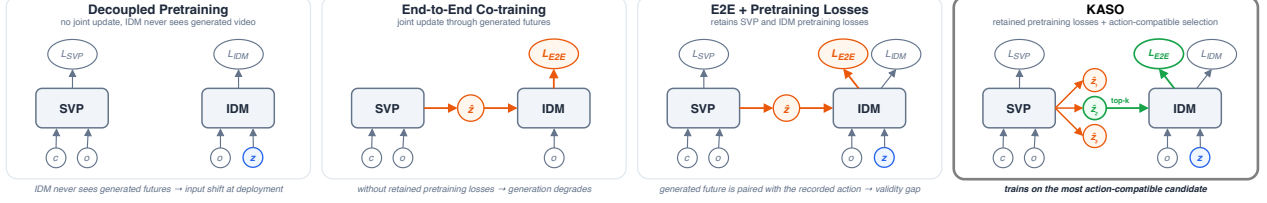


Figure 5: Comparison of Decoupled Pretraining, End-to-End Co-training, E2E + Pretraining Losses, and KASO. Decoupled Pretraining is the pretraining stage itself: the SVP and the IDM are trained on their own objectives and never exchange generated futures. The four strategies differ in whether the IDM 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 5 compare Decoupled Pretraining, E2E, and E2E+PT. E2E+PT retains both pretraining losses, but it does not eliminate conflicting supervision during co-training.

3.4.3 The Validity Gap

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,

$$\begin{aligned} (z, a) &\sim p_{\text{data}}(z, a \mid o, s, c), & \text{recorded in the same rollout,} \\ \hat{z} &\sim p_{\theta}(z \mid o, c), & a \sim p_{\text{data}}(a \mid o, s, c), & \hat{z} \perp a \mid (o, s, c), & \text{paired during co-training.} \end{aligned} \quad (18)$$

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. (18) 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.

A Controlled Toy Two-Stage System. To isolate the consequences of the validity gap, we construct a toy two-stage system (details in Appendix B.7) and train it with each strategy in Figure 5, 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 B.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. (18), 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, described next, 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.

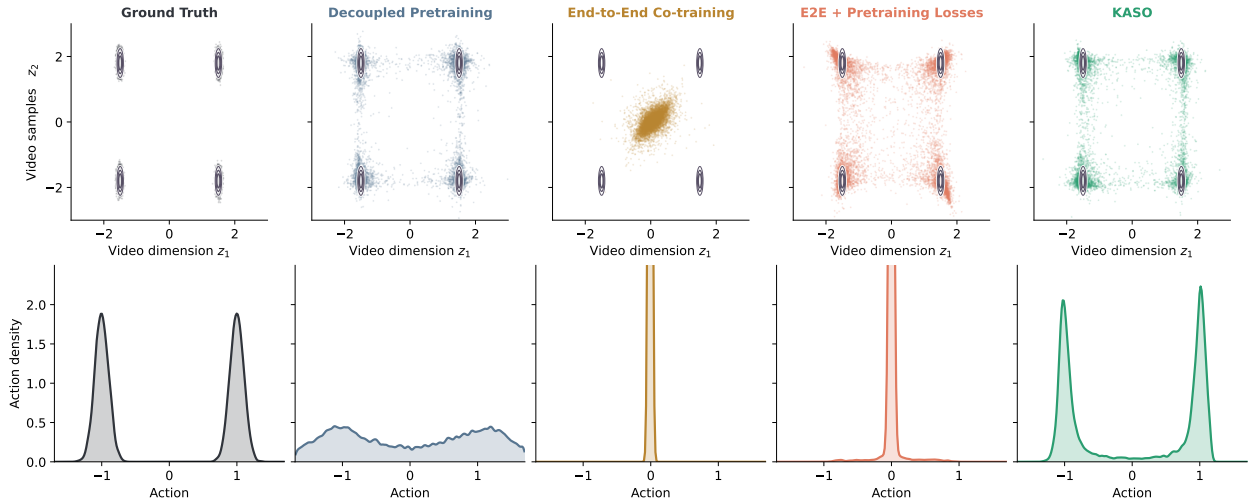


Figure 6: Video samples projected onto the z_1 – z_2 plane (top, with ground-truth contours overlaid) and induced action densities (bottom) learned by each strategy on the toy system. The displayed run uses seed 0 and single-sample evaluation for every learned strategy; seed-to-seed variation is discussed in Appendix B.7. Ground truth contains four video modes and two action modes; in the displayed run, Decoupled Pretraining preserves the four video modes but yields a diffuse action density; End-to-End Co-training collapses both the video and action distributions; with the deterministic IDM, E2E + Pretraining Losses preserves the video modes but collapses the action density to the mean of the two action modes; and KASO recovers the four video modes and the bimodal action structure.

3.4.4 KASO: Knowledge-Aligned Selective Optimization

Therefore, we propose KASO, an online top- k procedure for selecting action-compatible future visual-state sequences during co-training, as illustrated in Figure 5. 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 7 and Algorithm 1). 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. (17), 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

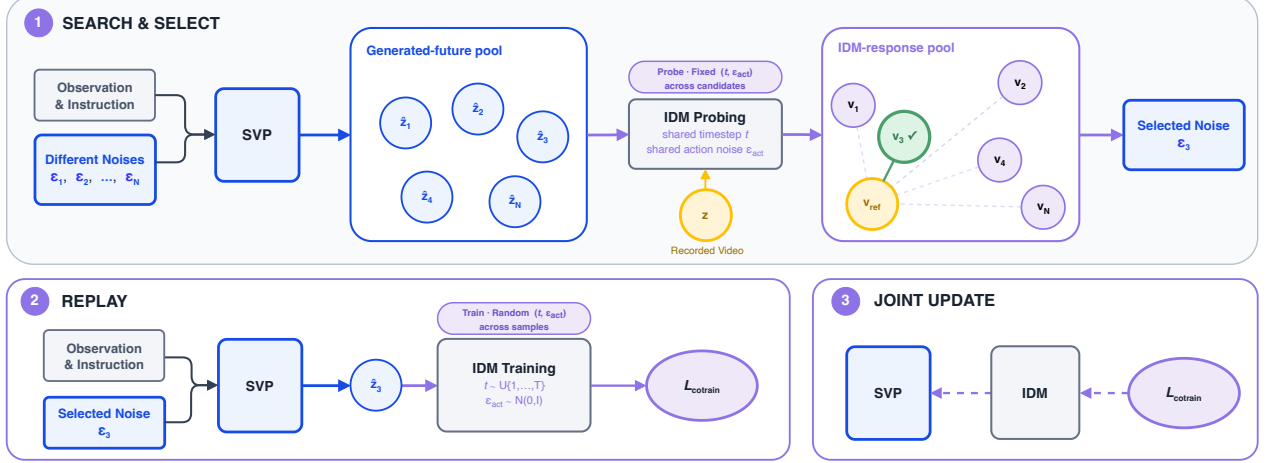


Figure 7: The KASO pipeline. Candidate visual predictions are sampled from the SVP and scored without gradients; selected noises are then replayed exactly so that the joint action loss updates both the IDM and the SVP.

conditioning as the only changing input. For the set $\mathcal{D}$ of dense action tokens, the candidate energy is

$$E_n = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \left\| v_{\phi,i}(\tilde{a}, t_p; o, s, \hat{z}_n) - v_{\phi,i}(\tilde{a}, t_p; o, s, z) \right\|_2^2. \quad (19)$$

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. (19) 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

$$\mathcal{L}_{\text{E2E}}^{\text{sel}} = \frac{1}{k} \sum_{n \in \mathcal{S}_k} \ell_{\text{FM}}(a; o, s, \hat{z}_n). \quad (20)$$

The full KASO objective retains the same two pretraining losses:

$$\mathcal{L}_{\text{KASO}} = \mathcal{L}_{\text{E2E}}^{\text{sel}} + \mathcal{L}_{\text{SVP}} + \mathcal{L}_{\text{IDM}}. \quad (21)$$

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. (14). The update also includes $\mathcal{L}_{\text{SVP}}$ on the recorded video and $\mathcal{L}_{\text{IDM}}$ 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 B.

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

1: Sample video flow times and noise; compute $\mathcal{L}_{\text{SVP}} = \mathcal{L}_u + \mathcal{L}_v$ Eq. (13)

2: $\mathcal{L}_{\text{IDM}} \leftarrow \ell_{\text{FM}}(a; o, s, z)$ Eq. (15)

// Candidate generation without gradients

3: **for** $n = 1, \dots, N$ **do**

4: Draw and retain generation noise ξ_n

5: $\hat{z}_n \leftarrow$ one-step generation from (o, c, ξ_n) Eq. (14)

6: **end for**

// Compatibility probe with the active IDM, no gradients

7: Draw shared probe noise ε ; $\tilde{a} \leftarrow t_p \varepsilon + (1 - t_p)a$

8: $v_{\text{ref}} \leftarrow v_\phi(\tilde{a}, t_p; o, s, z)$

9: **for** $n = 1, \dots, N$ **do**

10: $E_n \leftarrow \|v_\phi(\tilde{a}, t_p; o, s, \hat{z}_n) - v_{\text{ref}}\|_{\text{dense}}^2$ Eq. (19)

11: **end for**

// Top-k selection and differentiable replay

12: $\mathcal{S}_k \leftarrow$ indices of the k smallest energies in $\{E_n\}_{n=1}^N$

13: **for** $n \in \mathcal{S}_k$ **do**

14: $\hat{z}_n \leftarrow$ replay ξ_n with gradients

15: **end for**

// Joint update

16: $\mathcal{L}_{\text{E2E}}^{\text{sel}} \leftarrow \frac{1}{k} \sum_{n \in \mathcal{S}_k} \ell_{\text{FM}}(a; o, s, \hat{z}_n)$ Eq. (20)

17: $\mathcal{L}_{\text{KASO}} \leftarrow \lambda_{\text{E2E}} \mathcal{L}_{\text{E2E}}^{\text{sel}} + \lambda_{\text{SVP}} \mathcal{L}_{\text{SVP}} + \lambda_{\text{IDM}} \mathcal{L}_{\text{IDM}}$ Eq. (21)

18: $\theta, \phi \leftarrow \theta, \phi - \alpha \nabla_{\theta, \phi} \mathcal{L}_{\text{KASO}}$

3.4.5 Controlled Real-Robot Ablation

To isolate the effect of the alignment objective, we compare E2E, E2E+PT, and KASO under a deliberately reduced, matched alignment-stage and evaluation setup. Every variant initializes from the same full-scale components: the SVP, pretrained on 39,000 hours of instruction–video data, and the IDM, pretrained on 32,000 hours of action-labeled trajectories. Only the connection/alignment stage is run on a 300-hour co-training mixture, including 30 hours from G2-90D, the embodiment used for evaluation. Thus, this is a 300-hour alignment stage atop full-scale pretrained components, not a model trained on 300 hours in total. The exact evaluation environment and physical object instances are unseen during training, and no evaluation-specific SFT is performed. We evaluate two real-robot picking protocols. In the single-object protocol, the same small block is placed at different table positions for 25 trials. In the four-object protocol, a sponge, red apple, computer mouse, and remote control are presented simultaneously, and the robot is instructed to pick each designated target in 10 trials.

We report two complementary metrics. In the four-object protocol, the follow score counts trials in which the robot contacts the instructed target and no distractor object, regardless of whether the subsequent grasp and lift succeed. Pick success additionally requires completing the pickup. Because the single-object protocol contains no distractor, follow score is not defined for that protocol.

Table 2: Controlled real-robot ablation of retaining pretraining losses and action-compatible selection. Follow score requires contacting the instructed target and no distractor; pick success additionally requires completing the pickup. Four-object macro averages assign equal weight to the four targets. All variants initialize from the same full-scale pretrained components (SVP, 39,000 hours; IDM, 32,000 hours) and use the same 300-hour connection/alignment mixture, including 30 hours from G2-90D, without evaluation-specific SFT. Entries are percentages.

<i>(a) Follow score: four-object scene (10 trials per target)</i>						
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
<i>(b) Pick success</i>						
Method	Single-object scene (25 trials)	Four-object scene (10 trials per target)				
	Small block	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

Table 2 separates target following from completed pickup. In the four-object protocol, E2E+PT and KASO achieve the same 95% follow score, but KASO converts more correct target contacts into successful picks: its macro-average pick success is 37.5%, compared with 22.5% for E2E+PT (+15 percentage points), and it improves pick success for every tested target. In the single-object protocol, KASO raises pick success from 12% to 40% (+28 points). Relative to E2E, KASO improves the four-object follow score from 87.5% to 95%, the four-object macro-average pick success from 27.5% to 37.5%, and the single-object pick success from 12% to 40%.

Table 2 provides a controlled demonstration of KASO in a simple, matched setting. By holding the full-scale pretrained initialization, 300-hour alignment mixture, embodiment data, and evaluation protocol fixed, the comparison isolates the effect of the alignment objective. Accordingly, this table compares 300-hour alignment stages atop full-scale pretrained components rather than models trained from scratch on 300 hours. This experiment is separate from the full-scale capability evaluation in Section 5: the full model is trained with substantially more data and exhibits stronger capability, so the values in Table 2 should not be interpreted as its final performance.

4 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 3.4).

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 5.2. The three stages contain 39,000, 32,000, and 30,000 hours, respectively.

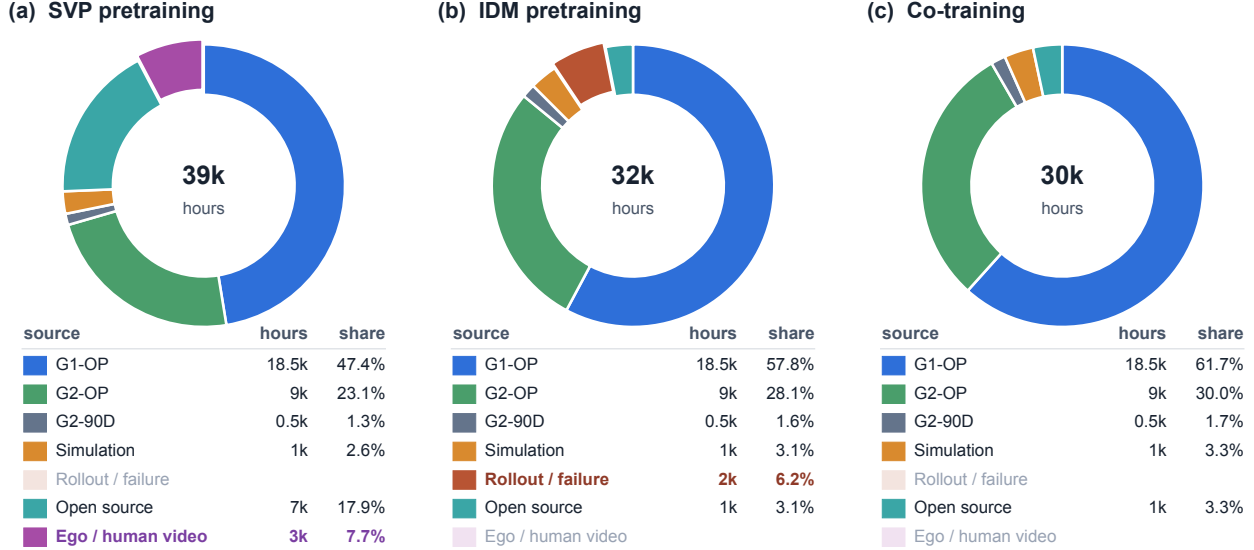


Figure 8: Stage-specific training-data mixtures. The panels show the 39,000-hour SVP pretraining mixture, the 32,000-hour IDM pretraining mixture, and the 30,000-hour co-training mixture. Slice angle denotes the share of data hours; colors and source order are held fixed across panels. Violet and coral 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.

5 Evaluating Zero-Shot OOD Manipulation at Scale

Our central evaluation resolves manipulation capability into 100 atomic tasks, revealing which individual skills emerge, improve, or remain difficult as training scales. We track this broad repertoire across data scale on both G1-OP and G2-90D, with G1-OP as the dominant embodiment in the co-training mixture and G2-90D as a data-scarce embodiment contributing less than 2% of it. We then complement the atomic-skill evaluation with two controlled complex-scene studies: a quantitative decomposition of grounding across object identity, color, size, shape, position, and order, and qualitative stress tests in which the current instruction conflicts with an ongoing behavior or a conventional scene relation. The section concludes with three standardized simulation benchmarks that test OOD conditions after each benchmark’s prescribed in-distribution adaptation; we interpret these results separately from the direct real-robot zero-shot evaluation. Because GE-Act 2.0 is designed as a low-level manipulation policy rather than a high-level planner, these experiments focus on executable visuomotor competence and fine-grained grounding rather than high-level planning or open-ended reasoning.

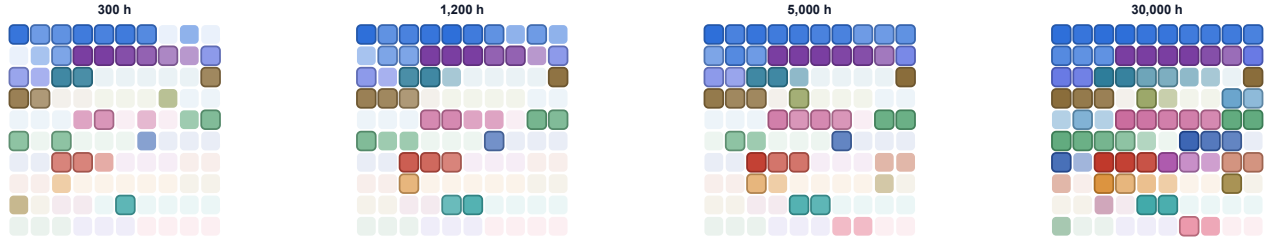
5.1 Zero-Shot OOD Evaluation without Per-Task SFT

Existing world-action and vision–language–action evaluations often include a task-specific adaptation stage (Black et al., 2024; Cheang et al., 2024; Kim et al., 2024b; Li et al., 2026; Wu et al., 2024b; Ye et al., 2026a; Zhang et al., 2026; Zhou et al., 2026). Such evaluations measure the resulting specialist policy and therefore answer a different question. In contrast, all our main real-robot experiments evaluate pretrained checkpoints directly, without per-task SFT or any subsequent adaptation. This protocol isolates capabilities already present in the pretrained policy rather than capabilities introduced through demonstrations of the evaluation task.

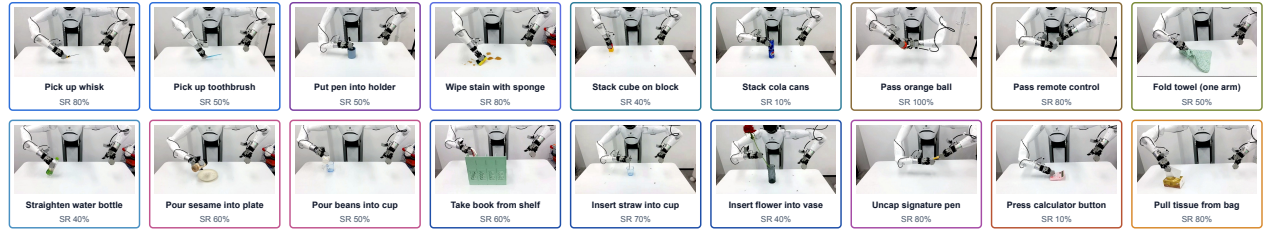
We use *zero-shot* to mean that a checkpoint receives no evaluation-task fine-tuning, demonstrations, or per-task checkpoint selection. The evaluation is *OOD*: the test scenes, backgrounds, lighting configurations, and object instances are excluded from pretraining and co-training. At each evaluation point, one checkpoint is deployed across the complete suite with fixed deployment settings; full protocol and checkpoint details are provided in Appendix B.

Zero-Shot OOD Atomic-Skill Success Rate

100 atomic tasks · 20 skill groups · four training-data scales



Representative Zero-Shot OOD Rollouts · 30,000 h Model



	300 h	1,200 h	5,000 h	30,000 h		300 h	1,200 h	5,000 h	30,000 h
Pick up red apple	90	100	100	100	Push block to left	30	40	0	70
Pick up water bottle	40	50	70	90	Push cup to center	0	20	30	60
Pick up water bottle (2)	50	70	90	100	Push kettle forward	30	20	20	50
Pick up towel	80	90	80	100	Push book to edge	0	0	0	30
Pick up stapler	70	100	80	90	Roll the bottle	0	0	0	10
Pick up red cube	80	80	70	90	Push tissue pack in	0	0	0	0
Pick up whisk	60	50	70	60	Insert straw into cup	20	30	40	70
Pick up toothpaste	0	20	40	80	Insert bread into machine	0	0	0	50
Pick up water cup	20	40	40	80	Insert flower into vase	0	0	0	40
Pick towel from rod	0	20	50	70	Insert plug into strip	0	0	0	0
Pick pen from water cup	0	10	30	60	Take book from shelf	0	0	0	60
Pick up controller	10	30	60	50	Put book into shelf	0	0	0	10
Pick up toothbrush	30	30	40	50	Close kettle lid	30	60	90	100
Put apple into container	100	100	100	100	Close blue box lid	30	50	40	90
Put blue cube into plate	100	100	100	100	Close tissue box lid	10	30	40	70
Put orange into bowl	80	100	100	100	Uncap signature pen	0	0	0	80
Put pen into drawer	60	60	90	100	Unscrew bottle cap	0	0	0	30
Put cube into paper cup	30	60	80	80	Separate paper cups	0	0	0	20
Put pen into holder	20	20	40	50	Press bread machine button	0	0	10	30
Wipe table with eraser	40	40	60	60	Press stapler to staple	0	0	10	30
Wipe stain with sponge	30	40	40	80	Press calculator button	0	0	0	10
Wipe plate with rag	20	20	30	70	Press lamp switch	0	0	0	0
Stack coasters	80	70	80	100	Pull tissue from bag	10	30	30	80
Stack bowls	70	80	80	90	Pull tray to front	0	0	10	30
Stack cube on block	0	10	20	40	Pull off pen cap	0	0	0	20
Stack paper cups	0	0	0	30	Open the drawer	0	0	0	10
Cover trash can lid	0	0	0	20	Pull cable from box	0	0	0	0
Stack cola cans	0	0	0	10	Unplug charger from strip	0	0	0	0
Cover cutlery box lid	0	0	0	0	Take towel off rod	0	0	10	60
Pass sponge eraser	60	90	100	100	Hang hat on rack	0	0	0	0
Pass orange ball	70	70	80	100	Hang rag on box	20	0	0	0
Pass remote control	40	60	60	80	Hang towel on rod	0	0	0	0
Pass cola can	0	40	60	70	Tear open envelope	0	0	0	10
Pass Rubik's cube	0	0	0	0	Tear paper towel off roll	0	0	0	0
Fold towel (one arm)	0	0	40	90	Close the open book	0	50	70	90
Fold towel (two arms)	0	0	0	10	Flip coaster to front	60	70	60	80
Fold rag twice	0	0	0	0	Flip book to front	0	0	0	0
Unfold folded towel	20	0	0	0	Flip plate over	0	0	0	0
Straighten water cup	0	0	0	60	Flip tissue pack over	0	0	0	0
Straighten water cup (2)	0	0	0	30	Open the book	0	0	0	0
Straighten water bottle	0	0	0	10	Sweep cubes into dustpan	0	0	0	10
Straighten water bottle (2)	0	0	0	40	Sweep cubes by hand	0	0	0	0
Straighten soda drink	0	0	0	10	Sweep cubes with brush	0	0	0	0
Pour fruits into plate	20	40	50	70	Unzip file bag	0	0	0	0
Pour sesame into plate	30	40	30	60	Unzip pencil case	0	0	0	0
Pour beans into cup	0	20	40	50	Zip up file bag	0	0	0	0
Pour sugar into cup	10	20	30	50	Shake bottle to check	0	0	10	30
Pour cubes into bowl	0	0	0	40	Shake glass bottle	0	0	10	20
Close the drawer	20	50	60	70	Stir coffee with chopstick	0	0	0	0
Push box to left	40	40	60	70	Stir water with spoon	0	0	0	0

Figure 9: 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.

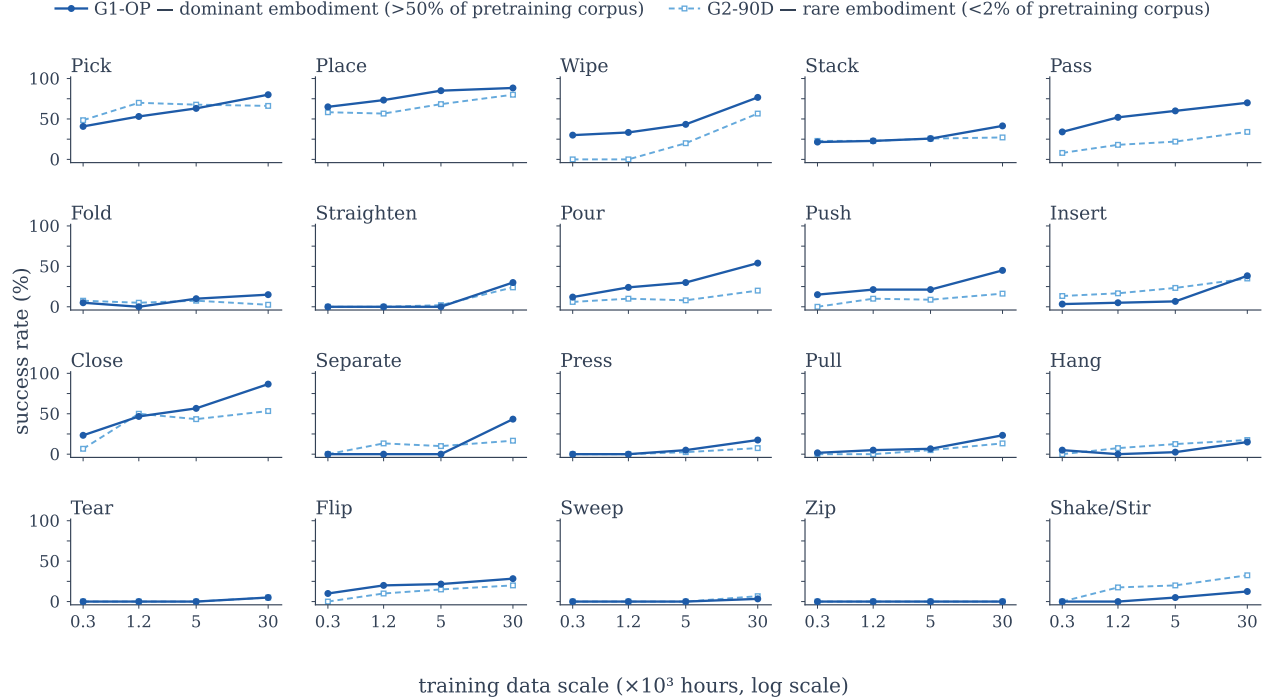


Figure 10: Skill-group zero-shot OOD scaling trajectories for G1-OP (more than 50% of the co-training mixture) and G2-90D (less than 2%). Each panel reports one group’s mean SR across the 300-, 1,200-, 5,000-, and 30,000-hour models.

5.2 Scaling Manipulation Skills

Atomic skills provide the most direct measure of an executable low-level manipulation policy. They minimize confounding from high-level planning and long-horizon error accumulation, allowing the evaluation to focus on whether the policy can perceive a local manipulation problem and execute the corresponding action.

Our real-robot suite contains 100 tasks across 20 skill groups, spanning pick, place, push, stack, tool use, articulated-object manipulation, insertion, deformable-object manipulation, and fine motor control. This breadth is intended to measure a repertoire rather than a small set of canonical or headline tasks. Every task has a pre-specified terminal success criterion, and the same suite is evaluated on G1-OP and G2-90D. Each task is evaluated over 10 trials for every combination of embodiment and training-data scale under a standardized reset procedure. We report empirical success rate (SR) at the task, skill-group, and suite levels; the complete taxonomy, reset protocol, and raw results are provided in Appendix C.

5.2.1 Scaling with Training Data Volume

A central question for large-scale robot pretraining is whether expanding the training corpus produces a broader and more reliable repertoire of zero-shot OOD manipulation skills. We train separate models with nested co-training pools containing 300, 1,200, 5,000, and 30,000 hours. The smaller pools are contained in the larger ones and preserve the overall mixture composition as closely as possible. All models start from the same pretrained components, are co-trained with KASO (Section 3.4.4), and use the same training recipe; each is trained until completing one epoch or reaching the maximum duration permitted by our compute budget, whichever comes first. This is therefore a practical end-to-end scaling comparison rather than a matched-compute isolation of training-data volume.

We report the results at three levels: task, skill group, and suite. Figure 9 shows every task as a wall for G1-OP, grouped into the 20 skill groups, with columns denoting the four training-data scales. Figure 10 summarizes the same evaluation as paired G1-OP and G2-90D trajectories for all 20 skill groups. Suite-level SR for both embodiments is reported below.

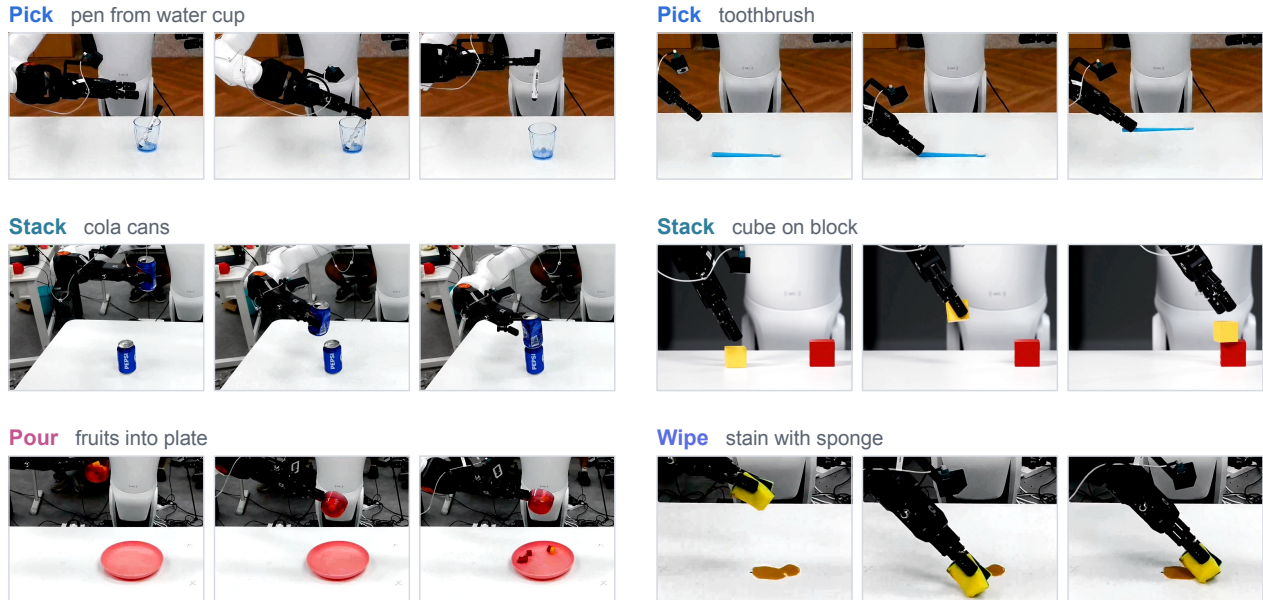


Figure 11: Representative successful zero-shot OOD rollouts on G2-90D across skill groups, each shown as three frames spanning the start, mid-execution, and outcome of the trial. Skill labels are colored by skill group, matching the palette of Figure 9.

G1-OP suite-level SR increases from 17.1% with 300 hours to 22.6% with 1,200 hours, 27.3% with 5,000 hours, and 44.1% with 30,000 hours. G2-90D follows the same upward trend, increasing from 13.4% to 21.0%, 23.5%, and 31.1% across the four data scales. These trajectories yield total gains of 27.0 and 17.7 percentage points for G1-OP and G2-90D, respectively. Both curves continue to rise substantially from 5,000 to 30,000 hours, providing no evidence of saturation within the measured data range.

The improvement is distributed broadly across the repertoire: 19 of 20 G1-OP skill groups and 18 of 20 G2-90D groups perform better at 30,000 hours than at 300 hours; the exceptions are Zip, with no success in any of its 240 trials across both embodiments and all four scales, and Fold on G2-90D, whose apparent decline spans three successful trials at 300 hours and one at 30,000. Zip is both mechanically demanding and nearly absent from the training data: 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, and the group receives only 16.4 hours across the whole corpus, the second-lowest coverage of any skill group. Section 5.2.2 takes up this relationship between coverage and capability across the full taxonomy. At the atomic-task level, the number of tasks with nonzero empirical SR rises from 39 to 76 for G1-OP and from 24 to 72 for G2-90D. Across the four scales, 87 of 100 G1-OP task trajectories and 78 of 100 G2-90D trajectories are non-decreasing. Excluding the mechanically trivial Close group, the largest G1-OP gains include Wipe (+46.7 points), Separate (+43.3), Pour (+42.0), and Pick (+39.2). Scaling therefore improves most of the measured repertoire, although the trajectories are not uniformly monotonic, particularly for G2-90D.

Scaling on the data-scarce embodiment. The 17.7-point gain on G2-90D is especially notable because it contributes less than 2% of the co-training mixture. Within the skill-attributed G2-90D training data at the largest scale, 10 of the 20 skill groups contain fewer than five hours, including six with fewer than two hours and four with fewer than one hour. Nevertheless, five of the six groups with fewer than two hours improve from zero SR with the 300-hour model to nonzero SR with the 30,000-hour model. Flip and Separate reach 20.0% and 16.7% SR with only 1.2 and 1.25 hours of G2-90D-specific data, respectively, while Pass reaches 34.0% with 3.9 hours. One hardware-sensitive exception is Shake/Stir, which 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. This hardware effect does not explain the within-G2-90D scaling trend, which compares models on the same embodiment across data scales. Together with the aggregate trend in Figure 10, these results suggest that a sparsely

represented embodiment can acquire substantially broader capability as the shared multi-embodiment training corpus scales. Figure 11 shows representative successful G2-90D rollouts across skill groups.

5.2.2 Scaling with Skill-Specific Data Volume

Figure 10 reveals several skill-level outcomes that are difficult to explain through apparent motor difficulty alone.

- **Demanding skills can emerge.** Stack requires precise placement and stability, Pass requires coordinated dual-arm transfer, and Straighten requires multi-stage reorientation with an angle-sensitive final pose; nevertheless, all three acquire meaningful capability as training scales.
- **Similar motions, divergent outcomes.** Wipe and Sweep involve closely related tool-mediated surface motions but achieve substantially different SR.

These contrasts motivate examining whether the uneven capability distribution is related to skill-specific training coverage.

Rather than relying on selected examples, we audit training coverage across the complete 20-group atomic-skill taxonomy. We process the full multi-embodiment corpus using its step-captioned instruction–action segments, convert their annotated frame spans to hours, and map them to the same action-centric taxonomy as the evaluation. We exclude Close from the quantitative relationship because its contact-based success criterion makes it mechanically easier than the other groups. For each of the remaining 19 groups, we pair its summed training hours with the SR of the G1-OP 30,000-hour model and analyze the relationship using log hours and logit SR.

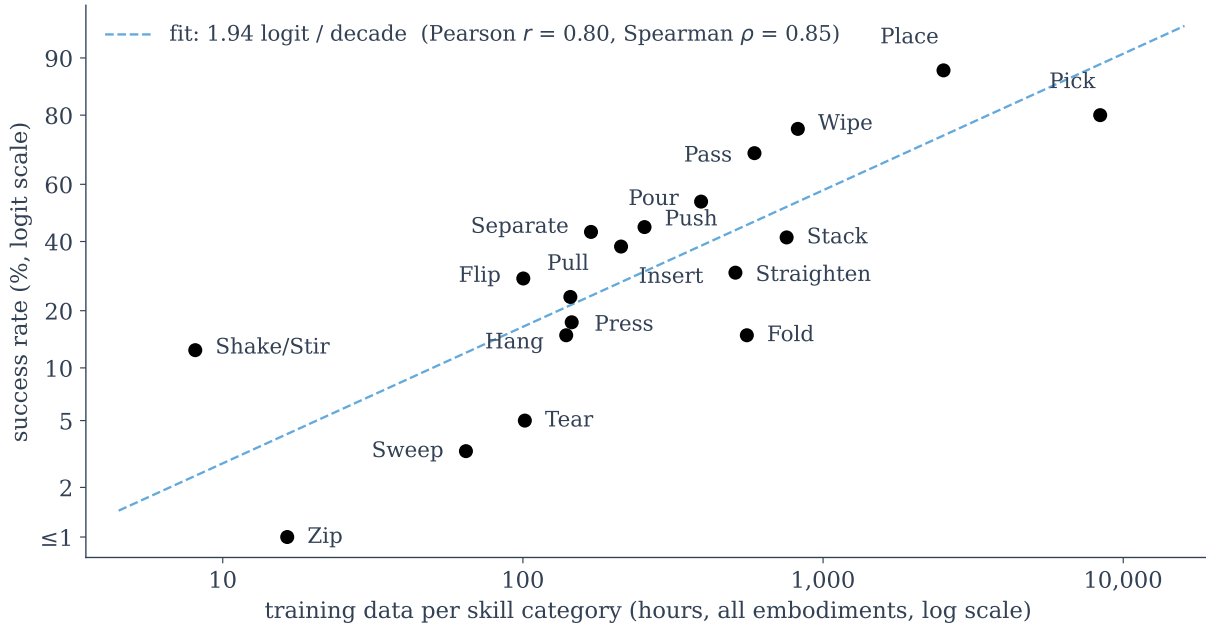


Figure 12: Skill-group training coverage and G1-OP zero-shot OOD success for the 30,000-hour model. Each point is one of the 19 groups retained after omitting the mechanically trivial Close group; the horizontal axis reports the corresponding training hours across all embodiments on a logarithmic scale, and the vertical axis reports SR on a logit scale. Coverage and success are strongly correlated (Pearson $r = 0.80$; Spearman $\rho = 0.85$), with a fitted gain of 1.94 logit units per decade of data.

Across the 19 nontrivial skill groups, training coverage is strongly associated with zero-shot OOD success (Pearson $r = 0.80$; Spearman $\rho = 0.85$). The fitted relationship corresponds to an increase of 1.94 logit units in SR for every tenfold increase in skill-specific training hours. The qualitative irregularities visible in Figure 10 therefore form a systematic distributional pattern rather than a collection of isolated cases.

The fitted relationship provides a consistent explanation for the qualitative observations above.

- Despite their demanding execution, skills with substantial coverage achieve nontrivial performance: Pass receives 590.8 hours and improves from 34.0% to 52.0%, 60.0%, and finally 70.0%; Straighten receives 510.5 hours and remains at 0.0% through 5,000 hours before reaching 30.0% at 30,000 hours; and Stack receives 756.2 hours and rises from 21.4% to 41.4%.
- Conversely, similar motions can lead to divergent outcomes: Wipe receives 824.1 hours and reaches 76.7% SR, whereas Sweep receives 64.6 hours and reaches only 3.3%.

Although intrinsic difficulty may explain part of the cross-skill variation, the within-skill scaling trajectories in Figure 10 and the corpus-wide relationship in Figure 12 make a difficulty-only explanation insufficient. Taken together, these results provide converging evidence that increasing both the scale and skill diversity of robot training data—particularly the coverage of long-tail behaviors—should improve pretrained zero-shot OOD capability across a broader range of manipulation settings.

5.3 Instruction Following in Complex Scenes

Instruction following requires both selecting the correct behavior among several visually plausible alternatives and remaining responsive when the current instruction conflicts with an ongoing action or a conventional scene relation. We evaluate these complementary aspects in complex tabletop scenes using the final checkpoint of the full KASO co-training run on G1-OP, without further adaptation. Because GE-Act 2.0 remains a low-level policy, these experiments test instruction-dependent control over familiar executable behaviors rather than high-level planning or open-ended reasoning.

5.3.1 Fine-Grained Grounding across Instruction Dimensions

To minimize confounding from familiarity with the underlying motor skill, execution is restricted to *pick* and *place*, the two most frequent and reliable primitives in the robot data. A failure can therefore be interpreted with less ambiguity than on an unfamiliar or long-horizon skill.

Within each primitive, we decompose instruction grounding into six dimensions: object identity, color, size, shape, position, and order. Size jointly evaluates largest, smallest, medium, and related comparative expressions; position covers direct left/right and near/far relations, whereas order covers middle and ordinal references such as “second from the left.” The suite contains 33 Pick instructions and 26 Place instructions, each evaluated over five trials, for a total of 295 real-robot rollouts. Each dimension uses a controlled scene in which multiple actions are visually feasible while only the instruction-specified target or destination is correct. Figure 13 shows the paired Pick and Place setups for all five physical scene families; the complete instruction list is provided in Appendix C.3.

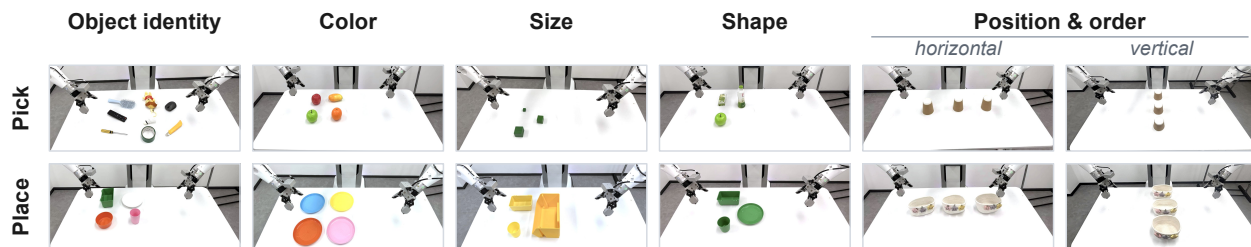


Figure 13: Controlled scenes for fine-grained instruction following, ordered as introduced in the text: object identity, color, size, shape, and position & order. Position and order share one setup, evaluated in a horizontal layout (left/right and ordinal references) and a vertical layout (near/far). Top row: Pick scenes; bottom row: Place scenes.

We report two complementary metrics: *Follow Score*, which measures whether the policy physically reaches only the instruction-specified referent, and SR, which requires the full Pick or Place behavior to be completed. For Pick, a trial satisfies Follow Score when the gripper contacts the specified object without contacting another object, even if it subsequently fails to lift the target. For Place, Follow Score is satisfied when the carried object contacts the specified target container and no other container, even if placement is not completed. The difference between Follow Score and

SR therefore separates incorrect instruction grounding from physical execution failure after the correct referent has been reached. Figure 14 reports both metrics for Pick and Place over the six grounding dimensions and places them beside the corresponding occurrence rates in natural referring expressions and the GE-Act 2.0 training corpus; instruction-level results are deferred to the appendix.

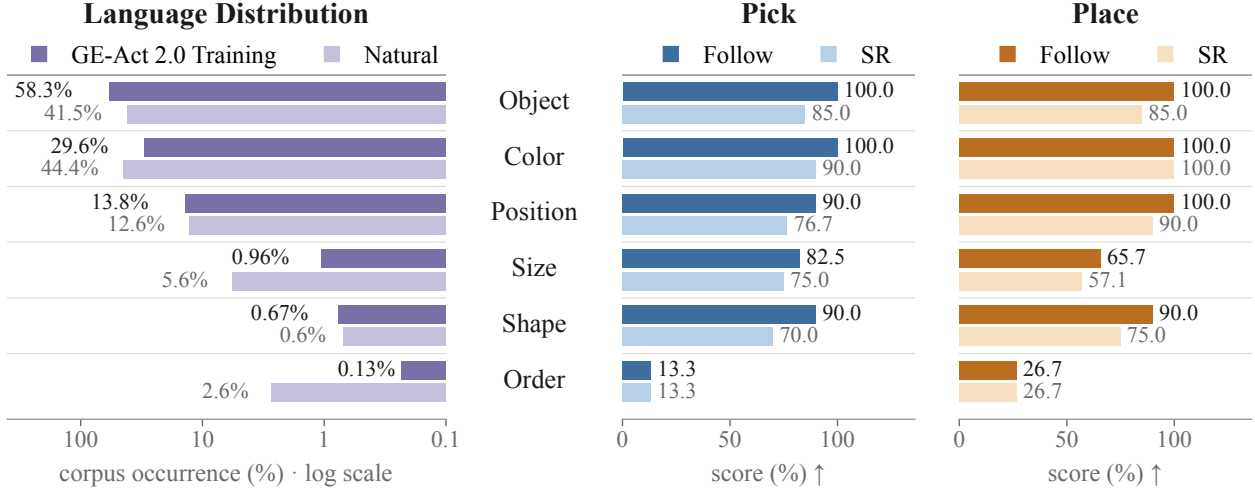


Figure 14: Fine-grained instruction following and language distribution across six grounding dimensions. Left: occurrence in 95,010 human-written RefCOCOg expressions (Natural) and the GE-Act 2.0 training corpus, shown on a logarithmic scale. Middle and right: Follow Score and full-task SR for Pick and Place, respectively. Dimensions are ordered by their occurrence in the GE training corpus; descriptor categories are counted independently and may overlap.

GE-Act 2.0 reliably grounds object identity, color, direct position, and shape, while size is less consistent and order remains the primary weakness. Follow Score is at least 90.0% for both Pick and Place across those four dimensions, but decreases to 82.5%/65.7% for size and 13.3%/26.7% for order. Across all 295 rollouts, the overall Follow Score is 83.1%, compared with a full-task SR of 72.9%, showing that some failures occur after the policy has already reached the correct referent.

This capability profile broadly mirrors the long-tailed distribution of referring expressions: in both 95,010 human-written RefCOCOg expressions (Mao et al., 2016) and the GE-Act 2.0 training corpus, object and color form the head, position is intermediate, and size, shape, and order are comparatively rare. The tail is especially pronounced in GE training, where size, shape, and order occur in only 0.96%, 0.67%, and 0.13% of instructions, respectively. Order is therefore both the least represented and weakest dimension, while size shows a softer version of the same pattern. Shape is an informative exception, achieving 90.0% Follow Score for both primitives despite its low occurrence, showing that frequency is not the only determinant of grounding difficulty. Because frequency is not independently controlled and the dimensions differ in perceptual and relational complexity, this experiment establishes association rather than causation. Nevertheless, the combined distribution and performance profile suggests that increasing the diversity and volume of rare referring expressions—especially ordered and comparative relations—is a promising route to stronger pretrained zero-shot grounding; matching rules and complete counts are provided in Appendix C.3.

5.3.2 Instruction Following under Behavioral and Semantic Conflicts

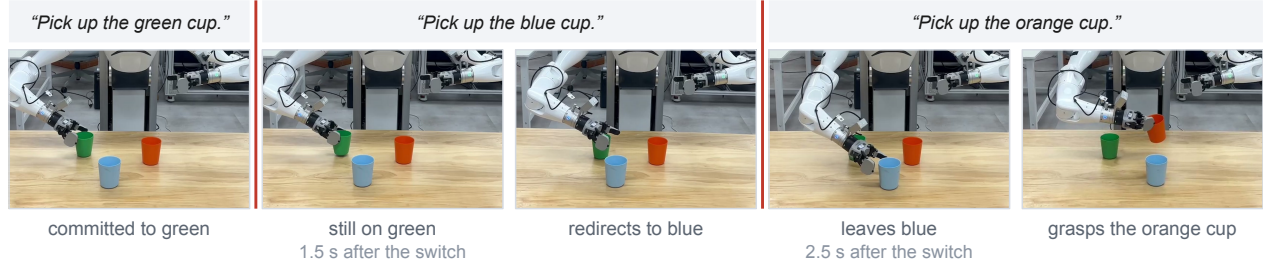
We next examine two harder forms of instruction conflict: behavioral conflict, where a revised command opposes a robot–scene state already committed toward the previous behavior, and semantic conflict, where the requested relation contradicts a conventional scene association. In preliminary experiments 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.

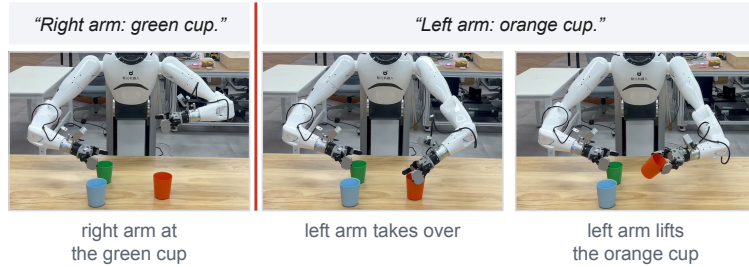
- *Target switch.* The commanded object changes from A to B, and later from B to C, each time when the end effector is already close to, or beginning to grasp, the previous target (Figure 15a). Immediately after each switch, the robot retains brief residual motion toward the previous target, but subsequently 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 15b). 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 15c). The policy completes the unusual placement, following the explicit object–destination relation rather than defaulting to the more conventional scene association.

(a) **Target switch** *behavioral conflict*



(b) **Arm switch** *behavioral conflict*



(c) **Cup into the shoebox** *semantic conflict*



Figure 15: 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.

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

5.4 OOD Performance on Simulation Benchmarks

Real-robot evaluation is our primary measure of a pretrained policy’s zero-shot capability, but matching hardware, scenes, and evaluation conditions across independently developed models is difficult. We therefore complement the main real-robot experiments with three standardized simulation benchmarks spanning environment randomization, instruction following, and controlled visual and embodiment perturbations. These benchmarks cannot directly establish out-of-the-box real-world generalization: the simulator-to-real gap is substantial, and each benchmark includes an in-distribution training or adaptation stage. Because GE-Act 2.0 is pretrained mostly on real-world data, we first SFT it on each benchmark’s prescribed in-distribution training condition, putting it on the same footing as models adapted specifically to that simulator, and then evaluate only held-out OOD simulation conditions. We deliberately do not develop a benchmark-specific post-training strategy.

RoboTwin Clean-to-Random (Chen et al., 2025b) trains on Easy/Clean and tests background, lighting, clutter, table-height, and combined-Hard shifts; GenieSim-Instruction (Yin et al., 2026) trains on its released instruction-following data and tests unseen scene configurations; and LIBERO-Plus (Fei et al., 2025) adapts on standard LIBERO and evaluates camera, robot-state, language, lighting, background, sensor-noise, and layout perturbations. Despite this deliberately simple adaptation, GE-Act 2.0 is the strongest compared model on RoboTwin and GenieSim-Instruction, and remains competitive on LIBERO-Plus. Full protocols, comparisons, and per-axis results are provided in Appendix A.

6 Conclusion

We introduced GE-Act 2.0, a world–action model designed to study the pretraining and scaling of visual generation and action prediction as a unified system. GE-Act 2.0 combines a control-oriented autoencoder that preserves control-relevant information, a single-step visual planner that enables decoupled pretraining, and KASO, which aligns generated futures with compatible action supervision during co-training.

Across zero-shot OOD evaluations on 100 real-robot tasks and two embodiments, increasing manipulation data produces broad improvements across nearly all evaluated skill groups. Performance also improves on the sparsely represented embodiment, suggesting useful transfer from the broader training corpus, while skill-specific data coverage strongly predicts downstream success. Complementary experiments demonstrate fine-grained instruction grounding and adherence to explicit commands under challenging behavioral and semantic conflicts. Together, these findings establish GE-Act 2.0 as a scalable approach to learning transferable manipulation capabilities through world–action pretraining.

7 Limitations and Outlook

GE-Act 2.0 opens a direct route for learning from egocentric manipulation video, but the present study has not yet explored this source at the scale its availability permits. We view egocentric video as a particularly promising source of diverse physical interactions and task semantics, and preliminary signals from our training mixtures are encouraging. A key direction for future work is therefore to scale egocentric-video pretraining and rigorously establish how its volume, diversity, and composition affect visual-generation pretraining and downstream robot capability.

GE-Act 2.0 currently focuses on System-1-style instruction-conditioned control, mapping the current observation and language instruction directly to low-level manipulation without explicit deliberation. Robust task execution in open environments will additionally require complementary System-2 capabilities for deliberate reasoning, long-horizon planning, task decomposition, memory, and self-correction. We believe that coupling a scalable world–action model with such a deliberative system is a promising path toward general-purpose embodied intelligence, and future work will study how these two levels can plan, act, and improve together in closed loop.

8 Contributions and Acknowledgments

Core Contributors

Renhang Liu	Wenzhi Zhao	Zhuo Yang	Liliang Chen	Pengfei Zhou
Shengcong Chen	Guanghai Ren			

Contributors

Youlun Peng	Rongjun Jin	Nan Wang	Sukai Wang	Xindong He
Jinyuan Feng	Ziyu Xiong	Linqing Zhong	Yifei Wei	Feng Han
Long Zhang	Da Huang	Nanshu Zhao	Chenghao Yin	Mo Wu
Zhaodong Yan	Kongtao Hu	Yuxiang Yan	Aogelijiang Niyazi	Yu Fang
Jia Zeng	Lizhu Meng	Daizhen Lv	Haoyu Cao	Zhiwen Hou
Lianjin Ye	Yuehan Niu	Zhikai Cai	Xuan Hu	Hui Min
Xiongfeng Cai	Yue Liao	Jing Wu		

Academic Advisors

Soujanya Poria	Ye Li	Sanping Zhou
----------------	-------	--------------

Project Leaders

Liliang Chen	Guanghai Ren
--------------	--------------

Supervisors

Guanghai Ren	Maoqing Yao
--------------	-------------

Acknowledgments

We appreciate Maniformer’s R&D team for their in-depth technical support on data. We thank Yifei Chen for helpful discussions and assistance with polishing the paper.

References

- K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.
- J. Bruce, M. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, Y. Aytar, S. Bechtle, F. Behbahani, S. Chan, N. Heess, L. Gonzalez, S. Osindero, S. Ozair, S. Reed, J. Zhang, K. Zolna, J. Clune, N. de Freitas, S. Singh, and T. Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.15391.
- C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, H. Zhang, and M. Zhu. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.
- J. Chen, H. Cai, J. Chen, E. Xie, S. Yang, H. Tang, M. Li, Y. Lu, and S. Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025a. arXiv:2410.10733.
- R. Chen, Y. Yang, Z. Tang, D. Huo, T. Lin, H. Wu, H. Liu, Y. Chen, L. Zheng, B. Yuan, T. Li, M. Wang, D. Qi, B. Hu, W. Mei, Y. Xuan, H. Yang, Y. Zhu, M. Xu, Z. Ma, and X. Chang. ABot-M0.5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026.

- T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. 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>.
- X. Chen, J. Guo, T. He, C. Zhang, P. Zhang, D. C. Yang, L. Zhao, and J. Bian. IGOR: Image-GOal representations are the atomic control units for foundation models in embodied AI. *arXiv preprint arXiv:2411.00785*, 2024.
- Y. Chen, Y. Ge, W. Tang, Y. Li, Y. Ge, M. Ding, Y. Shan, and X. 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.
- D. Driess, J. T. Springenberg, B. Ichter, L. Yu, A. Li-Bell, K. Pertsch, A. Z. Ren, H. Walke, Q. Vuong, L. X. Shi, and S. Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.
- S. Fei, S. Wang, J. Shi, Z. Dai, J. Cai, P. Qian, L. Ji, X. He, S. Zhang, Z. Fei, et al. LIBERO-Plus: In-depth robustness analysis of vision-language-action models, 2025. URL <https://arxiv.org/abs/2510.13626>.
- K. Frans, D. Hafner, S. Levine, and P. 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.
- Z. Geng, M. Deng, X. Bai, J. Z. Kolter, and K. He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.
- Z. Geng, Y. Lu, Z. Wu, E. Shechtman, J. Z. Kolter, and K. 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.
- A. Gladstone, H. Ji, and Y. Du. Explorative modeling: Unlocking a third pretraining axis and end-to-end generation. *arXiv preprint arXiv:2607.27372*, 2026.
- Y. Guo, L. X. Shi, J. Chen, and C. Finn. Ctrl-World: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.
- Y. HaCohen, N. Chiprut, B. Brazowski, D. Shalem, D. Moshe, E. Richardson, E. Levin, G. Shiran, N. Zabari, O. Gordon, P. Panet, S. Weissbuch, V. Kulikov, Y. Bitterman, Z. Melumian, and O. Bibi. LTX-Video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.
- D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.
- D. Hafner, T. P. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.
- Y. Hu, Y. Guo, P. Wang, X. Chen, Y.-J. Wang, J. Zhang, K. Sreenath, C. Lu, and J. Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2412.14803.
- S. Huang, L. Chen, P. Zhou, S. Chen, Z. Jiang, Y. Hu, Y. Liao, P. Gao, H. Li, M. Yao, and G. Ren. EnerVerse: Envisioning embodied future space for robotics manipulation. *arXiv preprint arXiv:2501.01895*, 2025.
- S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. RL-Bench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

- J. Jang, S. Ye, Z. Lin, J. Xiang, J. Bjorck, Y. Fang, F. Hu, S. Huang, K. Kundalia, Y.-C. Lin, et al. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.
- Y. Jiang, S. Chen, S. Huang, L. Chen, P. Zhou, Y. Liao, X. He, C. Liu, H. Li, M. Yao, and G. Ren. EnerVerse-AC: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025.
- D. Kim, C.-H. Lai, W.-H. Liao, N. Murata, Y. Takida, T. Uesaka, Y. He, Y. Mitsufuji, and S. Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations (ICLR)*, 2024a. *arXiv:2310.02279*.
- M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024b.
- S. Kim, S. Jang, B. Yoon, D. Kim, J. Won, and J. Shin. RoboCurate: Harnessing diversity with action-verified neural trajectory for robot learning. *arXiv preprint arXiv:2602.18742*, 2026.
- L. Li, Q. Zhang, Y. Luo, S. Yang, R. Wang, F. Han, M. Yu, Z. Gao, N. Xue, X. Zhu, Y. Shen, and Y. Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026. LingBot-VA.
- Y. Liao, P. Zhou, S. Huang, D. Yang, S. Chen, Y. Jiang, Y. Hu, J. Cai, S. Liu, J. Luo, L. Chen, S. Yan, M. Yao, and G. Ren. Genie Envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025.
- S. Lin, X. Xia, Y. Ren, C. Yang, X. Xiao, and L. Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.
- Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. *arXiv:2210.02747*.
- B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023a.
- X. Liu, C. Gong, and Q. 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*.
- X. Liu, X. Zhang, J. Ma, J. Peng, and Q. 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*.
- Y. Liu, S. Wang, D. Wei, X. Cai, L. Zhong, J. Yang, G. Ren, J. Zhang, M. Yao, C. Li, X. He, L. Chen, and J. Luo. Unified embodied VLM reasoning with robotic action via autoregressive discretized pre-training. *arXiv preprint arXiv:2512.24125*, 2025.
- S. Luo, Y. Tan, L. Huang, J. Li, and H. Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.
- J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. 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, C. Xiang, F. Bao, H. Liu, H. Tan, H. Bi, J. Li, J. Liu, J. Pang, K. Jing, L. Liu, M. Cai, R. Cui, R. Zhao, R. Wang, S. Huang, Y. Feng, Y. Rong, Z. Wang, and J. Zhu. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026.
- L. Mur-Labadia, M. Muckley, A. Bar, M. Assran, K. Sinha, M. Rabbat, Y. LeCun, N. Ballas, and A. Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.
- NVIDIA, N. Agarwal, A. Ali, M. Bala, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025a.

- NVIDIA, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. 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, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.
- Physical Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.
- B. Qiu, L. Chen, Y. Liao, N. Wang, L. Wang, J. Luo, W. Zhao, S. Chen, D. Chen, Y. Li, C. Gao, S. Yan, S. Liu, M. Yao, and G. 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>.
- T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2202.00512.
- Y. Shang, Z. Li, Y. Ma, W. Su, X. Jin, Z. Wang, L. Jin, X. Zhang, Y. Tang, H. Su, C. Gao, W. Wu, X. Liu, D. Shah, Z. Zhang, Z. Chen, J. Zhu, Y. Tian, T.-S. Chua, W. Zhu, and Y. Li. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026.
- O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolan, J. Brandt, C. Couprie, J. Mairal, H. Jégou, P. Labatut, and P. Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.
- Y. Song, P. Dhariwal, M. Chen, and I. 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.
- M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, O. Hénaff, J. Harmsen, A. Steiner, and X. Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.
- Wan Team, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.
- S. Wang, J. Shi, Z. Fu, X. He, F. Liu, C. Yang, Y. Zhou, Z. Fei, J. Gong, J. Fu, M. Z. Shou, X. Huang, X. Qiu, and Y.-G. Jiang. World action models: The next frontier in embodied AI. *arXiv preprint arXiv:2605.12090*, 2026.
- C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems (RSS)*, 2024. arXiv:2401.00025.
- C. Wu, X. Chen, Z. Wu, Y. Ma, X. Liu, Z. Pan, W. Liu, Z. Xie, X. Yu, C. Ruan, and P. Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024a.
- H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2024b. arXiv:2312.13139.
- P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg. DayDreamer: World models for physical robot learning. In *Conference on Robot Learning*, 2022.

- W. Wu, F. Wang, F. Lu, H. Sun, S. Liu, Y. Wang, Y. Yan, Y. Wang, S. Ma, X. Wang, Y. Liu, S. Yang, T. Zhou, K. Zhang, L. Zhou, C. Su, N. Xue, B. Tan, H. Zhang, Y. Zhang, F. Liao, X. Zhu, Y. Shen, and K. Zheng. From foundation to application: Improving VLA models in practice. *arXiv preprint arXiv:2607.06403*, 2026. LingBot-VLA 2.0 technical report.
- J. Xie, W. Mao, Z. Bai, D. J. Zhang, W. Wang, K. Q. Lin, Y. Gu, Z. Chen, Z. Yang, and M. Z. Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2408.12528.
- J. Yao, Y. Song, Y. Zhou, and X. Wang. Towards scalable pre-training of visual tokenizers for generation. *arXiv preprint arXiv:2512.13687*, 2025a.
- J. Yao, B. Yang, and X. 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.
- A. Ye, B. Wang, C. Ni, G. Huang, G. Zhao, H. Li, H. Li, J. Li, J. Lv, J. Liu, M. Cao, P. Li, Q. Deng, W. Mei, X. Wang, X. Chen, X. Zhou, Y. Wang, Y. Chang, Y. Li, Y. Zhou, Y. Ye, Z. Liu, and Z. Zhu. GigaWorld-Policy: An efficient action-centered world-action model. *arXiv preprint arXiv:2603.17240*, 2026a.
- S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Liden, K. Lee, J. Gao, L. Zettlemoyer, D. Fox, and M. Seo. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.11758.
- S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang, A. Malik, K. Lee, W. Liang, N. Ranawaka, J. Gu, Y. Xu, G. Wang, F. Hu, A. Narayan, J. Bjorck, J. Wang, G. Kim, D. Niu, R. Zheng, Y. Xie, J. Wu, Q. Wang, R. Julian, D. Xu, Y. Du, Y. Chebotar, S. Reed, J. Kautz, Y. Zhu, L. Fan, and J. Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b.
- C. Yin, D. Huang, D. Yang, J. Wang, N. Zhao, C. Xu, W. Sun, L. Hou, Z. Li, J. Wu, et al. Genie Sim 3.0: A high-fidelity comprehensive simulation platform for humanoid robot, 2026. URL <https://arxiv.org/abs/2601.02078>.
- T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and W. T. Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.
- T. Yin, M. Gharbi, R. Zhang, E. Shechtman, F. Durand, W. T. Freeman, and T. Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b. arXiv:2311.18828.
- S. Yu, S. Kwak, H. Jang, J. Jeong, J. Huang, J. Shin, and S. Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.06940.
- H. Yuan, Z. Liang, A. Chen, Y. Wang, H. Li, P. Lin, Y. Huang, Z. Lei, T. Zhang, J. Zhang, J. Zhang, J. Fan, G. Zhou, Q. Peng, C. Lv, X. Chen, A. Yang, F. Huang, J. Lin, D. Liu, J. Zhou, C. Wu, and X.-H. Chen. Qwen-RobotManip technical report: Alignment unlocks scale for robotic manipulation foundation models, 2026. URL <https://arxiv.org/abs/2606.17846>.
- X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. arXiv:2303.15343.
- Q. Zhang, L. Li, L. Zhang, S. Yang, Y. Luo, S. Li, R. Wang, J. Wang, J. Shao, G. Xu, J. Zhou, Y. Shen, Y. Jin, F. Xu, S. Ma, J. Liao, G. Lu, Z. Shi, Y. Wen, Y. Zhao, W. Tang, X. Wang, C. Li, J. Zhu, K. L. Cheng, N. Xue, X. Zhu, Y. Shen, and Y. Xu. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026. LingBot-VA 2.0.
- L. Zhong, Y. Liu, Y. Wei, Z. Xiong, M. Yao, S. Liu, and G. 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.

- P. Zhou, L. Chen, S. Chen, D. Chen, W. Zhao, R. Jin, G. Ren, and J. Luo. Act2Goal: From world model to general goal-conditioned policy. *arXiv preprint arXiv:2512.23541*, 2025.
- P. Zhou, S. Chen, D. Chen, et al. τ_0 -WM: A unified video-action world model for robotic manipulation. *arXiv preprint arXiv:2606.01027*, 2026.
- F. Zhu, H. Wu, S. Guo, Y. Liu, C. Cheang, and T. Kong. IRASim: A fine-grained world model for robot manipulation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2406.14540.

A OOD Generalization in Simulation

Real-robot evaluation is central to this report, but matching hardware, scenes, and evaluation conditions across independently developed models is difficult. We therefore complement the main real-robot experiments with three standardized simulation benchmarks spanning environment randomization, instruction following, and controlled visual and embodiment perturbations. Because GE-Act 2.0 is pretrained mostly on real-world data, we first SFT it on each benchmark’s designated in-distribution training condition, placing it on the same footing as models adapted specifically to that simulator, and evaluate it only on held-out OOD simulation conditions. This protocol provides a more comparable cross-model evaluation while preserving a separation between adaptation and OOD evaluation, following the principle used by Qwen-RobotManip (Yuan et al., 2026).

A.1 RoboTwin Clean-to-Random Generalization

RoboTwin Clean-to-Random tests whether a policy adapted only to a clean simulator remains effective as the scene distribution changes (Chen et al., 2025b). We SFT GE-Act 2.0 on the Easy/Clean training setting and evaluate it without further adaptation under changes to background, lighting, clutter, and table height, as well as the combined Hard condition. Table 3 reports the task-averaged success rate for every evaluation condition. The compared models are StarVLA (StarVLA Community, 2026), GR00T N1.7 (NVIDIA et al., 2025b), and $\pi_{0.5}$ (Physical Intelligence et al., 2025).

Table 3: RoboTwin Clean-to-Random success rate (%). Models are adapted only on the Easy/Clean setting; the remaining columns evaluate OOD environment variations. The best result in each column is 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

GE-Act 2.0 achieves the strongest overall clean-to-random performance, leading five of the six reported conditions. Under the combined Hard randomization, it reaches 60.52% success, 12.62 percentage points above the strongest baseline, $\pi_{0.5}$ at 47.90%. Its performance decreases by 16.19 points from Easy to Hard, compared with a 25.20-point decrease for $\pi_{0.5}$, indicating better retention as multiple shifts are combined. Lighting is the only condition in which GE-Act 2.0 does not lead, reaching 65.92% versus 69.20% for $\pi_{0.5}$.

A.2 GenieSim-Instruction: OOD Instruction Following

GenieSim-Instruction provides an OOD test of whether a policy can select the correct manipulation behavior from diverse language specifications (Yin et al., 2026). The benchmark itself separates training from evaluation: we SFT on its released instruction-following training demonstrations and evaluate on prescribed configurations with unseen background textures, object positions, and related scene variations. The ten tasks probe color, number, shape, size, object type, specific-object reference, common-sense reference, logical composition, and object straightening. We report the official normalized task score and its average across tasks in Table 4. The compared models are π_0 (Black et al., 2024), GR00T N1.7 (NVIDIA et al., 2025b), LingBot-VLA 2.0 (Wu et al., 2026), $\pi_{0.5}$ (Physical Intelligence et al., 2025), and ACoT-VLA (Zhong et al., 2026).

Table 4: Results on the OOD GenieSim-Instruction benchmark (normalized score, $\uparrow$). Evaluation configurations vary background textures, object positions, and other scene factors relative to the training demonstrations. The best result in each row is bolded.

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

GE-Act 2.0 obtains the highest average score of 0.770, compared with 0.757 for ACoT-VLA and 0.746 for $\pi_{0.5}$. It leads on four tasks, with its clearest margins on logical composition and common-sense reference, where it improves over the previous best by 0.08 and 0.06, respectively. The advantage is not uniform: the strongest baseline remains ahead on number, shape, size, and several direct attribute-matching tasks. The aggregate improvement therefore reflects complementary instruction-following strengths rather than uniform dominance across every language category.

A.3 LIBERO-Plus Robustness across Perturbation Axes

LIBERO-Plus tests whether a policy adapted on standard LIBERO remains reliable under controlled changes to its observations, instructions, and execution conditions (Fei et al., 2025). We SFT GE-Act 2.0 on the standard LIBERO training data and evaluate it on seven LIBERO-Plus perturbation axes: camera viewpoint, robot initial state, language, lighting, background, sensor noise, and object layout. Table 5 reports the success rate for each axis and the overall benchmark average. The compared models are StarVLA (StarVLA Community, 2026) and $\pi_{0.5}$ (Physical Intelligence et al., 2025).

Table 5: LIBERO-Plus success rate (% , $\uparrow$) after adaptation on standard LIBERO. Each column evaluates a held-out perturbation axis. Overall is the success rate over all 10,030 task instances, so axes are weighted by their instance counts, following the benchmark’s protocol; all models are scored the same way. The best result in each column is bolded.

Model	Camera	Robot	Language	Light	Background	Noise	Layout	Overall
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 reaches 80.4% overall, exceeding StarVLA by 6.3 percentage points while remaining 4.0 points below $\pi_{0.5}$. It performs best under camera and sensor-noise perturbations, reaching 94.1% and 95.5%, respectively, and remains close to $\pi_{0.5}$ under language and layout changes. Robot-state and especially background perturbations remain the principal weaknesses, showing that robustness is not yet uniform across the seven axes.

Taken together, the simulation results show the strongest generalization under compounded scene randomization and OOD instruction following. The per-axis LIBERO-Plus results provide a more diagnostic picture, identifying background and robot-state shifts as important remaining failure modes. These standardized comparisons therefore complement the real-robot atomic-task evaluation by exposing both broad OOD capability and benchmark-specific limitations.

B 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.

B.1 Architectures

Table 6: 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. (5).
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.

B.2 Temporal Schedule and Sample Construction

Frames and actions. The defaults of Eq. (3) 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 .

B.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. (9) is computed exactly with forward-mode autodiff, and the detached correction $(t - r) du_\theta / dt$ is clamped element-wise to $[-5, 5]$; the time embedders are zero-initialized so that the correction starts at zero. The losses of Eq. (13) are averaged over latent elements with dense and sparse frames weighted equally, and each per-sample loss is normalized as $\mathcal{L} / \text{sg}[\mathcal{L} + 0.1]$ 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.

B.4 IDM Pretraining

The IDM is trained on recorded futures with Eq. (15), 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 3.4.5 uses the same IDM checkpoint.

B.5 KASO Co-training

Fused update. Each optimizer step draws one batch of instruction–video–action samples and accumulates three micro-steps on it: $\mathcal{L}_{\text{SVP}}$ on the recorded video, $\mathcal{L}_{\text{IDM}}$ on the recorded future, and $\mathcal{L}_{\text{E2E}}^{\text{sel}}$ on the selected generated future, with coefficients $(\lambda_{\text{SVP}}, \lambda_{\text{IDM}}, \lambda_{\text{E2E}}) = (1, 0.1, 1)$ 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. (19) 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. All ladder rungs 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 largest rung uses the full corpus of Figure 8.

For the controlled ablation of Section 3.4.5, 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 $\mathcal{L}_{\text{E2E}}$ alone.

B.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 5 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 C.

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

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

where $\mathcal{L}_{\text{sel}}^{(8)}$ 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 $(B(\hat{z}_k) - a)^2$, selects the minimum-energy candidate, and then re-integrates from that candidate’s retained noise with gradients enabled. Only this replayed trajectory contributes $\mathcal{L}_{\text{sel}}^{(8)}$. 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.

C Full Results

C.1 Atomic-Skill Taxonomy and Evaluation Details

Our real-robot evaluation suite contains 100 atomic task instances in 20 skill groups (Table 7). 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 7: 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 centre; 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.

C.2 Task-Level Data-Scaling Results

Table 8 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 5.2.1. The group headings and task descriptions match the taxonomy in Table 7.

Table 8: 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
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

Table 8: 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
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
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

C.3 Instruction-Following Prompts and Results

Protocol. The fine-grained grounding study of Section 5.3.1 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 9 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 13 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.

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 9 lists the subcategory beside every instruction. Instructions are reproduced verbatim as issued to the policy, including minor typographical irregularities.

Table 9: 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

Table 9: Instruction-level results of the fine-grained grounding study (continued).

Dimension	Instruction	Follow	SR
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
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 14 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 10 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 10: Lexical matching rules and descriptor frequencies used in Figure 14. 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; <i>orange</i> as a fruit name excluded	42,161	44.4	29.6
Size	<i>large, small, big, tiny, medium</i> and comparative/superlative forms	5,299	5.6	0.96
Shape	<i>round, square, rectangular, cylindrical, triangular</i>	574	0.6	0.67
Position	<i>left, right, near, far</i> ; body-side phrases such as <i>right hand</i> excluded unless another spatial relation is present	11,927	12.6	13.8
Order	<i>middle, center, ordinals, leftmost, rightmost</i> ; generic <i>between</i> excluded	2,440	2.6	0.13