HIVE-BENCH: EVALUATING PATCH-LEVEL VISUAL REPRESENTATIONS FOR EGOCENTRIC ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained visual encoders are widely reused in robot manipulation, yet most evaluations read them through a pooled embedding of third-person images, whereas bimanual policies consume dense patch tokens from onboard cameras. We introduce HIVE-Bench, to our knowledge the first benchmark that compares dense patch-level representations for egocentric bimanual manipulation under one shared policy: a flow-matching action head reads the patch tokens of 20 encoders on 24 RoboCasa-GR1 and RoboTwin 2.0 tasks, across frozen and fine-tuned encoders, training regimes, and data, model, and layer sweeps, and we relate more than fifty diagnostics and published vision-benchmark scores to closed-loop success. Three findings run against common practice. Averaging each view's patch tokens into one token removes most of the success of frozen DINOv2 and DINOv3. Measures of world reconstruction, from segmentation, depth, and correspondence scores to probes that decode robot state or object position, either do not track success or lose it once vision–language models join, whereas an action readout that decodes commanded motion from two head-camera frames tracks it in every setting we test (Spearman $|\rho| \geq 0.54$). And better vision is not better control: scale and newer pretraining raise benchmark scores but not success.

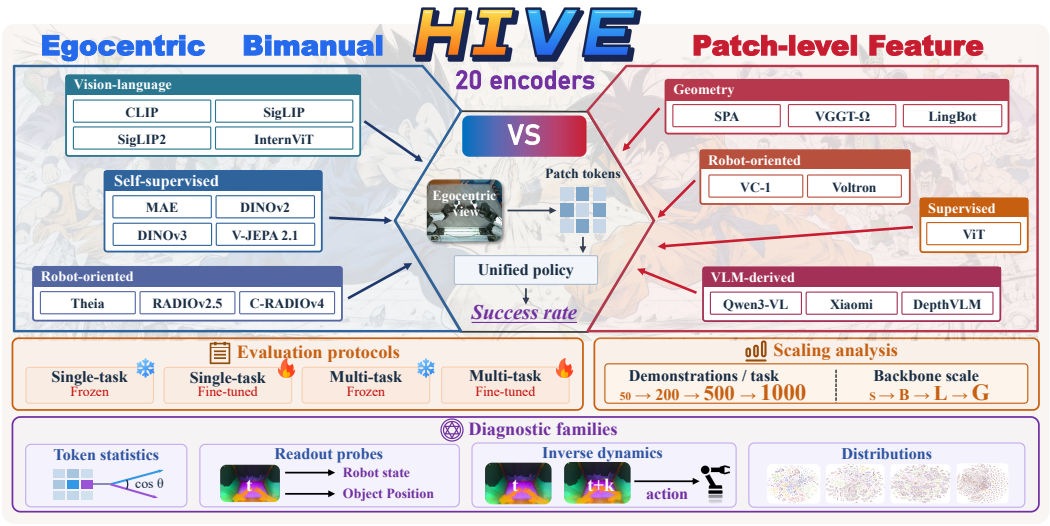


Figure 1: HIVE-Bench. Twenty pretrained encoders from seven families compete through one shared patch-conditioned policy on egocentric bimanual tasks, across data and model scales, training regimes, and adaptation.

1 INTRODUCTION

Robot manipulation policies increasingly reuse pretrained visual encoders instead of learning vision from scratch, which transfers poorly across objects, robots, and viewpoints (Nair et al., 2022; Majumdar et al., 2023; Jiang et al., 2025a; Parisi et al., 2022; Xie et al., 2024). The candidates come from very different training signals: self-supervision, vision–language alignment, 3D reconstruction,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

robot data, and the hidden states of vision–language models (VLMs) (He et al., 2022; Oquab et al., 2024; Radford et al., 2021; Zhu et al., 2025; Bai et al., 2025). Choosing among them still requires training and rolling out policies, so the field relies on shared evaluations and on cheap proxies that promise to rank encoders before any policy is trained.

Most of these evaluations read each encoder through one global vector, a pooled embedding or a class token, and observe the scene from a third-person camera (Majumdar et al., 2023; Hu et al., 2023; Dasari et al., 2023; Burns et al., 2023; Shang et al., 2024; Jiang et al., 2025a; Dong et al., 2026). Their proxies mostly measure how well an encoder reconstructs the world, through segmentation, depth, correspondence, or the decoded state of the robot and its objects (Burns et al., 2023; Zhu et al., 2025; Dong et al., 2026; Hu et al., 2023). Modern bimanual policies differ from this setting on both counts. They read dense patch tokens, which keep the local detail that separates nearby grasp points (Kim et al., 2024; Black et al., 2024; Zhou et al., 2026a; Tsagkas et al., 2025), and they see the scene from head and wrist cameras that move with the robot and are often occluded (Zhao et al., 2023; Fu et al., 2024; Chen et al., 2025b; Chi et al., 2024). We therefore ask *how pretrained encoders compare when a policy reads their patch tokens from onboard cameras*, and *which measurable properties of those tokens track the resulting success*.

We introduce **HIVE-Bench** (Figure 1), to our knowledge the first benchmark that compares dense patch-level representations for egocentric bimanual manipulation under one shared policy. A flow-matching DiT action head (Peebles & Xie, 2023; Bjorck et al., 2025) reads the patch tokens of 20 pretrained encoders from seven families on 12 RoboCasa-GR1 and 12 RoboTwin 2.0 tasks (Nasiriany et al., 2024; Chen et al., 2025b), with frozen and fine-tuned encoders, single-task and multi-task training, and sweeps over demonstrations, model size, and VLM layer. More than fifty diagnostics in four families describe each encoder, and we relate them, together with published vision-benchmark scores, to closed-loop success and test whether each association survives a change of setting.

Three results run against common practice. *The patch interface matters*: for frozen DINOv2 and DINOv3, averaging each view's patch tokens into one token removes two thirds to four fifths of the success, so pooled evaluations probe a different interface from the one patch-conditioned policies use. *Reading the action beats reconstructing the world*: an IDM probe that decodes the commanded displacement from two head-camera frames is the only measurement that tracks success in every setting we test, including VLMs and fine-tuned encoders ($|\rho| \geq 0.54$), whereas state and object readouts lose their association once VLMs join and segmentation, depth, and correspondence scores do not track success at all. *Better vision is not better control*: scale and newer pretraining raise benchmark scores but not success, and DINOv3 beats DINOv2 on segmentation, depth, and correspondence at every matched size yet trails it by about 20 points on RoboTwin. Label-free token statistics, including the popular effective rank, are weak screens that fade once fine-tuned encoders join, and fine-tuning raises all fourteen frozen and fine-tuned pairs while largely keeping the frozen order. Our contributions are:

- **The first egocentric patch-level benchmark:** one shared policy reads the dense patch tokens, not a pooled vector, of 20 encoders from onboard cameras on two bimanual suites, through documented token interfaces (Section 3).
- **Evaluation from many angles:** single-task and multi-task training, frozen and fine-tuned encoders, demonstration and model-size scaling, VLM layers, and patch versus pooled tokens, over 500 trained policies (Section 4).
- **A comprehensive study of what tracks success:** over fifty diagnostics and published benchmarks related to closed-loop success with graded statistics across cohorts, encoder types, and adaptation, distilled into rules for choosing encoders (Sections 5 and 6).

2 Related Work

Pretrained representations for control. Global embeddings made pretrained vision easy to reuse in control (Parisi et al., 2022; Nair et al., 2022; Majumdar et al., 2023). Robot-oriented encoders learn from egocentric video or robot data (Majumdar et al., 2023; Karamcheti et al., 2023; Zhu et al., 2025), and distilled encoders merge several teachers (Shang et al., 2024; Heinrich et al., 2025; Ranzinger et al., 2026). Evaluations have compared adaptation (Majumdar et al., 2023), downstream learners (Hu et al., 2023), pretraining data (Dasari et al., 2023), and objectives (Jing et al., 2023); most read one pooled feature from third-person views, and each trains its own policy.

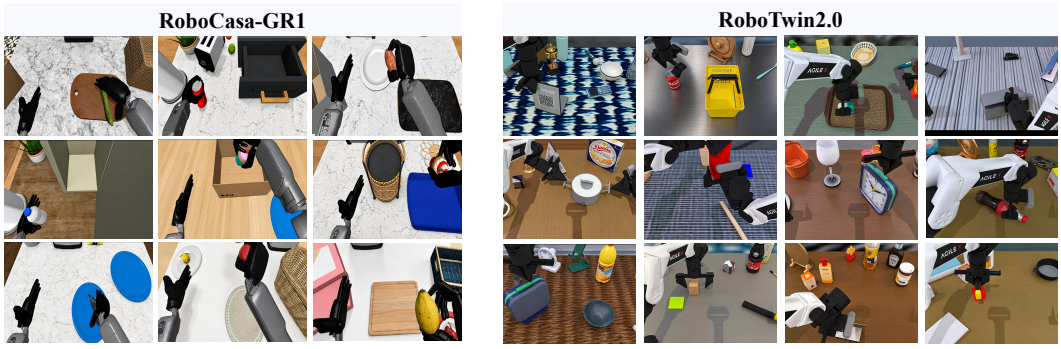


Figure 2: Simulation environments. Sample scenes from RoboCasa-GR1 (left), where a humanoid with dexterous hands works in varied kitchens, and RoboTwin 2.0 (right), where two arms with grippers work on tabletops with randomized clutter, lighting, and textures.

Spatial policy interfaces and egocentric bimanual manipulation. ALOHA, ACT, Mobile ALOHA, UMI, DexMimicGen, and GR00T advanced bimanual skills with onboard views, dexterous hands, and humanoids (Zhao et al., 2023; Fu et al., 2024; Chi et al., 2024; Jiang et al., 2025b; Bjorck et al., 2025). Policies differ in how much spatial detail they keep, from TokenLearner compression in RT-1 to dense pretrained tokens in Patch Policy (Brohan et al., 2023b; Octo Model Team et al., 2024; Zhou et al., 2026a), and VLA systems and world-action models integrate vision with action prediction (Brohan et al., 2023a; Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025; Ye et al., 2026; Yuan et al., 2026). HIVE-Bench holds the action head fixed, so only the encoder and its token interface change between two compared policies.

Diagnosing representation quality. Proposed proxies include segmentation ability (Burns et al., 2023), objectness (Wen et al., 2025), manipulation relevance (Jiang et al., 2025a), token-norm entropy (Shang et al., 2024), attention allocation (Wang et al., 2026c), effective rank (Garrido et al., 2023), physical structure (Zhu et al., 2025; Dong et al., 2026), temporal progression (Tsagkas et al., 2026), and action recoverability from world-model latents (Nilaksh et al., 2026). Each measures one aspect under its own policy. We measure four diagnostic families and published vision benchmarks under one patch-level policy, across regimes and adaptation, and grade every association. Appendix L extends this discussion and compares 23 studies in a coverage matrix.

3 HIVE-BENCH

HIVE-Bench compares encoders by the closed-loop success of one shared patch-conditioned policy and asks which measurable token properties go with that success. Appendices A–F give every detail.

3.1 BENCHMARKS AND PROTOCOLS

We use 12 RoboCasa-GR1 kitchen tasks performed by a humanoid with dexterous hands (Nasiriany et al., 2024; Bjorck et al., 2025) and 12 domain-randomized RoboTwin 2.0 tasks performed by two arms with grippers (Chen et al., 2025b) (Figure 2, Table 1; tasks in Table 18). RoboCasa covers object transfer and cabinet, drawer, and microwave interactions; RoboTwin covers tool use, handover, lifting, placement, rotation, and articulated objects under randomized clutter, lighting, and textures. The benchmarks differ in robot, action space, cameras, tasks, data, and randomization at once, so differences between them are observational. No policy receives proprioception: what a policy knows about the robot's configuration must come from its images.

We cross single-task training (ST, one policy per task) and multi-task training (MT, one language-conditioned policy for all tasks) with frozen and fine-tuned encoders: all 20 encoders frozen in both regimes on RoboCasa and multi-task on RoboTwin, and seven of them fine-tuned, single-task on RoboCasa and multi-task on RoboTwin. Further studies vary the number of demonstrations (50 to 1,000 per task), the checkpoint size (six families, 11 additional checkpoints), and the VLM layer, and one control replaces each view's patch tokens by their mean. Every configuration is trained with one seed and evaluated with three evaluation seeds of 50 rollouts per task, for over 500 trained policies

Table 1: Benchmarks and protocols. Both benchmarks have 12 tasks, condition the policy on the current images without proprioception, and predict 16-step action chunks. Demos are per task; "Executed" is the part of each chunk run before replanning; the last two columns give the regimes and, in parentheses, the number of encoders. Every diagnostic reads the head camera.

Benchmark	Robot	Demos	Views	Action	Executed	Frozen	Fine-tuned
RoboCasa-GR1	Humanoid, hands	1,000	Head	29-D	12 of 16	ST, MT (20)	ST (7)
RoboTwin 2.0	Two arms	500	Head, 2 wrists	14-D joints	16 of 16	MT (20)	MT (7)

and 250,000 evaluation rollouts. A suite score is the mean of the twelve task scores; its $\pm$ propagates the per-task deviations across evaluation seeds, assuming independent task errors.

3.2 Encoders and the token interface

The 20 encoders span seven families: a supervised ViT (Dosovitskiy et al., 2021); the self-supervised MAE, DINOv2, DINOv3, and V-JEPA 2.1 (He et al., 2022; Oquab et al., 2024; Siméoni et al., 2025; Mur-Labadia et al., 2026); the geometry-oriented SPA, VGGT-Ω, and LingBot-Vision (Zhu et al., 2025; Wang et al., 2026a; Fu et al., 2026); the vision–language CLIP, SigLIP, SigLIP2, and InternViT (Radford et al., 2021; Zhai et al., 2023; Tschannen et al., 2025; Chen et al., 2024); the robot-oriented VC-1 and Voltron (Majumdar et al., 2023; Karamcheti et al., 2023); the multi-teacher Theia, RADIOv2.5, and C-RADIOv4 (Shang et al., 2024; Heinrich et al., 2025; Ranzinger et al., 2026); and three VLMs: Qwen3-VL and its derivatives DepthVLM and Xiaomi-Robotics-1 (Bai et al., 2025; Yu et al., 2026; Xiaomi Robotics Team et al., 2026).

The policy reads every patch token of every view, without pooling or normalization; class and register tokens are removed. Views are stored at 224×224, giving 196 or 256 tokens per view for patch size 16 or 14, 576 for V-JEPA 2.1 at 384 pixels, and 64 for each VLM (Table 2). For the VLMs we keep the final-layer hidden states at the image-token positions; they do not depend on the instruction, and we do not use Xiaomi-Robotics-1's action head. Two interfaces mix views before the policy reads them: VGGT-Ω receives the three RoboTwin views as one input, and a VLM's image tokens of later views attend to earlier ones. We therefore compare an encoder under a documented extraction interface, not in isolation (Appendix B).

3.3 The shared policy

The action head is a DiT flow-matching model derived from GR00T (Bjorck et al., 2025; Peebles & Xie, 2023; Lipman et al., 2023) with 16 blocks of width 768 (Figure 7). It operates on 32 learned query tokens and 16 noisy action tokens H_t; the eight even blocks cross-attend to the context C_t, the eight odd blocks self-attend, and adaptive layer normalization conditions every block on the flow time. C_t concatenates the patch tokens of all views, followed in multi-task runs by projected features of a frozen CLIP text encoder, so vision enters the policy only through

$$\mathrm{CrossAttn}(H_t, C_t) = \mathrm{softmax}\big((H_t W_Q)(C_t W_K)^\top / \sqrt{d_h}\big)\, C_t W_V. \tag{1}$$

The head regresses the flow-matching velocity toward the normalized action chunk, with the flow time drawn from $\mathrm{Beta}(1.5, 1)$ and eight noise draws per sample, and integrates 15 (RoboCasa) or 10 (RoboTwin) Euler steps at inference. Every policy is trained for 100k steps with a batch of 128 on four GPUs, using AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, a learning rate of 10^{-4} with warmup and cosine decay, and gradient clipping at 1. The budget is thus equal per policy, not per task. Frozen runs update the head and projections, and fine-tuned runs also update the encoder (Appendix C).

3.4 Diagnostics and statistics

Four diagnostic families describe each encoder, all measured on head-camera frames (Appendix D).

- **Token statistics** need only frames and describe the norms, variance, spatial frequencies, and temporal change of patch tokens, sampled on the first 100 steps of every fifth demonstration. *Norm-histogram entropy* bins a frame's patch-token norms into 30 bins spanning its range; it is low both when norms are nearly equal and when a few large norms stretch the range.

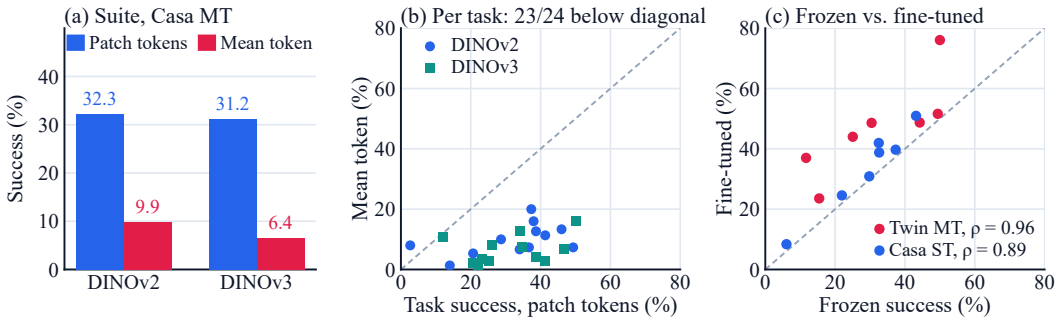


Figure 3: Two controlled comparisons. (a, b) Replacing each view's patch tokens by their mean, with the frozen encoder, head, and RoboCasa multi-task recipe unchanged. (c) Fine-tuned against frozen success of seven encoders on RoboTwin multi-task and RoboCasa single-task; the dashed line marks equal success.

- **Readout probes** read frozen tokens through a four-layer transformer with a learned query token. The *IDM probe* (width 256, 2.4–2.8M parameters) reads two frames 16 steps apart and predicts the standardized terminal command displacement $a_{t+15} - s_t$, not an action sequence; the *state probe* (at the token width) decodes the joint state s_t from one frame. Probes train for four epochs on 1,000 RoboCasa or 400 RoboTwin episodes and are evaluated on 500 or 100 held-out episodes. Since s_t is visible in the first frame, part of the IDM target is readable from one frame. An object probe on four RoboTwin tasks has no chance baseline, so no conclusion rests on it.
- **A trained-policy diagnostic** measures how much a trained RoboTwin policy's action chunk varies across 100 background textures of the same scene and diffusion noise.
- **A distribution-shaping probe** reshapes pooled features toward a Gaussian or a sphere; its associations hold on one benchmark each, so it is reported in Appendix I.1.

We also collect published vision-benchmark scores for the checkpoints that have them (Section 5.2).

Statistics. The unit is an encoder. The main cohorts are the 17 vision encoders on RoboCasa single-task (Casa ST), RoboCasa multi-task (Casa MT), and RoboTwin multi-task (Twin MT); Twin+FT adds the seven fine-tuned RoboTwin checkpoints, and the three VLMs are examined separately. We report Spearman's ρ with a 100,000-shuffle permutation p, a Benjamini–Hochberg q over the 49 to 51 diagnostics of a cohort, and 95% bootstrap intervals. An association is *robust* if $p < 0.05$ and $q < 0.10$, *nominal* if $p < 0.05$ only, and a *trend* if its sign agrees across cohorts; post-hoc groupings are *exploratory*. Encoders share families and teachers, so we rely on agreement across cohorts and tasks and on leave-one-family-out checks rather than on single p-values. Numbered findings hold in every cohort; observations cover the rest (Appendix F).

4 How Do Encoders Compare in Closed Loop?

Table 2 reports frozen success on the three suites together with each encoder's token interface; DINOv2, DINOv3, and V-JEPA 2.1 are Base checkpoints and C-RADIOv4 is the H checkpoint. Appendix G gives every study in full, and Appendix K every per-task value.

Finding 1: The top and bottom of the ranking are stable across the three suites; the middle is not. VGGT-Ω, DINOv2, and LingBot-Vision stay in the top six on all three suites, and CLIP, ViT, C-RADIOv4, and RADIOv2.5 in the bottom five, tens of points apart, while a suite score is uncertain by about one point. DINOv3, VC-1, Voltron, and the three VLMs each move by eight places or more. Over all 20 encoders, rankings agree at $\rho = 0.89$ between the RoboCasa regimes but only at about 0.60 between benchmarks, and at 0.72–0.83 without the VLMs, which take the first three places on RoboTwin and sit mid-table on RoboCasa, a shift that no single factor explains in our design. Family labels do not order the encoders: geometry contains both VGGT-Ω and SPA, self-supervision both DINOv2 and MAE.

Finding 2: For frozen DINOv2 and DINOv3, one mean token per view removes most of the success of the patch-token policy. Mean pooling lowers DINOv2 from 32.3% to 9.9% and DINOv3 from 31.2% to 6.4% on RoboCasa multi-task (Figure 3a). The drop is not driven by a

Table 2: Frozen success and rank across three suites. Encoder names are colored by family: geometry, self-supervised, VLM, vision–language, robot-oriented, multi-teacher, and supervised. $N \times D$ gives the patch tokens per view and their width; †projected to 1,024 from a hidden width of 2,560. SR is the twelve-task mean success (%) with the propagated uncertainty over three evaluation seeds; rank 1 is the highest success in a column, and ties share their mean rank. **First**, **second**, and **third** are marked per suite. On RoboTwin, VGGT-Ω receives the three views as one input and VLM tokens of later views attend to earlier ones.

			RoboCasa ST		RoboCasa MT		RoboTwin MT	
Encoder	**Params**	$N \times D$	**SR**	**Rk**	**SR**	**Rk**	**SR**	**Rk**
VGGT-Ω	1B	196×2,048	**43.17 ±1.4**	**1**	**35.39 ±1.3**	**1**	50.00 ±1.3	4
LingBot-Vision	300M	196×1,024	31.78 ±0.9	6	29.39 ±1.5	4	46.78 ±1.4	6
SPA	86M	196×768	21.94 ±1.4	15	10.94 ±0.8	18	11.67 ±0.8	20
DINOv2	86M	256×768	**37.33 ±0.9**	**2**	**32.28 ±1.5**	**2**	49.39 ±1.0	5
DINOv3	86M	196×768	32.50 ±1.4	4	**31.22 ±1.3**	**3**	30.44 ±1.4	11
MAE	86M	196×768	22.89 ±1.1	13	25.28 ±1.5	7	26.67 ±1.6	13
V-JEPA 2.1	80M	576×768	23.39 ±1.0	12	18.67 ±1.2	13	39.61 ±1.3	9
Xiaomi-Robotics-1	4.4B	64×1,024†	24.83 ±1.6	10	25.17 ±1.7	8	**53.22 ±1.3**	**2**
DepthVLM	4B	64×1,024†	28.22 ±1.6	8	18.50 ±1.1	14	**53.67 ±1.2**	**1**
Qwen3-VL	4B	64×1,024†	23.78 ±1.7	11	19.56 ±1.2	12	**51.44 ±1.2**	**3**
SigLIP2	86M	196×768	29.78 ±1.6	7	29.11 ±1.5	5	44.22 ±1.2	7
InternViT	300M	256×1,024	22.39 ±1.0	14	20.11 ±1.0	11	30.94 ±1.3	10
SigLIP	86M	196×768	16.61 ±1.1	16	13.67 ±1.3	15.5	28.39 ±1.4	12
CLIP	86M	196×768	15.17 ±1.5	17	11.44 ±1.0	17	21.33 ±0.8	16
VC-1	86M	196×768	**32.61 ±1.2**	**3**	28.44 ±1.4	6	25.06 ±1.5	14
Voltron	86M	196×768	31.94 ±1.2	5	24.83 ±1.0	9.5	22.72 ±1.0	15
Theia	86M	196×768	27.00 ±1.3	9	24.83 ±1.1	9.5	43.83 ±0.9	8
C-RADIOv4	631M	196×1,280	6.06 ±0.9	19	6.28 ±0.6	19	15.44 ±1.2	18
RADIOv2.5	98M	196×768	5.44 ±0.8	20	4.39 ±0.8	20	12.61 ±1.1	19
ViT	86M	196×768	14.67 ±1.1	18	13.67 ±1.2	15.5	18.61 ±1.0	17

few tasks: 23 of 24 task points fall below the diagonal, and the exception is a task on which both variants stay below 10% (Figure 3b). The comparison is controlled but narrow (two encoders, one regime, mean pooling), yet it shows that evaluations through one mean embedding probe a different interface from the one a patch-conditioned policy uses.

Finding 3: In RoboCasa single-task and RoboTwin multi-task fine-tuning, every encoder gains and the frozen order largely survives. All fourteen pairs gain, by 2.2 to 26.1 points on RoboTwin and 1.1 to 9.4 on RoboCasa. DINOv2 and SigLIP2, strong when frozen, gain little; SPA, DINOv3, and VC-1 gain 18 to 25 points on RoboTwin, and VGGT-Ω rises from 50.0% to 76.1%. Fine-tuned success follows frozen success at $\rho = 0.96$ on RoboTwin and 0.89 on RoboCasa (Figure 3c), whereas the size of the gain has no consistent relation with frozen success ($\rho = -0.14$ and $+0.50$). With seven pairs per benchmark both statements are exploratory; Appendix G.3 shows every pair.

More demonstrations raise success but rarely reorder the encoders. SPA and C-RADIOv4 stay below 11% at every RoboCasa budget, and the one clear change of order is SigLIP2 on RoboTwin, which starts behind DINOv3 and ends close to DINOv2 (Figure 4a, b). Shared policies are further from saturation: on tasks C1–C4, going from 500 to 1,000 demonstrations adds -0.2 to $+3.8$ points of single-task success but $+10.3$ to $+13.0$ points of multi-task success.

Model size helps some families and not others. DINOv2 improves from Small to Base, but Large and giant add little; DINOv3 Base is its family's best RoboCasa checkpoint, and its RoboTwin curve stays between 24.6% and 30.6% (Figure 4c, d). V-JEPA 2.1 rises with size on RoboCasa and its 1B model reaches the best RoboTwin point of the sweep (55.9%), whereas SPA and C-RADIOv4 stay low at every size, so no single scaling rule emerges.

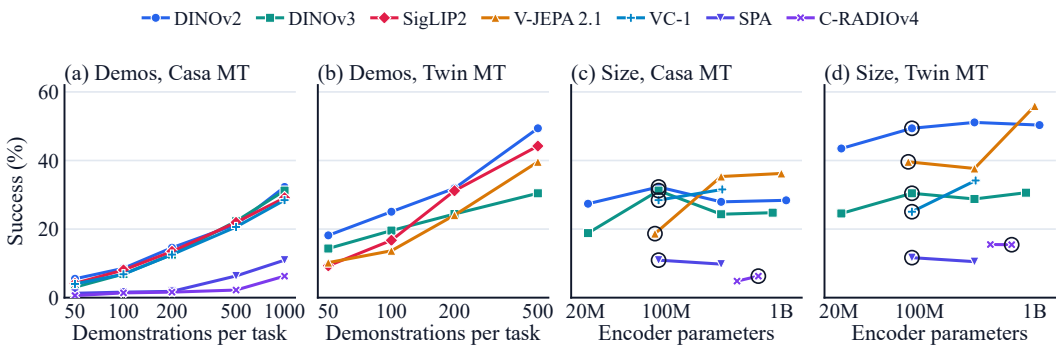


Figure 4: Demonstration and model scale under multi-task training: suite success against demonstrations per task (a, b) and encoder parameters (c, d); rings mark the default checkpoints of Table 2.

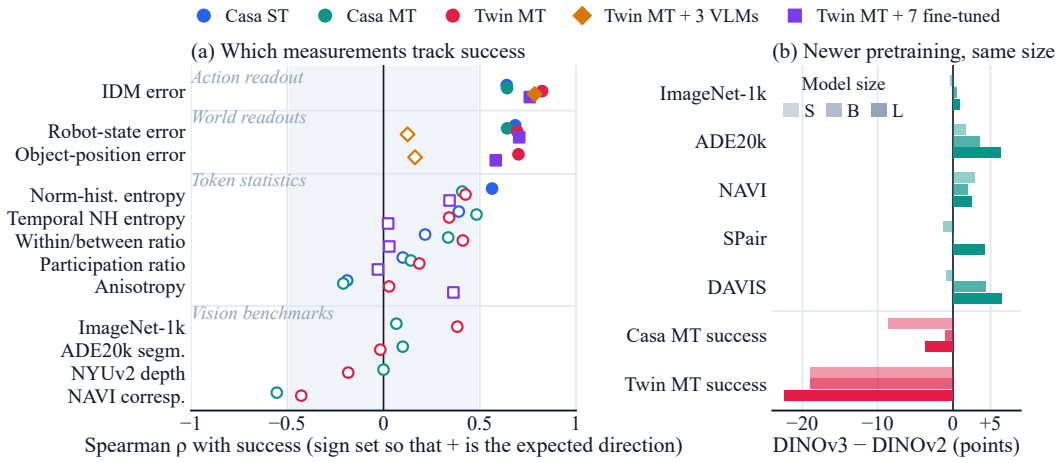


Figure 5: Which measurements track closed-loop success. (a) Spearman's ρ between each measurement and suite success, signed so that $+$ is the expected direction (lower error, higher benchmark score). Filled markers have a permutation $p < 0.05$; the shaded band, $|\rho| < 0.49$, is not significant for 17 encoders. Vision benchmarks cover the 9 to 11 checkpoints with published scores. Only the action readout stays strong in every setting. (b) DINOv3 minus DINOv2 at matched sizes: gains on ADE20k and NAVI at every size, yet lower success at every size.

The VLM layer and task sharing change success unevenly. Reading VLM layer 16 instead of the final layer raises all four RoboCasa configurations by 6.0 to 10.7 points but changes RoboTwin by less than one. Under the fixed per-policy budget, a shared policy scores lower than separate policies for 17 of 20 encoders on RoboCasa.

5 What Tracks Closed-Loop Success?

Prior work ties the value of an encoder for control to how well it reconstructs the world: segmentation, depth, correspondence, or the state of the robot and its objects (Burns et al., 2023; Zhu et al., 2025; Dong et al., 2026; Hu et al., 2023). We test such measures against an action readout, label-free token statistics, and published vision benchmarks, and ask whether each association survives a change of setting. Figure 5 summarizes the answer, and Table 13 lists all diagnostics.

5.1 Reading the Action Beats Reconstructing the World

Finding 4: The action readout is the only measurement that tracks success in every setting we test. IDM error goes with lower success on Casa ST ($\rho = -0.64$, 95% interval $[-0.90, -0.14]$), Casa MT (-0.64, $[-0.93, -0.13]$), and Twin MT (-0.82, $[-0.94, -0.51]$), and it keeps this association when the VLMs join ($p \leq 0.015$) and when fine-tuned checkpoints are pooled in (-0.76), the only measurement significant in all seven settings of Table 3. It survives leaving out any encoder

Table 3: Readout associations across settings. Spearman's ρ between probe error and suite success; **bold** marks a permutation $p < 0.05$ and grey a weaker association. "+VLMs" adds the three Qwen3-VL-derived VLMs to the 17 vision encoders; "+FT" adds the seven fine-tuned RoboTwin checkpoints; "Frozen→FT" correlates a frozen encoder's error with the success of its fine-tuned version; the last column counts pairs whose error falls with fine-tuning.

	17 vision encoders			+VLMs (20)			+FT	Frozen→FT		Falls
Readout	Casa ST	Casa MT	Twin MT	Casa ST	Casa MT	Twin MT	Twin	Twin	Casa	**w/ FT**
Action readout										
IDM error	**−0.64**	**−0.64**	**−0.82**	**−0.54**	**−0.67**	**−0.78**	**−0.76**	−0.64	−0.61	7/7
World readouts										
State error	**−0.68**	**−0.64**	**−0.69**	**−0.53**	**−0.58**	−0.12	**−0.71**	−0.32	−0.54	7/7
Object error	–	–	**−0.70**	–	–	−0.16	**−0.58**	**−0.93**	–	2/7

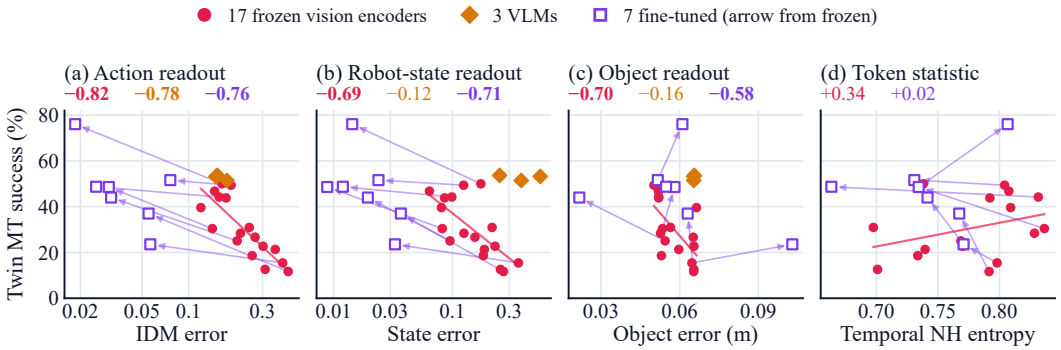


Figure 6: What happens to each association when the encoders change on RoboTwin multi-task. Numbers above each panel give Spearman's ρ for the 17 vision encoders (line: fit), with the VLMs, and with the fine-tuned checkpoints (bold: $p < 0.05$); arrows join each frozen encoder to its fine-tuned version. Action and state errors fall along the frozen trend in all seven pairs, whereas object error and temporal norm-histogram entropy mostly move against it.

($|\rho| \geq 0.57$) or family ($|\rho| \geq 0.49$), and takes the expected sign in all 36 task-level correlations (median 0.60), most strongly on the tasks where encoders differ most. The third of encoders with the highest IDM error averages 16 to 26 points less success than the best third. The associations are nominal on RoboCasa ($q = 0.16$–0.17) and robust on RoboTwin ($q \leq 0.005$).

Finding 5: World readouts match the action readout among vision encoders but break down across encoder types and during adaptation. Among vision encoders, state error tracks success as well as IDM error, and the two correlate at 0.92 and 0.90: the IDM target $a_{t+15} - s_t$ contains the current state, which the policies must read from images without proprioception. The readouts part ways where encoders differ in kind. The VLMs lead RoboTwin, yet rank 15th, 19th, and 20th of 20 in state error and 4th, 5th, and 9th in IDM error, so adding them erases the state (-0.12) and object (-0.16) associations but not the IDM one (-0.78; Figure 6a–c). Fine-tuning lowers IDM and state errors in all seven pairs, moving each checkpoint along the frozen trend, while object error rises in five of them; frozen IDM error also predicts fine-tuned success on both benchmarks (-0.64, -0.61; exploratory). A patch-conditioned policy thus needs to read how the robot should move more than where things are.

The action readout separates weak encoders better than it orders strong ones. The five encoders with the lowest IDM error exceed the average by 4 to 12 points and the five with the highest fall 11 to 14 points below it (Figure 17), yet V-JEPA 2.1 has the lowest RoboCasa IDM error and ranks only 9th and 11th in success. Fine-tuned checkpoints all reach IDM errors of 0.018 to 0.076 while their success spans 24% to 76%, so the frozen probe value is the one to use for selection.

5.2 Better vision is not better control

Finding 6: Segmentation, depth, and correspondence scores do not track success, and their gains from scale or newer pretraining do not carry over. We use published scores under one

protocol per benchmark, linear probes on ImageNet-1k, ADE20k, VOC, and NYUv2 (Fu et al., 2026) and NAVI, SPair, and DAVIS (Siméoni et al., 2025; Ranzinger et al., 2026), for the 9 to 11 checkpoints that have them. None correlates with multi-task success at $p < 0.05$: ImageNet-1k is at $+0.07$ and $+0.38$, segmentation and depth between -0.43 and $+0.10$, and correspondence and video between -0.27 and -0.55 (Figure 5a; Table 17). From DINOv2 Small to giant, ImageNet-1k rises from 80.8% to 87.0% and NAVI from 53.4 to 60.1, while success stops rising after Base on RoboCasa and after Large on RoboTwin. DINOv3 beats DINOv2 at every matched size on ADE20k (by 1.7 to 6.4 mIoU) and NAVI, yet trails it by 19 to 22 points on RoboTwin (Figure 5b), and C-RADIOv4 pairs near-top correspondence scores with the lowest success. With few families these results are exploratory, but none favors scene understanding as a proxy for control.

5.3 Label-free token statistics are weak screens

Finding 7: Norm-distribution statistics keep their direction across frozen encoders, stay below the action readout, and fade under fine-tuning. Norm-histogram entropy is positive on the three frozen cohorts ($+0.56$, $+0.41$, $+0.43$), temporal norm-histogram entropy too ($+0.39$ to $+0.48$), and the within/between-frame variance ratio negative (-0.22 to -0.41), yet none reaches the range of IDM error (the paired bootstrap advantage of IDM is $+0.08$ to $+0.40$ in $|\rho|$). Adding the seven fine-tuned checkpoints weakens all three to $+0.34$, $+0.02$, and -0.03, while IDM error stays at -0.76 (Figure 5a): temporal entropy falls in six of seven pairs as success rises (Figure 6d), and the participation ratio, or effective rank (Garrido et al., 2023), converges from 3.2–23.6 to 5.7–7.7. Across frozen encoders the effective rank reaches only $+0.10$ to $+0.19$, and anisotropy flips sign between cohorts.

5.4 Combining diagnostics

Finding 8: No combination improves on the action readout consistently across cohorts. In leave-one-encoder-out regression, IDM error alone lowers the error of the cohort mean by 25% to 43%; a second diagnostic helps only on Casa ST and slightly, and five diagnostics are worse everywhere (Table 14). Choosing the best single diagnostic on two cohorts and testing it on the third picks a readout probe every time (mean held-out $\rho = 0.66$); the best pair gives 0.64.

Which diagnostics matter depends on the benchmark more than on the training regime. Over all diagnostics, the profile of correlations with success agrees at Pearson $r = 0.94$ between RoboCasa single-task and multi-task but only at 0.44 between RoboCasa and RoboTwin (Figure 23); IDM error, state error, both norm-histogram entropies, and the within/between ratio keep their sign in both comparisons.

6 Discussion and Conclusion

Implications. For patch-conditioned egocentric policies, the question to ask of an encoder is less how well it reconstructs the scene than whether the robot's motion can be read from its tokens, and the interface through which it is read, from pooling to the extraction layer, is part of the answer. To choose a frozen encoder, fix the interface, set aside weak candidates with an IDM probe on head-camera frames of the target data, and run the shortlist in closed loop; after fine-tuning the frozen order largely survives, but each gain has to be measured.

Limitations. The evidence comes from two simulated suites, 17 to 20 related encoders, one action head trained with one seed and evaluated at its final checkpoint, and policies without proprioception; the three VLMs share one base model, and the vision-benchmark scores come from published evaluations. Without single-frame controls, part of the IDM signal may come from the state (Appendix J).

Conclusion. HIVE-Bench evaluates dense patch-level representations for egocentric bimanual manipulation. Keeping separate patch tokens matters; an action readout tracks success across every setting we test, whereas world-reconstruction measures and better vision benchmarks do not carry over; and fine-tuning helps every encoder while largely keeping the frozen order. We hope HIVE-Bench makes such comparisons routine for new encoders.

Ethics statement

This work uses simulated robot environments, publicly released demonstration data, and publicly released pretrained models under their respective licenses. It involves no human subjects, personal data, or real-robot deployment. Encoder rankings describe a specific policy interface and set of simulated tasks and should not be read as general quality judgments of the evaluated models.

Reproducibility statement

Section 3 summarizes the benchmarks, encoders, policy, and diagnostics. Appendix A and Appendices B and C give the task sets, every checkpoint with its preprocessing and token interface, the policy equations, and all training and evaluation settings; Appendix D defines every diagnostic and probe, Appendix E lists the encoders behind every analysis, and Appendix F gives the statistical procedures, including permutation seeds. Appendix K reports every per-task success rate and every diagnostic value. The anonymized supplementary material contains the policy and diagnostic code, the configuration files, the result export, and the analysis scripts that regenerate every table and figure of the paper from that export.

AI use statement

In this work, we used generative AI tools (large language models) for three kinds of tasks. First, to draft and edit sections of the paper and to improve its readability, including restructuring text from an earlier draft into this format. Second, to search for, summarize, and identify related literature and to format references. Third, under the authors' direction, to write and debug the analysis and plotting scripts that compute the reported statistics and draw the figures from our result export, and to suggest robustness checks such as the leave-one-family-out correlations. The research question, the benchmark design, the choice of encoders, and all training runs, rollouts, and probe measurements were designed and carried out by the authors; we did not use generative AI tools to generate data, to formulate mathematical claims or proofs, or to translate. The authors decided which analyses to run and report, checked every AI-assisted analysis and every number in the text against the result export and the scripts' outputs, and verified every cited work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Atiksh Bhardwaj, Edward Weiyi Duan, Prithwish Dan, Wei-Chiu Ma, and Preston Culbertson. Decoding task progress from VLA representations. *arXiv preprint arXiv:2608.13474*, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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

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

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023b.

Kaylee Burns, Zach Witzel, Jubayer Ibn Hamid, Tianhe Yu, Chelsea Finn, and Karol Hausman. What makes pre-trained visual representations successful for robust manipulation? *arXiv preprint arXiv:2312.12444*, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Alexandre Chapin, Bruno Machado, Emmanuel Dellandréa, and Liming Chen. Object-centric representations improve policy generalization in robot manipulation. *arXiv preprint arXiv:2505.11563*, 2025.

Ryan Chen, Ziteng Pang, and Bradly C Stadie. Thoughts and lessons on using visual foundation models for manipulation. *Transactions on Machine Learning Research*, 2025a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607. PMLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649, 2021.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Robotics: Science and Systems*, 2024.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024.

Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1183–1198, 2023.

Jiahua Dong, Yunze Man, Pavel Tokmakov, and Yu-Xiong Wang. Capturing visual environment structure correlates with control performance. In *International Conference on Learning Representations*, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Zelin Fu, Bin Tan, Changjiang Sun, Shaohui Liu, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, and Nan Xue. Vision pretraining for dense spatial perception. *arXiv preprint arXiv:2607.05247*, 2026.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning*. PMLR, 2024.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning*, pp. 10929–10974. PMLR, 2023.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15180–15190, 2023.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. RVT: Robotic view transformer for 3D object manipulation. In *Conference on Robot Learning*, pp. 694–710. PMLR, 2023.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19383–19400, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

Gunshi Gupta, Karmesh Yadav, Yarin Gal, Dhruv Batra, Zsolt Kira, Cong Lu, and Tim GJ Rudner. Pre-trained text-to-image diffusion models are versatile representation learners for control. *Advances in Neural Information Processing Systems*, 37, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. RADIOv2.5: Improved baselines for agglomerative vision foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Kyle Hsu, Tyler Ga Wei Lum, Ruohan Gao, Shixiang Shane Gu, Jiajun Wu, and Chelsea Finn. What makes certain pre-trained visual representations better for robotic learning? In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

Yingdong Hu, Renhao Wang, Li Erran Li, and Yang Gao. For pre-trained vision models in motor control, not all policy learning methods are created equal. In *International Conference on Machine Learning*, pp. 13628–13651. PMLR, 2023.

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

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pp. 4904–4916. PMLR, 2021.

Guangqi Jiang, Yifei Sun, Tao Huang, Huanyu Li, Yongyuan Liang, and Huazhe Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets. In *International Conference on Learning Representations*, 2025a.

Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. DexMimicGen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025b.

Ya Jing, Xuelin Zhu, Xingbin Liu, Qie Sima, Taozheng Yang, Yunhai Feng, and Tao Kong. Exploring visual pre-training for robot manipulation: Datasets, models and methods. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems*, 2023.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3D diffuser actor: Policy diffusion with 3D scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning*. PMLR, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision*, pp. 71–91. Springer, 2024.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pp. 19730–19742. PMLR, 2023.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning*. PMLR, 2024.

Qiwei Liang, Boyang Cai, Minghao Lai, Sitong Zhuang, Tao Lin, Yan Qin, Yixuan Ye, Jiaming Liang, and Renjing Xu. Bootstrap dynamic-aware 3D visual representation for scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13419–13429, 2026.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Yihan Lin, Jiawei He, Shifeng Bao, Chen Zhao, Yang Li, Xiaobo Wang, Yan Wang, Cheng Chi, and Jing Zhang. JEPA-WAM: Learning vision-language-action policies with joint-embedding world modeling. *arXiv preprint arXiv:2608.09381*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36: 655–677, 2023.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, pp. 1678–1690. PMLR, 2021.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421. Springer, 2020.

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

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pp. 892–909. PMLR, 2022.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024.

Nilaksh, Saurav Jha, Artem Zholus, and Sarath Chandar. Reconstruction or semantics? What makes a latent space useful for robotic world models. *arXiv preprint arXiv:2605.06388*, 2026.

NVIDIA. RADIO: Reduce all domains into one. https://github.com/NVlabs/RADIO, 2026. GitHub repository.

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

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, pp. 17359–17371. PMLR, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Baoqi Pei, Guo Chen, Jilan Xu, Yuping He, Yicheng Liu, et al. EgoVideo: Exploring egocentric foundation model and downstream adaptation. *arXiv preprint arXiv:2406.18070*, 2024.

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

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017b.

Han Qi, Haocheng Yin, and Heng Yang. Control-oriented clustering of visual latent representation. In *International Conference on Learning Representations*, 2025.

Jianing Qian, Yunshuang Li, Bernadette Bucher, and Dinesh Jayaraman. Task-oriented hierarchical object decomposition for visuomotor control. *arXiv preprint arXiv:2411.01284*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pp. 416–426. PMLR, 2022.

Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. AM-RADIO: Agglomerative vision foundation model reduce all domains into one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12490–12500, 2024.

Mike Ranzinger, Greg Heinrich, Collin McCarthy, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. C-RADIOv4 (tech report). *arXiv preprint arXiv:2601.17237*, 2026.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025.

Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. In *Conference on Robot Learning*. PMLR, 2024.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pp. 785–799. PMLR, 2022.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8922–8931, 2021.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pp. 402–419. Springer, 2020.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in Neural Information Processing Systems*, 35:10078–10093, 2022.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Nikolaos Tsagkas, Andreas Sochopoulos, Duolikun Danier, Sethu Vijayakumar, Alexandros Kouris, Oisin Mac Aodha, and Chris Xiaoxuan Lu. Attentive feature aggregation or: How policies learn to stop worrying about robustness and attend to task-relevant visual cues. *arXiv preprint arXiv:2511.10762*, 2025.

Nikolaos Tsagkas, Andreas Sochopoulos, Duolikun Danier, Chris Xiaoxuan Lu, and Oisin Mac Aodha. The temporal trap: Entanglement in pre-trained visual representations for visuomotor policy learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

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

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. BridgeData V2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21486–21499, 2026a.

Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. RepWAM: World action modeling with representation visual-action tokenizers. *arXiv preprint arXiv:2606.13674*, 2026b.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14549–14560, 2023.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Advances in Neural Information Processing Systems*, 37:124420–124450, 2024a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024b.

Yixiao Wang, Mingxiao Huo, Zhixuan Liang, Yushi Du, Lingfeng Sun, Haotian Lin, Jinghuan Shang, Chensheng Peng, Mohit Bansal, Mingyu Ding, and Masayoshi Tomizuka. VER: Vision expert transformer for robot learning via foundation distillation and dynamic routing. In *International Conference on Learning Representations*, 2026c.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, et al. OpenWAM: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026d.

Xin Wen, Bingchen Zhao, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. A data-centric revisit of pre-trained vision models for robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16133–16142, 2023.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, et al. Xiaomi-Robotics-1: Scaling vision-language-action models with over 100K hours of real-world trajectories. *arXiv preprint arXiv:2607.15330*, 2026.

Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Jingyun Yang, Zi-ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. EquiBot: SIM(3)-equivariant diffusion policy for generalizable and data efficient learning. In *Conference on Robot Learning*. PMLR, 2024a.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10371–10381, 2024b.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024c.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Hanxun Yu, Xuan Qu, Yuxin Wang, Jianke Zhu, and Lei Ke. Unlocking dense metric depth estimation in VLMs. *arXiv preprint arXiv:2605.15876*, 2026.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pp. 1094–1100. PMLR, 2020.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Robotics: Science and Systems*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

Long Zhao, Nitesh B Gundavarapu, Liangzhe Yuan, Hao Zhou, Shen Yan, Jennifer J Sun, Luke Friedman, Rui Qian, Tobias Weyand, Yue Zhao, et al. VideoPrism: A foundational visual encoder for video understanding. In *International Conference on Machine Learning*. PMLR, 2024.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

Gaoyue Zhou, Zichen Jeff Cui, Ada Langford, Bowen Tan, Yann LeCun, and Lerrel Pinto. Patch policy: Efficient embodied control via dense visual representations. *arXiv preprint arXiv:2607.18236*, 2026a.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

Yijing Zhou, Qiwei Liang, Sitong Zhuang, Jiaxi Li, Xianpeng Wang, Boyang Cai, Yunyang Mo, and Renjing Xu. Action-effect memory pretraining for robot manipulation. *arXiv preprint arXiv:2606.12499*, 2026b.

Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. *arXiv preprint arXiv:2402.02500*, 2024.

Haoyi Zhu, Honghui Yang, Yating Wang, Jiange Yang, Limin Wang, and Tong He. SPA: 3D spatial-awareness enables effective embodied representation. In *International Conference on Learning Representations*, 2025.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

APPENDIX

The appendix gives the details behind every result in the main text. Reviewers who want the full protocol should start with Appendices A–C; those who want the evidence behind a numbered finding should go to Appendices G and H, where each finding is expanded under its main-text number.

- Appendix A: benchmarks, coverage of every study, encoder families, extraction interfaces, the policy equations, training settings, and the diagnostic definitions used in the main text.
- Appendix B: pretraining data and objectives, checkpoints, preprocessing, and token interfaces of every encoder and size variant.
- Appendix C: optimization, the two conditioning interfaces, execution, and evaluation.
- Appendices D, E, and F: formulas of every diagnostic and probe, the encoders behind every analysis, and the statistical procedures.
- Appendix G: extended closed-loop results, including demonstration scale, model scale, VLM layers, and task sharing.
- Appendix H: extended diagnostic results, including the full association table, the VLM and selection analyses, token statistics, combinations, fine-tuning shifts, and per-task consistency.
- Appendix I: the distribution-shaping probe and the published vision-benchmark scores.
- Appendix J: extended discussion, hypotheses, and limitations.
- Appendix K: every per-task success rate and every diagnostic value.
- Appendix L: extended related work and a 23-study coverage comparison.

A BENCHMARK, ENCODER, AND POLICY DETAILS

This appendix expands Section 3. It describes what is fixed and what varies: the benchmarks and their coverage (Section A.1), the encoders and the interface through which the policy reads them (Section A.2), the action head (Section A.3), training and evaluation (Section A.4), and the diagnostics (Section A.5).

A.1 BENCHMARKS, INTERFACES, AND COVERAGE

We use 12 RoboCasa-GR1 kitchen tasks (Nasiriany et al., 2024; Bjorck et al., 2025) and 12 domain-randomized RoboTwin 2.0 tasks (Chen et al., 2025b). RoboCasa-GR1 combines object transfer with cabinet, drawer, and microwave interactions performed by a humanoid with dexterous hands; RoboTwin covers adjustment, tool use, handover, lifting, placement, rotation, and articulated objects performed by two arms with grippers. The two benchmarks differ in more than their cameras. Robot, action space, task set, demonstration budget, and visual randomization all change together, so comparisons between them are observational rather than controlled. Figure 2 shows sample scenes of both, Table 4 summarizes the interfaces, and Appendix K lists the tasks and every per-task result.

Table 4: Benchmark interfaces. Demonstrations are per task; RoboTwin actions are absolute joint positions. Both benchmarks condition on the current images only, without proprioception, and predict 16-step action chunks. The RoboCasa policy executes the first 12 actions of a chunk before replanning, and the RoboTwin policy executes all 16. Every diagnostic reads the head camera.

Benchmark	Embodiment	Demos	Policy views	Action	Proprio.	Flow steps
RoboCasa-GR1	Humanoid, hands	1,000	Head	29-D	None	15
RoboTwin 2.0	Two arms, grippers	500	Head + 2 wrists	14-D joints	None	10

Protocols and coverage. The design crosses single-task (ST) and multi-task (MT) learning with frozen and fine-tuned encoders. ST trains a separate policy per task; MT trains one language-conditioned policy on all twelve tasks. On RoboCasa we train all 20 encoders frozen in both regimes and fine-tune 7 of them single-task; on RoboTwin we train all 20 frozen and the same 7 fine-tuned, multi-task only. The data-scale study covers four encoders on RoboCasa single-task (tasks C1–C4), six on RoboCasa multi-task, and four on RoboTwin. The model-scale study covers six families on both benchmarks and adds 11 checkpoints, for 31 pretrained checkpoints in total. Two interface studies read layer 16 of the VLMs and replace the patch tokens of DINOv2 and

DINOv3 by their mean. Throughout the paper, "20 encoders" refers to the default checkpoints of Table 2, and Figure 9 shows the coverage per encoder.

A.2 Visual representations and extraction interfaces

We group the encoders into seven families by their dominant pretraining signal (Table 5): supervised recognition, self-supervision, geometry, robot-oriented learning, multi-teacher distillation, vision–language alignment, and VLM-derived features. Appendix B lists each checkpoint with its training data, amount, objective, native resolution, and token interface.

Table 5: Seven representation families and their primary references; Appendix B details every checkpoint.

Family	Encoders	Family	Encoders
Self-supervised	MAE , DINOv2 DINOv3 , V-JEPA 2.1	**Vision–language**	CLIP , SigLIP SigLIP2 , InternViT
Geometry	SPA , VGGT-Ω LingBot-Vision	**VLM-derived**	Qwen3-VL , DepthVLM Xiaomi-Robotics-1
Multi-teacher	Theia , RADIOv2.5 C-RADIOv4	**Robot-oriented** **Supervised**	VC-1 , Voltron ViT

What we compare is therefore an encoder *under a documented extraction interface*, not an encoder in isolation. Within a benchmark and recipe, the task data, demonstration subsets, camera streams, action representation, action head, optimizer, schedule, and evaluation rollouts are fixed. The token width and count, the preprocessing, the extracted layer, and the interface family follow the encoder. Views are stored at 224×224. Patch sizes 16 and 14 give 196 and 256 tokens per view, V-JEPA 2.1 reads 384-pixel images (576 tokens), and the VLMs produce 64 image tokens per view. Class and register tokens are removed. VGGT-Ω contributes aggregator features and LingBot-Vision its final normalized patch features. The three VLMs are Qwen3-VL and two checkpoints initialized from it; each contributes the hidden states at its image-token positions in the final layer, or in layer 16 where stated. These states do not depend on the instruction, and we do not use the action head of Xiaomi-Robotics-1. Views are encoded separately and concatenated, with two exceptions that mix information across views before the policy reads it: VGGT-Ω receives the RoboTwin head and wrist views together as one three-frame input, and the image tokens of a later view in a VLM attend to those of earlier views. Appendix B gives every checkpoint and preprocessing step.

A.3 A shared patch-conditioned action policy

The action head is a DiT-style flow-matching model (Peebles & Xie, 2023; Lipman et al., 2023) derived from GR00T (Bjorck et al., 2025) (Figure 7). It predicts a 16-step action chunk from the current images, together with the instruction in the multi-task case. No policy receives proprioceptive input, so whatever a policy knows about the robot's own configuration must come from the images, including the positions of its arms and hands.

Context. Let $I_t^{(v)}$ denote camera view v at time t and ℓ the instruction. The encoder f_ϕ returns N_v patch tokens of width D_{enc} per view. Single-task policies use their concatenation as context; multi-task policies append projected token-level features of a frozen CLIP text encoder g:

$$C_t^{\text{ST}} = \text{concat}_{v=1}^{V} f_\phi(I_t^{(v)}), \qquad C_t^{\text{MT}} = \left[C_t^{\text{ST}}; P_\ell g(\ell)\right]. \tag{2}$$

The context width is $D_{\text{ctx}} = D_{\text{enc}}$ for vision encoders. The VLMs and LingBot-Vision use the same form after their image tokens and CLIP ViT-B/16 text tokens are linearly projected to $D_{\text{ctx}} = 1{,}024$ (Appendix C). The context is neither pooled nor normalized.

How the tokens reach the action head. The head has a fixed internal width $D_{\text{DiT}} = 768$ (12 heads of dimension 64) and 16 blocks. It operates on a query sequence $H_t = [Q; e(A_\tau)]$ of 32 learned tokens Q and 16 noisy action tokens; the action tokens are embedded together with the discretized flow time and receive learned position embeddings. The eight even-numbered

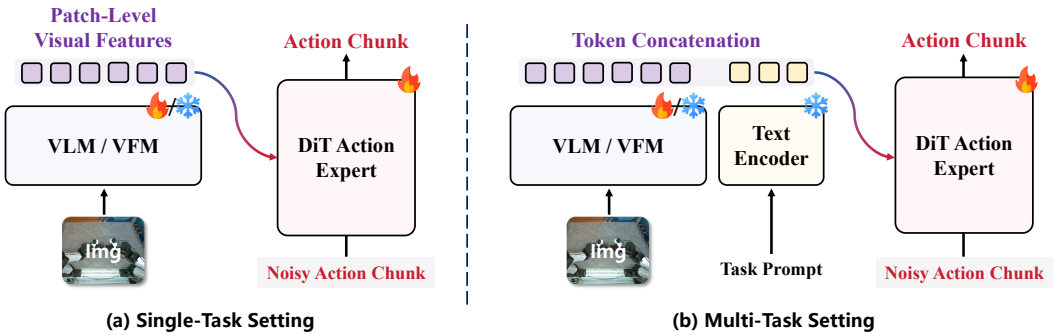


Figure 7: The shared policy. (a) Single-task: the patch tokens of a frozen or fine-tuned encoder, a vision foundation model (VFM) or a VLM, condition a DiT action expert that denoises a 16-step action chunk. (b) Multi-task: the patch tokens are concatenated with the tokens of a frozen text encoder before they enter the action expert.

blocks cross-attend from H_t to the context,

$$\mathrm{CrossAttn}(H_t, C_t) = \mathrm{softmax}\Big(\frac{(H_t W_Q)(C_t W_K)^\top}{\sqrt{d_h}}\Big) C_t W_V, \tag{3}$$

where $W_K, W_V \in \mathbb{R}^{D_{\mathrm{ctx}} \times 768}$. The eight odd-numbered blocks apply self-attention within H_t. Adaptive layer normalization conditions every block on the flow time, and a two-layer decoder whose hidden width equals D_{ctx} maps the action tokens of the last block to the velocity field. Vision thus enters the policy only through Eq. (3). The attention logits depend on the projected keys $W_K^\top z_i$, not on the raw token norms alone, so a token with a large norm need not receive a large weight. The mean-pooling control of Section G.2 keeps this head unchanged and replaces each view's patch tokens by their mean.

Training objective. For a normalized action chunk $A_t \in \mathbb{R}^{16 \times d_a}$, sample $\epsilon \sim \mathcal{N}(0, I)$ and $b \sim \mathrm{Beta}(1.5, 1)$, set $\tau = (s - \min(b, s))/s$ with $s = 0.999$, and interpolate $A_\tau = (1-\tau)\epsilon + \tau A_t$. The head predicts the velocity toward the target:

$$\mathcal{L}_{\mathrm{act}}(\theta, \phi) = \mathbb{E}\left[\frac{1}{16 d_a} \left\| v_\theta(A_\tau, \lfloor 1000\tau \rfloor, C_t) - (A_t - \epsilon) \right\|_F^2\right]. \tag{4}$$

Each example receives eight noise and time draws per step. Frozen training updates the head and projections; fine-tuning also updates the visual encoder. The text encoder stays frozen.

Inference. Starting from $A^{(0)} \sim \mathcal{N}(0, I)$, the predicted velocity is integrated with K Euler steps,

$$A^{(k+1)} = A^{(k)} + \frac{1}{K} v_\theta\left(A^{(k)}, \left\lfloor \frac{1000k}{K} \right\rfloor, C_t\right), \qquad k = 0, \ldots, K-1, \tag{5}$$

with $K = 15$ on RoboCasa-GR1 and $K = 10$ on RoboTwin. The chunk is denormalized, executed, and the policy replans from new observations (Appendix C).

A.4 Training and Evaluation Settings

Optimization. Every policy is trained for 100k optimizer steps with a global batch of 128 (32 on each of four GPUs). The budget is equal *per policy*: a single-task policy spends it on one task, a multi-task policy on twelve, so each task receives less exposure under task sharing. We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, and weight decay 10^{-8}, gradient clipping at 1, mixed precision, and gradient checkpointing. The learning rate is 10^{-4} with warmup and cosine decay to the floor in Table 6; in RoboCasa multi-task runs, the text projector and the remaining non-head parameters use 3×10^{-5} while the action head keeps 10^{-4} (Appendix C). Every configuration is trained once, with one training seed.

Evaluation and reported uncertainty. The trained policy is evaluated with three evaluation seeds, each rolling out every task 50 times. A task score is the mean over the three evaluation seeds,

Table 6: Training settings. ST and MT denote single-task and multi-task policies on the vision-encoder interface; VLM denotes the interface of the three VLMs and LingBot-Vision, whose task regime follows the selected task set. All rows use 100k steps and dropout 0.2. The last column is the context width D_{ctx}; the DiT width is 768 throughout.

Benchmark	Interface	Warmup	LR floor	D_{ctx}
RoboCasa-GR1	ST	3,000	5×10^{-8}	D_{enc}
	MT	3,000	5×10^{-8}	D_{enc}
	VLM	3,000	5×10^{-8}	1,024
RoboTwin 2.0	MT	5,000	5×10^{-7}	D_{enc}
	VLM	5,000	5×10^{-7}	1,024

and its uncertainty σ_k is the standard deviation across them; it measures evaluation noise, not the variation between training runs. A suite score is the unweighted mean of the twelve task scores. Per-seed suite scores are not part of the result export, so the $\pm$ attached to a suite score is the propagated value $\sqrt{\sum_k \sigma_k^2}/12$, which assumes independent task errors; it is an approximation, not the standard deviation of the suite mean across seeds.

A.5 Diagnostics

The diagnostics fall into four families that differ in what they need. *Token statistics* need only frames and describe how norms, variance, and spatial frequencies are distributed over patches and time. *Readout probes* need labels or actions and train a small network on frozen tokens to recover the robot's state, an object's position, or a command displacement between two frames. A *trained-policy diagnostic* requires a trained action policy. A *distribution-shaping probe* needs only frames and fits a weak adapter that reshapes the tokens; its results are exploratory and appear in Appendix I.1. We then ask how diagnostics combine to predict, rank, or select encoders, and repeat them after fine-tuning. All diagnostics read the head camera, also on RoboTwin, where the policy additionally sees two wrist views. The definitions that matter for the results are the following; Appendix D gives every formula and a metric dictionary.

- **Norm-histogram entropy.** The patch-token norms of a frame are binned into 30 equal-width bins spanning that frame's own range, and the entropy of the resulting histogram is computed. Because the bins adapt to each frame, the value depends on how the norms spread over their range: identical norms occupy one bin, and a few very large norms push all others into the lowest bins, so both cases give low entropy.
- **Inverse-dynamics (IDM) error.** A probe receives the frozen tokens of two head-camera frames 16 steps apart and predicts the standardized terminal command displacement $a_{t+15} - s_t$, the difference between the last commanded action of the window and the current state. We keep the name IDM for this probe, but its target is one displacement vector, not the 16-step action sequence, the executed motion, or contact dynamics. The current state s_t is visible in the first frame, so part of the target is readable from one frame.
- **State and object probes.** The state probe decodes the robot's joint state from one head-camera frame. The object probe decodes the camera-frame 3D position of the task object from the first head-camera frame of four RoboTwin tasks and reports the mean absolute error over the three coordinates; the export contains no mean-predictor baseline for it.
- **Action sensitivity to texture.** The variance of a trained RoboTwin policy's predicted chunk across 100 background textures of the same initial scene. It characterizes the policy trained on an encoder, not the encoder alone, and is measured only on RoboTwin.

The probes differ in capacity (Table 11). The IDM probe works at width 256 (2.4–2.8M trainable parameters), whereas the state and object probes work at the width of the tokens they read (19.5M parameters at 768 and up to 217M for the 2,560-wide VLM states). Each probe is trained on 1,000 (RoboCasa) or 400 (RoboTwin) demonstration episodes and evaluated on 500 or 100 held-out episodes; the same split selects the best epoch and yields the reported error, so these are validation errors. Token statistics are computed on the first 100 steps of every fifth demonstration of each task, at a stride of 5.

B Encoder Inventory and Extraction Interfaces

This appendix documents every visual representation used in the paper: what it was trained on, how much data it saw, which objective shaped it, and exactly how the policy reads it. Figure 8 shows the training paradigm usual for each family. Table 7 covers pretraining, Table 8 the checkpoints and token interfaces of the default models, and Table 9 the additional sizes of the model-scale study. Pretraining facts follow the original papers and model cards; interface facts follow the released HIVE-Bench code.

B.1 Usual training paradigms

Figure 8 draws the objective that typically trains each family in Table 5. A flame marks a module updated by the loss named in that panel, and a snowflake marks a module held fixed. The panels describe the family's dominant pattern. Table 7 records the data and the objective of each checkpoint.

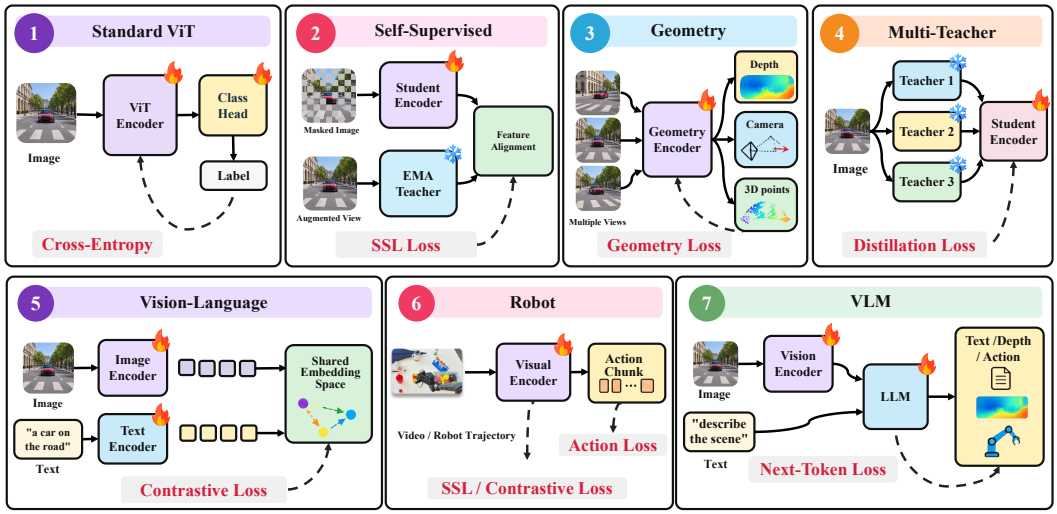


Figure 8: Usual training paradigms of the seven encoder families. A flame marks a module trained by the loss named in the panel, and a snowflake marks a module held fixed.

Supervised. Panel 1 (Standard ViT) feeds an image through a ViT and a class head and trains both with cross-entropy against a category label. The suite uses the ImageNet-supervised ViT.

Self-supervised. Panel 2 feeds a masked image to a student encoder and an augmented view to an exponential-moving-average teacher. An alignment loss trains the student, and the teacher stays fixed. DINOv2, DINOv3, and V-JEPA 2.1 follow this student–teacher pattern. MAE trains the same ViT by reconstructing masked pixels, with no teacher.

Geometry. Panel 3 sends several views of one scene through one encoder and trains it to recover depth, camera pose, and 3D points. SPA and VGGT-Ω are trained for this geometric readout. LingBot-Vision keeps a ViT trained by masked self-distillation.

Multi-teacher. Panel 4 freezes several teachers on the same image and trains a student encoder to match their features by distillation. Theia distills CLIP, DINOv2, and a supervised ViT. RADIOv2.5 and C-RADIOv4 distill larger vision–language, self-supervised, and segmentation teachers.

Vision–language. Panel 5 trains an image encoder and a text encoder so that a picture and its caption meet in one embedding space under a contrastive loss. CLIP, SigLIP, and SigLIP2 use this objective. InternViT is distilled from a larger vision transformer and then trained by next-token prediction through an LLM.

Robot-oriented. Panel 6 reads a video or a robot trajectory with a visual encoder. The usual objective pairs a self-supervised or contrastive loss with a loss on a predicted action chunk. VC-1

and Voltron use the self-supervised half. VC-1 is masked autoencoding on egocentric, manipulation, and navigation video. Voltron uses language-conditioned masked autoencoding on Something-Something-v2 clips with captions.

VLM-derived. Panel 7 passes image tokens from a vision encoder into an LLM, together with a text prompt, and trains both with next-token prediction. The same generative interface can emit text, depth, or actions. Qwen3-VL is trained as that language model. DepthVLM adds metric-depth supervision, and Xiaomi-Robotics-1 adds a flow-matching action head.

Table 7: Pretraining data, amount, and objective. "Native" is the resolution at which the released checkpoint was trained; "multi-res." marks training over several resolutions. Encoder names are colored by family as in Table 2.

Encoder	Pretraining data and amount	Objective	Native
VGGT-Ω	About 4M labeled multi-view sequences (about 3M from 30+ public and internal datasets, 0.8M annotated Internet videos) and 18M unlabeled videos	Camera, depth, and point-map prediction with one dense head, then teacher–student self-distillation on unlabeled video; DINOv3-initialized ViT	512
LingBot-Vision	About 161M images curated from a pool of about 2B web images; the ViT-L is distilled from a 1.1B ViT-g/16 teacher	Masked boundary modeling on DINO/iBOT-style self-distillation, with Gram anchoring and high-resolution adaptation	Multi-res.
SPA	3.8M frames from ScanNet, ScanNet++, Hypersim, ADT, S3DIS, and DROID	Masked (50%) multi-view volume rendering of RGB, depth, and semantics; initialized from ImageNet MAE	224
DINOv2	LVD-142M, 142M curated images; Base distilled from ViT-g/14	DINO and iBOT self-distillation with KoLeo regularization	224, 518
DINOv3	LVD-1689M, about 1.7B curated images; Base distilled from a 7B teacher	DINO and iBOT self-distillation; teacher trained with Gram anchoring and high-resolution adaptation	Multi-res.
MAE	ImageNet-1K, 1.28M images	Masked (75%) pixel reconstruction	224
V-JEPA 2.1	VisionMix-163M: LVD-142M images with SSv2 (168K), Kinetics (733K), HowTo100M (1.1M), and YT-1B (19M) videos; Base distilled from a 2B ViT-G	Masked latent prediction with a dense loss on visible and masked tokens and deep self-supervision	384
Xiaomi-Robotics-1	Qwen3-VL initialization; over 100K hours of real-world manipulation trajectories for pretraining and over 10K hours of cross-embodiment data for post-training	Vision–language–action training of a mixture-of-transformers VLM with a flow-matching action DiT	Dynamic
DepthVLM	Qwen3-VL-4B initialization; 4.4M samples from eight metric-depth datasets (ScanNet++, Taskonomy, HM3D, Matterport3D, Argoverse2, Waymo, DDAD, nuScenes)	Language and dense metric-depth losses through a 34M DPT head; head-only, then end-to-end training	Dynamic
Qwen3-VL	About 2.2T multimodal pretraining tokens (67B alignment, two stages of about 1T, 100B long context), then instruction post-training	Multimodal next-token prediction; SigLIP2-Large vision encoder with DeepStack injection into the LLM	Dynamic
SigLIP2	WebLI, 10B images with 12B alt-texts in 109 languages	Sigmoid image–text loss with a captioning decoder, self-distillation, and masked prediction	224
InternViT	Distilled from InternViT-6B, then trained inside InternVL; the V2.5 release adds incremental training on OCR, chart, math, and document data	Cosine feature distillation, then next-token prediction through an LLM	448, dynamic
SigLIP	English image–text pairs of WebLI	Sigmoid image–text loss	224
CLIP	WIT, 400M image–text pairs	Contrastive image–text loss	224
VC-1	Ego4D, manipulation and navigation video, and ImageNet; 5.6M frames	Masked autoencoding	224
Voltron	Something-Something-v2 with captions, about 168K clips at five frames per clip and epoch	Language-conditioned masked autoencoding of single frames (V-Cond)	224
Theia	ImageNet-1K, 1.28M images; trained from scratch	Feature distillation from CLIP, DINOv2, and ViT through shallow CNN translators	224
C-RADIOv4	About 700M images with commercially permissive licenses	Multi-teacher distillation from SigLIP2-g, DINOv3-7B, and SAM3	Multi-res.
RADIOv2.5	DataComp-1B images	Multi-teacher distillation from DFN CLIP, SigLIP, DINOv2-g, and SAM-H	Multi-res.
ViT	ImageNet-21K (14M images, 21,843 classes), then ImageNet-1K	Supervised classification	224

Preprocessing. Every view is stored and loaded at 224×224 and then passed through the encoder's own preprocessing. Hugging Face image processors are used for DINOv2, DINOv3, CLIP, SigLIP, SigLIP2, MAE, and ViT. DINOv2's processor resizes the short side to 256 and center-crops 224, so its policy sees the central 87.5% of each view; CLIP resizes and center-crops to 224, which leaves a 224×224 view unchanged; the others resize to 224×224. DINOv2, DINOv3, and MAE use ImageNet statistics, CLIP its own, and SigLIP, SigLIP2, and ViT a mean and standard deviation of 0.5. InternViT and VC-1 resize to 224×224 with ImageNet statistics, and V-JEPA 2.1 resizes

Table 8: Checkpoints and token interfaces of the default encoders. Input is the resolution at which the encoder sees each view, N the number of patch tokens per view, and D the token width. All data are stored at 224×224; the preprocessing that produces each input is described in the text.

Encoder	Checkpoint	Backbone	Params	Input	N	D
VGGT-Ω	`facebook/VGGT-Omega` (`vggt_omega_1b_512`)	Aggregator, patch 16	1B	224^2	196	2,048[a]
LingBot-Vision	`robbyant/lingbot-vision-vit-large`	ViT-L/16	300M	224^2	196	1,024[b]
SPA	`HaoyiZhu/SPA` (`spa_vit_base_patch16`)	ViT-B/16	86M	224^2	196	768
DINOv2	`facebook/dinov2-base`	ViT-B/14	86M	224^2 crop	256	768
DINOv3	`facebook/dinov3-vitb16-pretrain-lvd1689m`	ViT-B/16, 4 registers	86M	224^2	196	768
MAE	`facebook/vit-mae-base`	ViT-B/16	86M	224^2	196	768
V-JEPA 2.1	`vjepa2_1_vit_base_384` (torch hub)	ViT-B/16	80M	384^2	576	768
Xiaomi-Robotics-1	`XiaomiRobotics/Xiaomi-Robotics-1-5B`, VLM part	Qwen3-VL	4.4B	256^2	64	1,024[b]
DepthVLM	`JonnyYu828/DepthVLM-4B`	Qwen3-VL	4B	256^2	64	1,024[b]
Qwen3-VL	`Qwen/Qwen3-VL-4B-Instruct`	Qwen3-VL	4B	256^2	64	1,024[b]
SigLIP2	`google/siglip2-base-patch16-224`	ViT-B/16	86M	224^2	196	768
InternViT	`OpenGVLab/InternViT-300M-448px-V2_5`	ViT, patch 14	300M	224^2	256	1,024
SigLIP	`google/siglip-base-patch16-224`	ViT-B/16	86M	224^2	196	768
CLIP	`openai/clip-vit-base-patch16`	ViT-B/16	86M	224^2	196	768
VC-1	`facebook/vc1-base`	ViT-B/16	86M	224^2	196	768
Voltron	`v-cond-base` (`voltron-robotics`)	ViT-B/16	86M	224^2	196	768
Theia	`theaiinstitute/theia-base-patch16-224-cdiv`	DeiT-B/16	86M	224^2	196	768
C-RADIOv4	`c-radio_v4-h` (`NVlabs/RADIO` hub)	ViT-H/16	631M	224^2	196	1,280
RADIOv2.5	`radio_v2.5-b` (`NVlabs/RADIO` hub)	ViT-B/16	98M	224^2	196	768
ViT	`google/vit-base-patch16-224`	ViT-B/16	86M	224^2	196	768

[a] Concatenation of the frame-attention and global-attention outputs of the last aggregator block. [b] Projected to 1,024 channels by the VLM-style interface described below; VLM parameters include the language model and the vision tower.

to 384×384 with ImageNet statistics and runs in image mode with one temporal slot. VGGT-Ω, SPA, RADIOv2.5, and C-RADIOv4 receive $[0, 1]$ images and apply their own internal normalization, Theia receives 8-bit images and normalizes them internally, and Voltron uses its released transform. LingBot-Vision is resized bilinearly to 224×224 with ImageNet statistics. The three VLMs use the Qwen3-VL image processor, which enlarges a 224×224 view to 256×256 to meet its minimum pixel count; 16-pixel patches and 2×2 merging then yield 64 image tokens per view.

Token extraction. Class and register tokens are removed and only patch tokens are kept. VGGT-Ω contributes the patch tokens of its last aggregator block and drops the camera and register tokens. On RoboTwin it receives the head view and the two wrist views together as one three-frame sequence, so its global-attention layers exchange information across the views before the policy reads the 3×196 tokens; every other encoder processes each view on its own. Its diagnostics, like all others, read only the head camera. MAE runs with a masking ratio of 0; the released implementation still permutes the order of its output tokens, which leaves the policy unaffected because its cross-attention does not depend on token order, but makes MAE's position-dependent statistics meaningless (Section H.1). LingBot-Vision contributes its final normalized patch tokens. For the VLMs, the images of all views are placed in the chat template before the instruction, the model runs once, and the hidden states at the image-token positions of the chosen layer are kept, the final layer unless stated otherwise. The language model attends causally and the image tokens precede the instruction, so these hidden states do not depend on the instruction; image tokens of a later view can attend to those of earlier views. The instruction reaches the policy only through the CLIP text tokens described in Appendix C, which enter the context after the image tokens.

C Training and Evaluation Protocol

Two policy interfaces. Vision encoders use the interface of Eq. (2). Their patch tokens enter the action head at their native width, so $D_{\text{ctx}} = D_{\text{enc}}$ is the input width of the cross-attention key and value projections and the hidden width of the action decoder. Single-task policies read vision only.

Table 9: Checkpoints of the model-scale study (Section G.5). Every row is read through the same interface as its default checkpoint in Table 8; the default rows are repeated for reference.

Family	Size	Checkpoint	Backbone	Params	N	D
DINOv2	S	`facebook/dinov2-small`	ViT-S/14	21M	256	384
	B (default)	`facebook/dinov2-base`	ViT-B/14	86M	256	768
	L	`facebook/dinov2-large`	ViT-L/14	300M	256	1,024
	g	`facebook/dinov2-giant`	ViT-g/14	1.1B	256	1,536
DINOv3	S	`facebook/dinov3-vits16-pretrain-lvd1689m`	ViT-S/16	21M	196	384
	B (default)	`facebook/dinov3-vitb16-pretrain-lvd1689m`	ViT-B/16	86M	196	768
	L	`facebook/dinov3-vitl16-pretrain-lvd1689m`	ViT-L/16	300M	196	1,024
	H+	`facebook/dinov3-vith16plus-pretrain-lvd1689m`	ViT-H+/16	840M	196	1,280
V-JEPA 2.1	B (default)	`vjepa2_1_vit_base_384`	ViT-B/16	80M	576	768
	L	`vjepa2_1_vit_large_384`	ViT-L/16	300M	576	1,024
	g	`vjepa2_1_vit_giant_384`	ViT-g/16	1B	576	1,408
VC-1	B (default)	`facebook/vc1-base`	ViT-B/16	86M	196	768
	L	`facebook/vc1-large`	ViT-L/16	307M	196	1,024
SPA	B (default)	`spa_vit_base_patch16`	ViT-B/16	86M	196	768
	L	`spa_vit_large_patch16`	ViT-L/16	300M	196	1,024
C-RADIOv4	SO400M	`c-radio_v4-so400m`	ViT-SO400M/16	412M	196	1,152
	H (default)	`c-radio_v4-h`	ViT-H/16	631M	196	1,280

Multi-task policies append the token-level last hidden states of a frozen CLIP ViT-B/32 text encoder, padded to the longest instruction in the batch and linearly projected to D_{enc}. The three VLMs and LingBot-Vision use a second interface. Their image tokens are linearly projected to 1,024 channels, and the 77 token-level states of a frozen CLIP ViT-B/16 text encoder are projected to the same width and appended, so $D_{\text{ctx}} = 1{,}024$. This interface is used in every regime, so in single-task runs the text tokens are constant within a policy. In both interfaces the DiT itself keeps width 768, and the VLM, the text encoders, and frozen visual encoders receive no gradient.

Action head. The head follows the GR00T flow-matching design (Bjorck et al., 2025). It is a 16-block DiT with 12 heads of dimension 64 (width 768), alternating cross-attention blocks with self-attention blocks, adaptive layer normalization on the flow time, dropout 0.2 including the final layer, and a two-layer action decoder whose hidden width is D_{ctx}. The last block is projected to an output width before the decoder; the released RoboCasa configurations set it to 768, and the RoboTwin configurations keep the class default of 26. Its query sequence consists of 32 learned tokens followed by the 16 action tokens of the chunk, which are embedded together with the time discretized into 1,000 buckets and receive learned position embeddings. Flow time is drawn as $\tau = (s - \min(b, s))/s$ with $b \sim \text{Beta}(1.5, 1)$ and $s = 0.999$, every example is repeated with eight independent noise and time draws per step, and the loss is the mean squared error of the predicted velocity. Inference integrates the velocity with 15 Euler steps on RoboCasa and 10 on RoboTwin.

Data and action spaces. No policy receives proprioceptive input; everything a policy knows about the robot's configuration comes from its images. RoboCasa-GR1 policies use the egocentric head view of the GR1 simulation data and up to 1,000 demonstrations per task. Actions have 29 coordinates (left arm 7, right arm 7, left hand 6, right hand 6, waist 3) and are min–max normalized. RoboTwin 2.0 policies use the head camera and the left and right wrist cameras of the domain-randomized demonstrations, 500 per task. Actions are 14 absolute joint positions (two 6-DoF arms and two grippers) with min–max normalization. Frames are resized to 224×224 in the data loader. Data-scale runs select complete demonstrations with a fixed subset seed, so smaller budgets are nested in larger ones.

Optimization. All recipes train for 100k steps with a global batch of 128 (32 on each of four GPUs) and no gradient accumulation. We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, and weight decay 10^{-8}, gradient clipping at 1, bfloat16 mixed precision, gradient checkpointing, and DeepSpeed ZeRO stage 2. The learning rate of the action head and projections is 10^{-4}, with linear warmup and cosine decay to the floor in Table 6. RoboCasa multi-task runs lower two parameter groups to 3×10^{-5}: the text projector and the base group, which holds every trainable parameter outside the action head and the text projector (the vision projection, and the encoder when it is fine-tuned); the action head keeps 10^{-4}. Frozen runs keep the encoder fixed; fine-tuned runs update it with the same recipe.

Evaluation. The final checkpoint of each run is evaluated; no online validation selects checkpoints. Each configuration is trained with one seed and evaluated with three evaluation seeds of 50 rollouts per task. On RoboCasa the policy executes the first 12 of the 16 predicted actions before replanning, and an episode ends after at most 720 control steps. On RoboTwin it executes the full 16-step chunk without temporal ensembling, in the domain-randomized setting with unseen instructions. A task score is the mean over the evaluation seeds, and the task-level deviation σ_k is the population standard deviation of the three evaluation-seed scores. A suite score is the unweighted mean of the twelve task scores. Per-seed suite scores are not exported, so the reported suite deviation is the propagated value $\sqrt{\sum_k \sigma_k^2}/12$, which assumes independent task errors, rather than the deviation of the suite score across seeds. Figure 9 shows the encoders in each experiment.

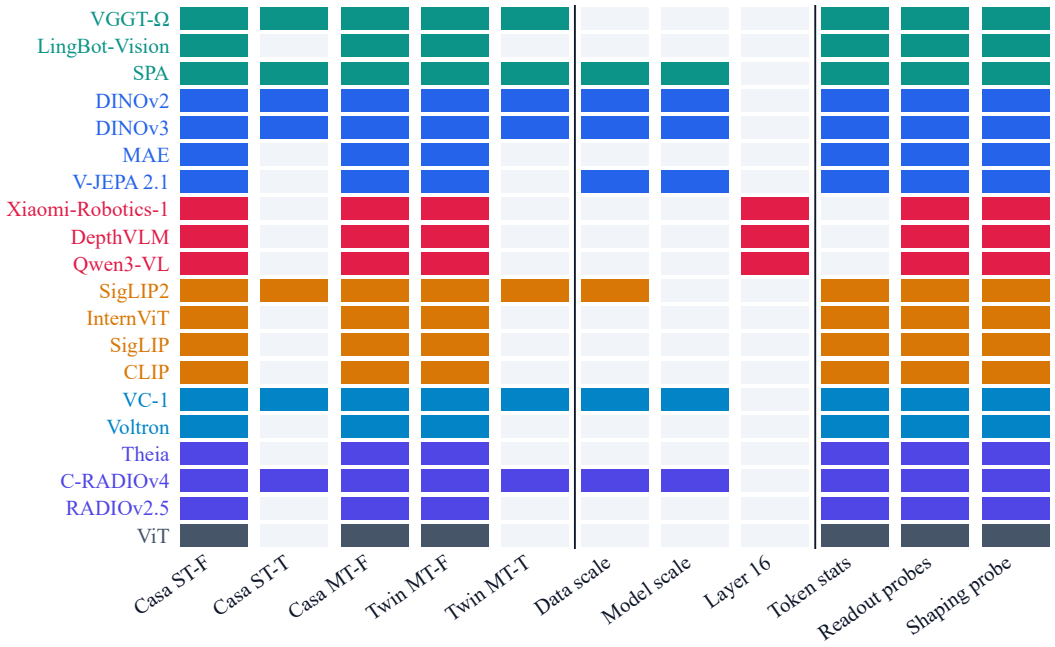


Figure 9: Experiment coverage. A colored cell marks a measured configuration, in the family color of the encoder; a grey cell is not measured. F denotes a frozen and T a fine-tuned encoder.

D Metric Definitions

Table 10 maps every diagnostic to its key in the result export and analysis code, the axis it measures, and whether it depends on the overall scale of the tokens. Scale-dependent statistics change when an encoder's features are multiplied by a constant, so their raw values mostly reflect normalization conventions and should not be compared across families without care. The subsections below give the formulas.

D.1 Notation and sampling

A frame t has spatial tokens $z_{t,1}, \ldots, z_{t,N} \in \mathbb{R}^D$ on an $H \times W$ grid. We write $r_{t,i} = \|z_{t,i}\|_2$ for token norms, $g_t = \frac{1}{N}\sum_i z_{t,i}$ for the mean-pooled frame feature, and $\cos(a, b) = a^\top b/(\|a\|\|b\| + \epsilon)$. Token statistics use every fifth demonstration video of each task, frames 0 to 100 at stride 5 (so $T \leq 21$ frames per video), from the RoboCasa head view or the RoboTwin head camera. Single-frame statistics are averaged over the frames of a video; every statistic is then averaged over the videos of a task and over the twelve tasks. The frames therefore cover a prefix of each demonstration, the first 100 control steps, rather than the whole episode. Statistics that compare patch positions (neighbor cosine, the frequency statistics, patch temporal drift, and temporal norm-histogram entropy) are not reported for MAE, whose output tokens are permuted.

Table 10: Metric dictionary. Axis: S spatial, T temporal, F spatial frequency, P readout probe, Pol trained policy, Sh distribution-shaping probe; *marks statistics that compare patch positions and are therefore not reported for MAE. Scale: "free" means invariant to multiplying all tokens by a constant.

Name in this paper	Key	Axis	Scale	Note
Anisotropy	`avg_token_cos`	S	free	Mean pairwise patch cosine
Mean token norm	`mean_token_norm`	S	dependent	
Neighbor cosine*	`neighbor_sim`	S	free	Adjacent patches only
Participation ratio	`token_cov_rank`	S	free	Exported as effective rank
Norm-histogram entropy	`token_norm_entropy`	S	free	Adaptive bins; shape of the norm histogram, not spread over patches
Token-norm variance	`token_norm_var`	S	dependent	
Token-to-global cosine	`token_to_global`	S	free	
Within/between-image var.	`within_between_var`	S, T	free	Patch spread relative to frame-to-frame spread
Temporal drift	`temporal_smoothness`	T	dependent	Near-duplicate of trajectory length
Temporal cosine shift	`temporal_cosine_shift`	T	free	
Temporal variance	`temporal_variance`	T	dependent	
Temporal participation ratio	`temporal_effective_rank`	T	free	
Temporal eigenvalue entropy	`temporal_spectral_entropy`	T	free	Covariance eigenvalues, not a Fourier spectrum
Lag-1 autocorrelation	`autocorrelation`	T	free	
Trajectory length	`total_trajectory_variation`	T	dependent	Temporal drift times $(T-1)/T$
Patch temporal drift*	`patch_temporal_smoothness`	T	dependent	
Temporal norm-histogram entropy*	`temporal_ token_norm_entropy`	T	free	Per patch position over time, normalized to $[0, 1]$
Within/between-trajectory var.	`trajectory_var_ratio`	T	free	
Low-, mid-, high-frequency share*	`low/mid/high_ratio`	F	free	Shares sum to one
Spectral entropy, centroid, bandwidth*	`spectral_*`	F	free	
IDM error	`IDM`	P	free	Standardized MSE of the terminal command displacement
State error	`State`	P	free	Standardized MSE; probe width follows the tokens
Object-position error	`Object`	P	–	MAE in meters; RoboTwin only; no chance baseline
Texture sensitivity	`action_robustness`	Pol	–	Variance of a trained policy's chunk; RoboTwin only
AUC, t_{50}, t_{25}, t_{10}, residual	`g_*`, `s_*`	Sh	free	Gaussian (g) and hard-sphere (s) targets have different floors
Feature-distance preservation	`g_dist`, `s_dist`	Sh	free	Exported as kept geometry
Sphere/Gauss ratios	`ball_gap`, `ball_auc_ratio`	Sh	free	Mix fit quality with the sphere floor

D.2 SPATIAL STATISTICS

Anisotropy (mean patch cosine) is $\frac{1}{N(N-1)} \sum_{i \neq j} \cos(z_i, z_j)$.

Mean token norm is $\bar{r} = \frac{1}{N} \sum_i r_i$.

Neighbor cosine is $\frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} \cos(z_i, z_j)$ over the set $\mathcal{E}$ of horizontally and vertically adjacent patch pairs, each counted once.

Participation ratio (reported in the export as effective rank) is computed from the patch covariance. With $C = \frac{1}{N-1} \sum_i (z_i - \bar{z})(z_i - \bar{z})^\top$ and eigenvalues $\lambda_k \geq 0$,

$$\mathrm{PR}(C) = \frac{\left(\sum_k \lambda_k\right)^2}{\sum_k \lambda_k^2}. \tag{6}$$

It equals the number of directions when variance is spread equally over them and 1 when all of it lies in a single direction of the token space.

Norm-histogram entropy. The norms $r_1, \ldots, r_N$ are binned into 30 equal-width bins that span the range of the frame's norms, from $\min_i r_i$ to $\max_i r_i$. With bin frequencies q_b,

$$H_{\text{norm}} = -\sum_{b=1}^{30} q_b \ln q_b, \qquad 0 \le H_{\text{norm}} \le \ln 30 \approx 3.40. \tag{7}$$

Because the bins adapt to each frame's range, a few patches with norms far above the rest compress all other patches into the lowest bins and lower the entropy. The statistic describes the shape of the norm histogram; it does not measure how evenly activation is spread over patches, since identical norms also give low entropy.

Token-norm variance is $\frac{1}{N}\sum_i (r_i - \bar{r})^2$.

Token-to-global cosine is $\frac{1}{N}\sum_i \cos(z_i, g)$.

Within/between-image variance. For the T frames of one video,

$$R_{\text{img}} = \frac{\frac{1}{T}\sum_t \frac{1}{N}\sum_i \|z_{t,i} - g_t\|_2^2}{\frac{1}{T}\sum_t \|g_t - \bar{g}\|_2^2 + \epsilon}, \qquad \bar{g} = \frac{1}{T}\sum_t g_t. \tag{8}$$

D.3 Temporal statistics

Let $S = [g_1, \ldots, g_T]^\top \in \mathbb{R}^{T \times D}$ be the pooled trajectory of one video and S_c the same trajectory centered in time.

Temporal drift is $\frac{1}{T-1}\sum_{t=1}^{T-1} \|g_{t+1} - g_t\|_2$.

Temporal cosine shift is $\frac{1}{T-1}\sum_t [1 - \cos(g_t, g_{t+1})]$.

Temporal variance is $\frac{1}{D}\sum_d \mathrm{Var}_t(g_{t,d})$.

Temporal participation ratio is the participation ratio of the eigenvalues of $G = S_c S_c^\top/(T-1)$.

Temporal eigenvalue entropy (exported as temporal spectral entropy) is $-\sum_j p_j \ln p_j$ with $p_j = \lambda_j / \sum_k \lambda_k$ from the same eigenvalues; it involves no Fourier transform.

Lag-1 autocorrelation. With centered channels $x_{t,d} = g_{t,d} - \bar{g}_d$ and $\mathcal{D}_+$ the channels of nonzero variance,

$$A(1) = \frac{1}{|\mathcal{D}_+|}\sum_{d \in \mathcal{D}_+} \frac{\sum_{t=1}^{T-1} x_{t,d}\, x_{t+1,d}}{\sum_{t=1}^{T} x_{t,d}^2}. \tag{9}$$

Trajectory length is $\frac{1}{T}\sum_{t=1}^{T-1} \|g_{t+1} - g_t\|_2$, which equals temporal drift up to the factor $(T-1)/T$.

Patch temporal drift is $\frac{1}{N(T-1)}\sum_{t,i} \|z_{t+1,i} - z_{t,i}\|_2$, comparing patches at the same grid position.

Temporal norm-histogram entropy. For each patch position i, the norms $r_{1,i}, \ldots, r_{T,i}$ are binned into 16 equal-width bins over their range, giving entropy H_i; the statistic is $\frac{1}{N}\sum_i H_i / \ln 16 \in [0, 1]$.

Within/between-trajectory variance. For the V videos of a task with trajectories $S^{(v)}$ and means $\bar{g}^{(v)}$,

$$R_{\text{traj}} = \frac{\frac{1}{V}\sum_v \frac{1}{D}\sum_d \mathrm{Var}_t\big(g_{t,d}^{(v)}\big)}{\frac{1}{D}\sum_d \mathrm{Var}_v\big(\bar{g}_d^{(v)}\big) + \epsilon}. \tag{10}$$

D.4 Frequency statistics

Each channel d is reshaped to an $H \times W$ map Z_d, its spatial mean is subtracted, and an orthonormal 2D discrete Fourier transform gives the power $P_d(u,v) = |\mathcal{F}(Z_d)(u,v)|^2$. The frequency radius $\varrho(u,v) = \sqrt{f_u^2 + f_v^2} / \max \sqrt{f_u^2 + f_v^2}$ uses the discrete frequencies f_u, f_v of the grid, and $p_d = P_d / \sum_{u,v} P_d$. Every statistic is computed per channel and then averaged over channels.

Low-, mid-, and high-frequency shares are $\sum p_d$ over $\varrho \le 0.15$, $0.15 < \varrho \le 0.35$, and $\varrho > 0.35$.

Spectral entropy is $-\sum_{u,v} p_d \ln p_d / \ln(HW)$.

Spectral centroid is $\mu_d = \sum_{u,v} p_d \, \varrho$.

Spectral bandwidth is $\sqrt{\sum_{u,v} p_d (\varrho - \mu_d)^2}$.

D.5 Readout probes and the texture diagnostic

All probes keep the encoder frozen, read it through the same preprocessing as the policy (Appendix B), and train only a small head. They use 1,000 training and 500 held-out RoboCasa demonstrations or 400 and 100 RoboTwin demonstrations, stratified over the twelve tasks; training runs for four epochs with AdamW (learning rate 5×10^{-4}, weight decay 10^{-4}, cosine decay to 1%), batch 256, gradient clipping at 1, and seed 42. The epoch with the lowest held-out error is kept, and the error on the same held-out split is reported, so every probe error is a validation error after selecting among four epochs. Targets are standardized with training-set statistics over the control coordinates (29 on RoboCasa, 14 on RoboTwin). Every transformer below is a pre-norm encoder with four heads, a feed-forward width of twice its model width, and dropout 0.1.

Inverse dynamics (IDM). The input is the token pair (Z_t, Z_{t+16}). Both token sets are projected to width 256 and receive sinusoidal position encodings within the frame and a segment embedding that marks the frame. A learnable query token with its own segment embedding is prepended to the $2N$ tokens, a four-layer transformer of width 256 mixes them, and a two-layer head (LayerNorm, 256 hidden units, GELU) maps the query's output to $\hat{y}$. The target is the standardized displacement $y = a_{t+15} - s_t$ between the commanded action at the end of the 16-step window and the current state. Because the actions are absolute position targets, y is a terminal command displacement, not the executed motion or the full action sequence. The reported value is the held-out mean squared error $\frac{1}{|\mathcal{V}|} \sum \|\hat{y} - y\|^2 / \dim y$.

State. After a LayerNorm on the tokens of a single frame, a learnable query token is appended, a four-layer transformer at the encoder width mixes them, and a two-layer head predicts the standardized state s_t from the query's output. The reported value is the held-out mean squared error.

Object position. On four RoboTwin tasks, the same architecture reads the first head-camera frame of an episode and predicts the camera-frame 3D position of the task object (the bottle, the block under the hammer, the alarm clock, and the laptop), with 400 training and 100 held-out episodes per task. The reported value is the mean absolute error in meters, averaged over the three coordinates. No mean-predictor baseline was exported, so the errors rank encoders but do not show how far each is above chance.

Table 11: Readout-probe capacity. Trainable parameters, excluding the frozen encoder, as a function of the width of the tokens a probe reads. The IDM probe projects to width 256; the state and object probes run a four-layer transformer at the token width, so their capacity grows roughly with its square.

Token width	768	1,024	1,280	2,048	2,560
IDM probe	2.38M	2.45M	2.51M	2.71M	2.84M
State and object probes	19.5M	34.7M	54.2M	138.6M	216.5M

Action sensitivity to texture. For each RoboTwin task, 100 initial observations share object pose, robot state, and camera and differ only in table and wall textures. The trained policy predicts a normalized action chunk $a^{(j)} \in \mathbb{R}^{H_a \times D_a}$ from each texture j with the same initial diffusion noise. The statistic is

$$V_{\text{tex}} = \frac{1}{12} \sum_{k=1}^{12} \frac{1}{H_a D_a} \sum_{h,d} \operatorname{Var}_j \big(a^{(j)}_{k,h,d}\big), \tag{11}$$

the across-texture variance averaged over chunk entries and tasks.

D.6 Distribution-shaping probe

Inputs. Frames are taken every tenth step from up to 1,000 RoboCasa or 500 RoboTwin demonstrations per task and thinned uniformly to 500 or 2,000 frames per task (head view only). Each frame's tokens are average-pooled to a common 14×14 grid (196 tokens) and projected to 256 dimensions with a fixed Gaussian matrix $R \in \mathbb{R}^{D \times 256}$, $R_{ij} \sim \mathcal{N}(0, 1/256)$. The mean-pooled projected tokens $\bar{x}_i$ serve as the source geometry.

Adapter and targets. A weak adapter—parameter-free layer normalization, one learned query attending over the tokens (width 128, four heads), and a linear map—sends frame i to $z_i \in \mathbb{R}^3$. For a batch of B outputs, the Gaussian target uses the closed-form kernel maximum mean discrepancy to $\mathcal{N}(0, I_3)$,

$$D_{\mathcal{G}} = \frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} \Big[\frac{1}{B^2} \sum_{i,j} e^{-\gamma \|z_i - z_j\|^2} - \frac{2}{(1+2\gamma)^{3/2}} \frac{1}{B} \sum_{i} e^{-\gamma \|z_i\|^2/(1+2\gamma)} + \frac{1}{(1+4\gamma)^{3/2}} \Big], \tag{12}$$

with $\Gamma = \{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}\}$. The hard-sphere target combines uniformity on the unit sphere with a radius penalty,

$$D_{\mathcal{S}} = \frac{1}{|K|} \sum_{\kappa \in K} \frac{1}{B(B-1)} \sum_{i \neq j} e^{\kappa(\hat{z}_i^\top \hat{z}_j - 1)} + \frac{1}{B} \sum_{i} \big(\|z_i\| - \sqrt{3}\big)^2, \qquad \hat{z} = \frac{z}{\|z\|}, \tag{13}$$

with $K = \{1, 2, 4\}$. A distance-preservation term compares normalized pairwise distances,

$$P = \operatorname*{mean}_{i<j} \Big(\frac{\|z_i - z_j\|}{\|z - z'\|} - \frac{\|\bar{x}_i - \bar{x}_j\|}{\|\bar{x} - \bar{x}'\|} \Big)^2, \tag{14}$$

and the training loss is $L = D / \max(|D_0|, 10^{-4}) + 0.05\, P$, where D_0 is the discrepancy on the first batch. Training uses AdamW (learning rate 3×10^{-4}), batch 256, ten epochs (240 steps on RoboCasa, 940 on RoboTwin), and seed 2026.

Summaries. With the loss sequence $L_1, \ldots, L_K$ and $\ell_s = L_s/|L_1|$: the AUC is the trapezoidal area under ℓ divided by K; t_x is the first step with $\ell_s \leq x$, divided by K (t_{50} is the half-time); the residual is D at the last step; feature-distance preservation is the Spearman correlation between $\{\|z_i - z_j\|\}$ and $\{\|\bar{x}_i - \bar{x}_j\|\}$ over the first 2,048 frames after training; tail oscillation is the standard deviation divided by the mean of L over the last 10% of steps; early loss is the mean of ℓ over the first 20%; rebound is $L_K / \min_s L_s$; relative drop is $1 - \min_s L_s / L_1$; adapter norm is the L2 norm of all adapter parameters. The residual ratio and AUC ratio divide the sphere value by the Gaussian value.

Baselines of the two targets. The two discrepancies have different floors. $D_{\mathcal{G}}$ is zero when the outputs follow $\mathcal{N}(0, I_3)$. The uniformity term of $D_{\mathcal{S}}$ is positive even for outputs spread exactly uniformly on the sphere: for uniform $\hat{z}$, $\mathbb{E}\, e^{\kappa(\hat{z}_i^\top \hat{z}_j - 1)} = (1 - e^{-2\kappa})/(2\kappa)$, which averages to 0.2676 over K. Sphere residuals, and ratios of sphere to Gaussian values, therefore mix fit quality with this offset. We use them only to rank encoders within a target and do not read a residual as a distance from the target distribution.

E Encoders per Analysis

Table 12 lists, for every analysis, how many encoders enter each cohort. Closed-loop results cover all 20 encoders. Every diagnostic–success analysis uses the 17 vision encoders, and the three VLMs are

compared separately in Section H.2. The vision-benchmark comparison uses the checkpoints with published scores, including sizes from the model-scale study. Token statistics cover all seven fine-tuned checkpoints, except the frequency statistics of fine-tuned VGGT-Ω, which the export copies from the frozen checkpoint. Statistics that compare patch positions are not defined for MAE, whose released implementation permutes its output tokens. Missing values are never filled in from other cells or cohorts.

Table 12: Encoders per analysis. Number of encoders per cohort for every analysis. Twin+FT adds the fine-tuned RoboTwin checkpoints to the frozen ones. A dash marks a cohort in which the analysis is not run.

Analysis	Casa ST	Casa MT	Twin MT	Twin+FT	Encoders
Closed-loop results (Section G)	20	20	20	27	All encoders
Token statistics	17	17	17	24	Vision encoders and all fine-tuned checkpoints
Position-dependent token statistics	16	16	16	22–23	As above, without MAE; frequency statistics also without fine-tuned VGGT-Ω
IDM and state errors	17	17	17	24	Vision encoders; VLMs compared separately
Object-position error	–	–	17	24	Vision encoders; measured on RoboTwin
Texture sensitivity	–	–	17	24	Vision encoders; measured on RoboTwin
Shaping-probe summaries	17	17	17	24	Vision encoders
Prediction, ranking, and selection	16	16	16	–	Encoders with every diagnostic used by the analysis
Published vision benchmarks	–	9–11	9–11	–	Sizes of DINOv2, DINOv3, and C-RADIOv4; SigLIP2; LingBot-Vision
Paired fine-tuning changes		7 pairs		–	Success in Casa ST and Twin MT; diagnostics after fine-tuning on RoboTwin

F Statistical Methods

Correlation. Ranks use mid-ranks for ties. For diagnostic values x and success values y over n encoders, Pearson's correlation is

$$r(x, y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}, \tag{15}$$

and Spearman's correlation is $\rho(x, y) = r(\operatorname{rank} x, \operatorname{rank} y)$.

Permutation p-value. With ρ_{obs} the observed value and ρ_b the value after the b-th random permutation of y, $p = \big(1 + \#\{b : |\rho_b| \geq |\rho_{\text{obs}}|\}\big)/(1 + B)$ with $B = 100{,}000$. Each test draws its permutations from its own generator, seeded by a CRC-32 hash of the cohort and diagnostic names, so a p-value does not change when other diagnostics are added or removed.

Bootstrap interval. The n encoders are resampled with replacement 4,000 times; the interval is the 2.5th to 97.5th percentile of the resampled ρ.

False-discovery control. Within a cohort, the m permutation p-values are sorted, $p_{(1)} \leq \cdots \leq p_{(m)}$, and the Benjamini–Hochberg value is $q_{(i)} = \min_{j \geq i} \min\big(1, m\, p_{(j)}/j\big)$.

Evidence classes. A *robust* association has $p < 0.05$ and $q < 0.10$ in its cohort, a *nominal* one $p < 0.05$ only, and a *trend* the same sign in all frozen cohorts without $p < 0.05$. Thresholds, windows, and median splits chosen after inspecting the data are *exploratory*; the group sizes behind every such split are given where it is used. Mechanisms that were not tested by an intervention are *hypotheses*.

Dependence between units. The unit of every correlation is an encoder. Encoders of one family share objectives, teachers, or training data, several distilled encoders share teachers with the encoders they are compared with, and in Twin+FT each fine-tuned checkpoint shares its initialization with its frozen counterpart. The permutation test treats the units as exchangeable, so its p-values are optimistic when such dependence is strong. We therefore rely on agreement across cohorts and tasks rather than on any single p-value, and we report the fine-tuned checkpoints both pooled and on their own.

Partial Spearman correlation given IDM. The ranks of the diagnostic and of success are each regressed by least squares on the ranks of IDM error, and r is computed between the two residual vectors.

Shape of a relation. Terciles sort the encoders by the diagnostic and split them into three groups of nearly equal size; the shape column of Table 13 reports which third has the highest or lowest mean success on every frozen cohort. Quartile separation takes the top and bottom success quartiles $\mathcal{T}$ and $\mathcal{B}$, computes $A = \frac{1}{|\mathcal{T}||\mathcal{B}|} \sum_{i \in \mathcal{T}, j \in \mathcal{B}} \left[\mathbb{1}(x_i > x_j) + \frac{1}{2} \mathbb{1}(x_i = x_j) \right]$, and reports $\max(A, 1 - A)$. The inverted-U test correlates success with $|\operatorname{rank} x_i - \frac{n+1}{2}|$; a negative value with a small permutation p means that the middle ranks succeed most. The jackknife range is the minimum and maximum of ρ with each encoder left out in turn.

Prediction. The leave-one-encoder-out regression converts each diagnostic to ranks within the cohort, fits $y = \beta_0 + \sum_m \beta_m \operatorname{rank} x_m$ by least squares on $n - 1$ encoders, predicts the remaining one, and reports the mean absolute error over all n held-out predictions; the quadratic version adds $(\operatorname{rank} x)^2$. The rank average orients each diagnostic by a sign $\sigma_m \in \{\pm 1\}$ so that higher is better, and scores encoder i by $\frac{1}{|\mathcal{S}|} \sum_{m \in \mathcal{S}} \operatorname{rank}(\sigma_m x_{i,m})$. The 2×2 tables mark an encoder favorable on a diagnostic when $\sigma_m x_m$ is at least its cohort median and report mean success in each of the four cells.

Selection across protocols. The exhaustive search scores every diagnostic, every pair, and every triple drawn from the fourteen best single diagnostics by the minimum, over the three frozen cohorts, of the oriented rank-average ρ. The leave-one-protocol-out procedure sets each orientation from the mean ρ on two cohorts, selects the single diagnostic or pair with the best minimum on those two, and reports its oriented ρ on the third cohort, which plays no part in the selection.

Agreement and transfer. Cross-benchmark stability is ρ between a diagnostic's values on RoboCasa and on RoboTwin over the shared frozen encoders. Sign-consistent predictive strength is the mean $|\rho|$ over the three frozen cohorts when all three signs agree and zero otherwise. Profile agreement is Pearson's r between the vectors of signed ρ for two settings over the diagnostics measured on all three frozen cohorts, with the RoboCasa profile taken as the mean of its single-task and multi-task values. Transfer correlates a diagnostic measured on one benchmark with success on the other, and the mean-rank association correlates it with each encoder's mean rank over the three frozen cohorts.

G Extended Closed-Loop Results

This appendix expands Section 4; Table 2 there gives the frozen suite scores. Every policy is trained with one seed and evaluated with three evaluation seeds of 50 rollouts per task; suite scores and their propagated uncertainty are defined in Section A.4. Unless stated otherwise, DINOv2, DINOv3, and V-JEPA 2.1 are Base checkpoints, C-RADIOv4 is the H checkpoint, and VLMs contribute the image tokens of their final layer. Paragraphs that expand a numbered finding of the main text refer to it by number; the others are supporting observations.

G.1 Frozen encoders across suites

We first ask whether an encoder ranking survives a change of training regime or benchmark. We freeze the visual backbone, train only the action head, and repeat the comparison on RoboCasa single-task, RoboCasa multi-task, and RoboTwin multi-task (Table 2).

Finding 1 in detail: The top and bottom groups are stable across the three frozen cohorts; the middle is not. Read the rank columns of Table 2 from left to right. VGGT-Ω, DINOv2, and LingBot-Vision stay in the top six on all three boards, and CLIP, ViT, C-RADIOv4, and RADIOv2.5 stay in the bottom five. The two groups are separated by tens of points, while the propagated uncertainty of a suite score is about one point. The middle of the list behaves differently: DINOv3, VC-1, Voltron, and the three VLMs each move by eight places or more between boards. Success rankings agree at $\rho = 0.89$ between the two RoboCasa regimes but only at about 0.60 between RoboCasa and RoboTwin. Small differences among mid-ranked encoders on one benchmark should therefore not be read as a general order.

Family labels do not order the encoders. The families in Table 2 do not stack. Geometry contains the best RoboCasa encoder (VGGT-Ω) and one of the weakest (SPA); self-supervision places DINOv2 near the top and MAE in the middle; SigLIP2 stays well ahead of CLIP and SigLIP on

every board. The three Qwen3-VL-derived VLMs move most between benchmarks: DepthVLM, Xiaomi-Robotics-1, and Qwen3-VL take the first three places on RoboTwin and sit in the middle of both RoboCasa boards. Cameras, embodiment, action space, tasks, randomization, and the cross-view attention of the VLM interface all change with the benchmark, so the shift cannot be attributed to a single factor. Section H.2 examines one correlate, the poor head-view state decoding of the VLMs.

Under a fixed per-policy step budget, one shared policy scores lower than separate policies for 17 of 20 encoders. On RoboCasa, replacing a separate policy per task by one shared policy lowers 17 of the 20 scores, while the top and bottom groups stay the same. The losses are uneven: SPA loses half of its success, SigLIP2 keeps almost all of it, and MAE gains. Each task receives fewer updates under sharing, so these changes combine the effect of sharing a policy with that of a smaller per-task budget. The loss is unrelated to single-task strength ($\rho = 0.01$ between single-task success and the fraction kept); Appendix I.1 relates it to a summary of the shaping probe.

G.2 Patch tokens versus one pooled token

The policy reads every patch token of every view. To test whether separate tokens matter, we keep DINOv2 and DINOv3 frozen, keep the head and the RoboCasa multi-task recipe unchanged, and replace each view's patch tokens by their mean. The head then cross-attends to one token per view instead of 256 (DINOv2) or 196 (DINOv3), so its context shrinks by more than two orders of magnitude.

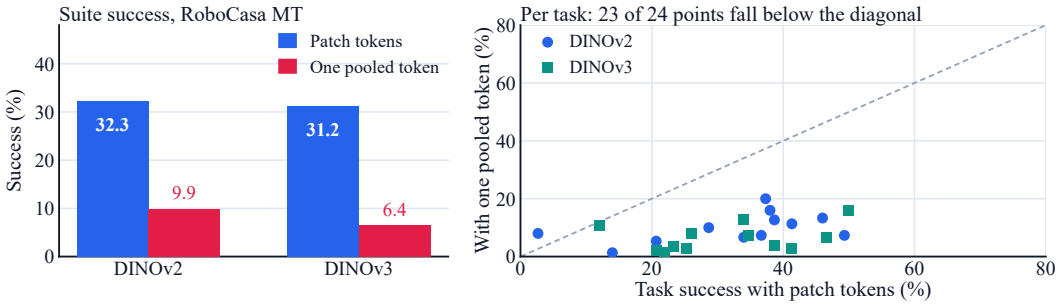


Figure 10: Patch tokens versus one mean-pooled token. Left: RoboCasa multi-task suite success of the same frozen encoders read through their patch tokens (blue) or through one mean token per view (rose). Right: success on each of the twelve tasks; points below the dashed diagonal are tasks on which pooling lowers success.

Finding 2 in detail: For frozen DINOv2 and DINOv3, one mean token per view removes most of the success of the patch-token policy. Mean pooling lowers DINOv2 from 32.3% to 9.9% and DINOv3 from 31.2% to 6.4% (Figure 10, left), between two thirds and four fifths of their success. The drop is not driven by a few tasks: 23 of the 24 task points fall below the diagonal, and the one exception is a task on which both variants stay below 10% (right). This is a controlled comparison, but a narrow one. It covers mean pooling of two encoders in one regime and does not test learned pooling, a class token, or other readouts; the loss of spatial layout is a plausible explanation that we did not test directly. For this head, averaging the patch tokens discards most of what the policy uses, so evaluations that read encoders through one mean embedding probe a different interface from the one a patch-conditioned policy uses, and their rankings need not carry over.

G.3 Fine-tuning the encoder

We fine-tune seven encoders end to end with the same recipe and compare each with its frozen run. The RoboTwin pairs are multi-task; RoboCasa has no multi-task fine-tuning, so the RoboCasa panel of Figure 11 uses the single-task runs, with the frozen single-task policies as reference.

Finding 3 in detail: In RoboCasa single-task and RoboTwin multi-task fine-tuning, every encoder gains and the frozen order largely survives. All fourteen pairs gain. On RoboTwin the gains range from 2.2 to 26.1 points: DINOv2 and SigLIP2, both strong when frozen, gain 2.2 and 4.5 points; SPA, DINOv3, and VC-1, which are weaker when frozen, gain between 18 and 25 points, and SPA more than triples its success; VGGT-Ω leads when frozen and also gains most. On RoboCasa single-task the gains are smaller, from 1.1 to 9.4 points. The gains are too small to reorder

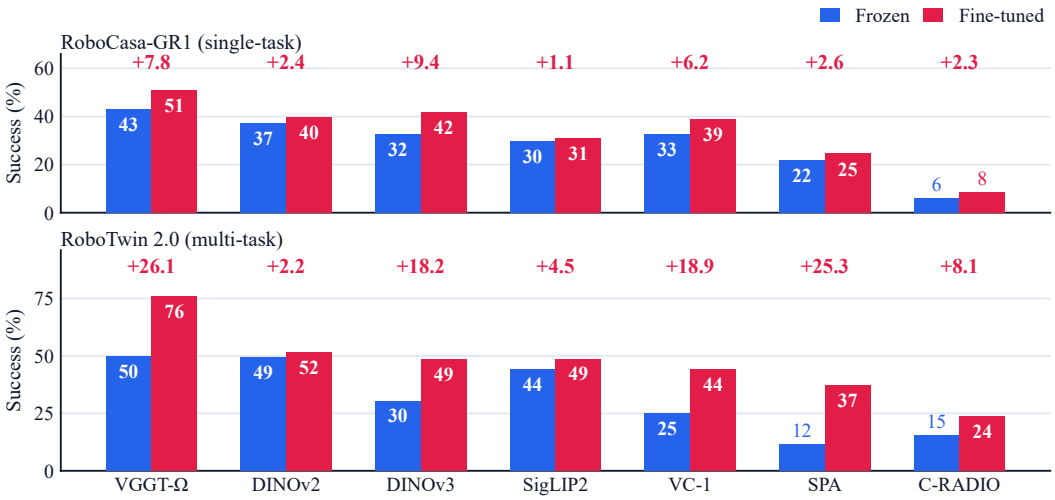


Figure 11: Frozen versus fine-tuned success. Blue is frozen and rose is fine-tuned. Bar labels are rounded to the nearest point; the top band gives the exact change in percentage points. RoboCasa: single-task; RoboTwin: multi-task.

the seven encoders much: fine-tuned success follows frozen success at $\rho = 0.96$ on RoboTwin, where only SPA and C-RADIOv4 swap places, and at $\rho = 0.89$ on RoboCasa. The size of the gain, in contrast, has no consistent relation with frozen success ($\rho = -0.14$ on RoboTwin, $+0.50$ on RoboCasa), so it needs its own measurement. With seven pairs per benchmark both statements are exploratory; Section H.7 relates the fine-tuned checkpoints to the diagnostics.

G.4 Demonstration scale

We retrain selected encoders with 50 to 1,000 demonstrations per task (Figure 12). The RoboCasa single-task sweep covers four tasks (C1–C4); the two multi-task sweeps cover all twelve.

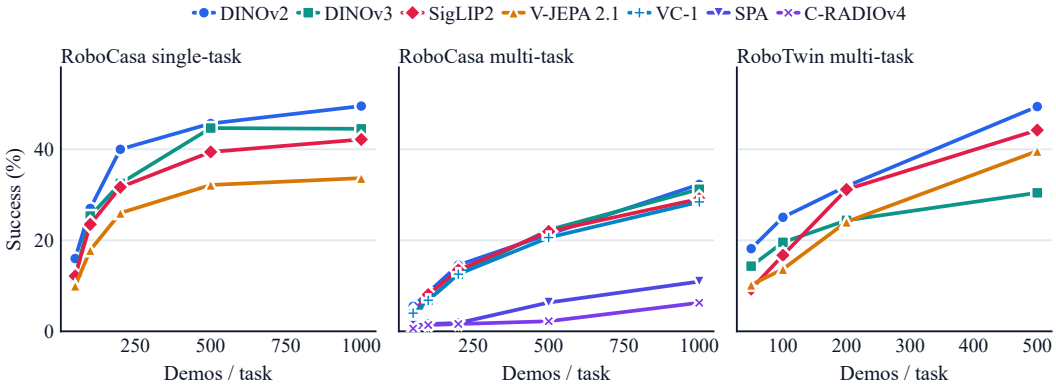


Figure 12: Success against demonstrations per task. The single-task panel averages tasks C1–C4; the multi-task panels average all twelve tasks. The comparison on matched tasks is given in the text.

More demonstrations raise success but do not close the gaps within the tested budgets. Curves that start above the floor keep rising and mostly keep their order. SPA and C-RADIOv4 stay below 11% at every RoboCasa multi-task budget, and the one clear change of order is SigLIP2 on RoboTwin, which starts behind DINOv3 and ends close to DINOv2. The single-task and multi-task panels average different task sets, so we compare their shapes on the same four tasks. On C1–C4, going from 500 to 1,000 demonstrations changes single-task success by -0.2 to $+3.8$ points for DINOv2, DINOv3, and SigLIP2, and multi-task success by $+10.3$ to $+13.0$ points. Within the tested range, single-task curves flatten on these tasks while multi-task curves are still rising, so comparisons made at small budgets are less settled for shared policies.

G.5 Model scale

We vary the checkpoint size within six families under multi-task training; Figure 13 shows the results, and Table 9 lists every checkpoint.

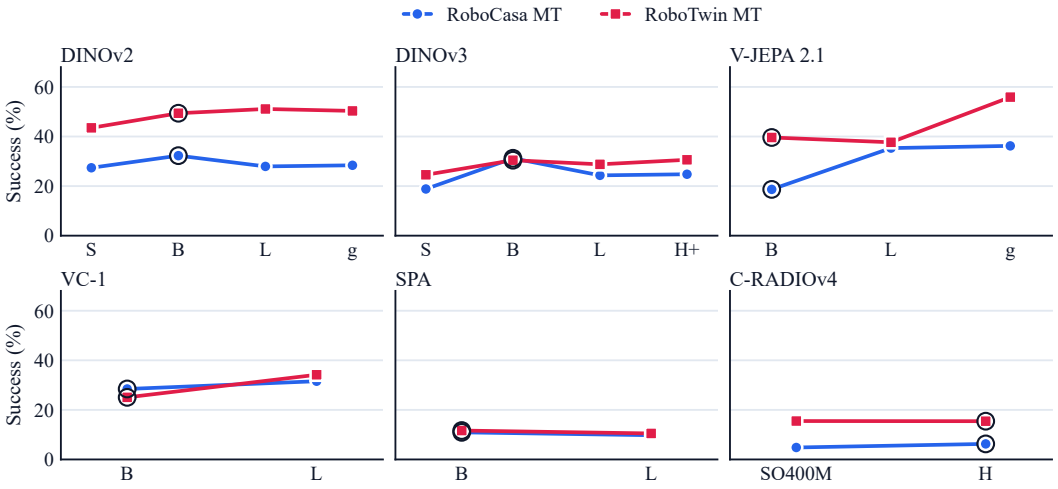


Figure 13: Multi-task success across checkpoint sizes. Blue: RoboCasa; rose: RoboTwin. Open rings mark the default checkpoints of Table 2. H+ is the largest distilled DINOv3 model; g is a ViT-g (1.1B for DINOv2, 1B for V-JEPA 2.1).

The effect of model size depends on the family and the benchmark. DINOv2 improves from Small to Base on both benchmarks, but Large and giant add nothing on RoboCasa and little on RoboTwin. DINOv3 Base is the best RoboCasa checkpoint of its family, and the RoboTwin curve stays between 24.6% and 30.6%. V-JEPA 2.1 rises steadily with size on RoboCasa, whereas on RoboTwin Large falls slightly below Base before the 1B model reaches the best point of the sweep (55.9%). VC-1 Large improves on both benchmarks, and SPA and C-RADIOv4 stay low at every size. These curves do not support a single scaling rule, and they do not identify why some families benefit.

G.6 Which VLM layer to read

VLM features are usually read at the final layer. We keep the image-token positions and read layer 16 instead in six configurations (Figure 14): all three VLMs on RoboCasa single-task, Qwen3-VL on RoboCasa multi-task, and Qwen3-VL and Xiaomi-Robotics-1 on RoboTwin.

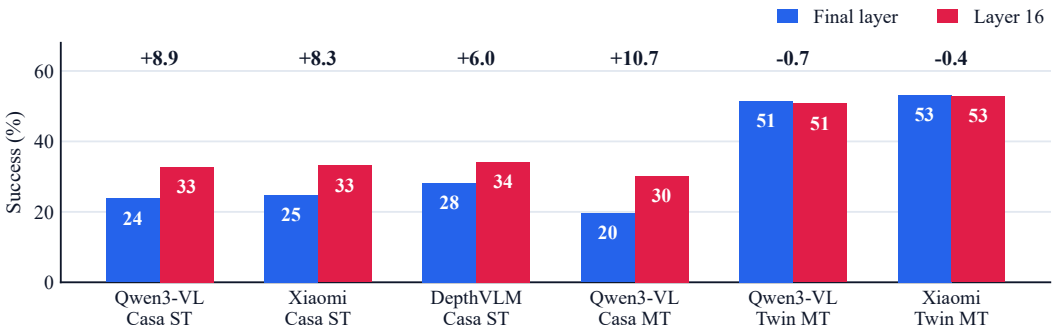


Figure 14: Final-layer image tokens versus layer 16. Blue is the final layer and rose is layer 16. Bar labels are rounded to the nearest point; the aligned numbers are the exact change.

Reading layer 16 changes VLM success in a setting-dependent way. On RoboCasa, layer 16 raises all four measured configurations: Qwen3-VL by 8.9 points single-task and 10.7 points multi-task, Xiaomi-Robotics-1 by 8.3 points and DepthVLM by 6.0 points single-task (Appendix K).

On RoboTwin, where the final layers already lead the board, the two layers differ by less than a point. These runs cover one intermediate layer and three VLMs, so they recommend the intermediate layer as an extraction option worth testing, not as a general improvement, and they do not show what the earlier layer retains.

H Extended Diagnostic Results

This appendix expands Section 5. It relates the diagnostics of Section A.5, and published vision-benchmark scores, to that success. Almost all evidence here is a correlation across 16 to 24 encoders, so every claim carries an evidence level. A *robust association* has a permutation $p < 0.05$ and a Benjamini–Hochberg $q < 0.10$ within its cohort; a *nominal association* has $p < 0.05$ only; a *trend* keeps its sign across cohorts without reaching $p < 0.05$. Groupings at thresholds chosen after seeing the data are *exploratory*, and explanations not tested by intervention are *hypotheses*. Numbered findings state regularities that hold on every cohort; patterns confined to one benchmark are observations.

H.1 Cohorts and statistics

We compare diagnostics with suite success in three frozen cohorts: RoboCasa single-task (Casa ST), RoboCasa multi-task (Casa MT), and RoboTwin multi-task (Twin MT). Each cohort contains the 17 vision encoders, for which every diagnostic family is available; the three VLMs are examined separately in Section H.2. A fourth cohort, Twin+FT, adds the seven fine-tuned RoboTwin checkpoints; the export repeats the frozen frequency statistics for fine-tuned VGGT-Ω, so those six values are left out. The released MAE implementation permutes its output tokens even without masking. The policy is unaffected, because its cross-attention ignores token order, but position-dependent statistics are undefined for MAE. Appendix E lists the encoders behind every analysis.

Our main statistic is Spearman's ρ between a diagnostic and suite success. Permutation p-values use 100,000 shuffles with a fixed seed per test, 95% intervals (in brackets) use 4,000 bootstrap resamples of encoders, and q-values control the false-discovery rate over the 49 to 51 diagnostics of a cohort. Errors and variances keep their raw direction, so a negative ρ means that a smaller value goes with higher success. Encoders share families, teachers, and training data, and a fine-tuned checkpoint shares its initialization with its frozen version, so the points of a correlation are not fully independent. Table 13 summarizes the associations, and Appendix F gives the formulas.

H.2 Readout probes

A frozen token set supports control only if what the policy needs can be read from it, and the readout probes test this directly. The IDM probe decodes the terminal command displacement from two head-camera frames 16 steps apart, the state probe decodes the robot's joint state from one frame, and the object probe decodes the 3D position of the task object from the first frame (Section A.5). None of our policies receives proprioception, so the robot's configuration has to come from the images. Figure 15 shows the probes and the policy diagnostic encoder by encoder.

Finding 4 in detail: Readout probes have the strongest association with success in every frozen cohort. IDM error goes with lower success on Casa ST ($\rho = -0.64$ $[-0.90, -0.14]$, $p = 0.007$), Casa MT (-0.64 $[-0.93, -0.13]$, $p = 0.007$), and Twin MT (-0.82 $[-0.94, -0.51]$, $p < 0.001$). State error behaves the same way (-0.68 $[-0.86, -0.28]$, -0.64 $[-0.87, -0.18]$, and -0.69 $[-0.90, -0.26]$). On RoboCasa the associations are nominal ($q = 0.16$ to 0.17), and on RoboTwin they are robust ($q \leq 0.042$), also after adding the fine-tuned checkpoints (IDM -0.76, state -0.71). They hold up when any single encoder is left out ($|\rho| \geq 0.57$) and when any one of the six families is left out ($|\rho| \geq 0.49$). The third of encoders with the highest IDM error averages 12.9%, 9.8%, and 16.8% success on the three cohorts, 16 to 26 points below the best third. Task by task, IDM error has the expected sign on all 36 task-level correlations (the 12 RoboCasa tasks in both regimes and the 12 RoboTwin tasks) and state error on 35, with median per-task correlations of 0.60 and 0.59. No token statistic reaches these values. The best of them, norm-histogram entropy, reaches $+0.56$, $+0.41$, and $+0.43$, with a per-task median of 0.42. A paired bootstrap puts the advantage of IDM over this entropy at $+0.08$, $+0.23$, and $+0.40$ in $|\rho|$; each interval includes zero, the RoboTwin one narrowly ($[-0.01, 0.80]$), so the ordering is consistent but not individually significant with 17 encoders.

Table 13: Association between diagnostics and suite success. Entries are Spearman's ρ with suite success over the 17 vision encoders. Bold marks a permutation $p < 0.05$ and † a Benjamini–Hochberg $q < 0.10$ within the cohort. n gives the number of encoders in each frozen cohort and in Twin+FT. *Statistics that depend on patch positions leave out MAE. Shape records which tercile of the diagnostic has the highest mean success on all three frozen cohorts ($\downarrow$ lowest, $\cap$ middle, $\uparrow$ highest), $\cup$ a middle tercile that is lowest on all three, and a dot no consistent pattern. The last column is the cross-benchmark rank stability of the diagnostic itself, its rank correlation between RoboCasa and RoboTwin across encoders. The object probe has no chance baseline (Section H.2). Appendix K lists all diagnostics.

Diagnostic	n	Casa ST	Casa MT	Twin MT	Twin+FT	Shape	Stability
Token statistics (frames only)							
Anisotropy (mean patch cosine)	17/24	+0.19	+0.21	−0.03	−0.36	·	0.60
Mean token norm	17/24	+0.02	+0.12	+0.44	+0.14	$\uparrow$	0.99
Neighbor cosine*	16/23	+0.31	+0.23	−0.09	−0.05	$\cup$	0.96
Participation ratio	17/24	+0.10	+0.14	+0.19	−0.03	$\cap$	0.89
Norm-histogram entropy	17/24	**+0.56**	+0.41	+0.43	+0.34	$\uparrow$	0.93
Token-norm variance	17/24	−0.34	−0.17	+0.01	−0.04	$\cap$	0.95
Within/between-frame variance ratio	17/24	−0.22	−0.34	−0.41	−0.03	$\downarrow$	0.98
Temporal drift	17/24	−0.06	+0.12	+0.34	+0.12	·	0.96
Temporal participation ratio	17/24	−0.10	+0.00	+0.15	−0.09	$\cap$	0.94
Temporal norm-histogram entropy*	16/23	+0.39	+0.48	+0.34	+0.02	$\uparrow$	0.71
Lag-1 autocorrelation	17/24	+0.06	−0.07	−0.19	+0.33	·	0.68
Low-frequency share*	16/22	+0.40	+0.30	+0.10	+0.25	·	0.97
Mid-frequency share*	16/22	**+0.55**	+0.45	+0.07	+0.31	·	0.94
High-frequency share*	16/22	−0.41	−0.32	−0.06	−0.28	·	0.97
Spectral entropy*	16/22	−0.44	−0.34	−0.08	−0.30	·	0.97
Readout probes (labels or actions)							
Inverse-dynamics (IDM) error	17/24	**−0.64**	**−0.64**	**−0.82**†	**−0.76**†	·	0.72
State-decoding error	17/24	**−0.68**	**−0.64**	**−0.69**†	**−0.71**†	·	0.85
Object-position error	–/24	–	–	**−0.70**†	**−0.58**†	$\downarrow$	–
Trained-policy diagnostic							
Action sensitivity to texture	–/24	–	–	−0.40	**−0.59**†	$\downarrow$	–

Finding 5 in detail: The two-frame IDM probe and the single-frame state probe rank vision encoders almost identically. Across the 17 vision encoders, IDM error and state error correlate at $\rho = 0.92$ on RoboCasa and 0.90 on RoboTwin. On RoboCasa, the nine encoders in the better half by IDM error are exactly the nine in the better half by state error. Neither probe is ahead on every cohort. This overlap is expected from the targets. The IDM target $a_{t+15} - s_t$ contains the current state, which a single frame already shows, and the policies have to read that state from images because they receive no proprioception. We therefore treat the two probes as one family of readouts of the robot's configuration and commanded motion. Whether the change between two frames adds anything beyond state would need single-frame and repeated-frame versions of the IDM probe, which we did not run.

The three Qwen3-VL-derived checkpoints are outliers for state decoding. The VLMs lead RoboTwin, with success ranks 1 to 3 among all 20 encoders, yet their state errors rank 15th, 19th, and 20th. Their IDM errors rank 4th, 5th, and 9th. Adding them to the cohort lowers the state association on RoboTwin from -0.69 to -0.12, while IDM stays at -0.78 (Figure 16). On RoboCasa, where the VLMs sit mid-table in success, both associations change by at most 0.15. The probes read the image-position states of one head frame, whereas the RoboTwin policy reads three views whose VLM tokens attend across views and have passed through a language model. The data do not separate these differences, so we leave the outlier status unexplained and list the camera ablation that would resolve it in Section J.4.

The readouts separate weak encoders better than they order strong ones. Figure 17 simulates selecting five of the 17 encoders. On every cohort, the five with the lowest IDM error exceed the average encoder by 4 to 12 points, and the five with the highest fall 11 to 14 points below it. Among encoders with low IDM error the order is less informative. On RoboCasa, the en-

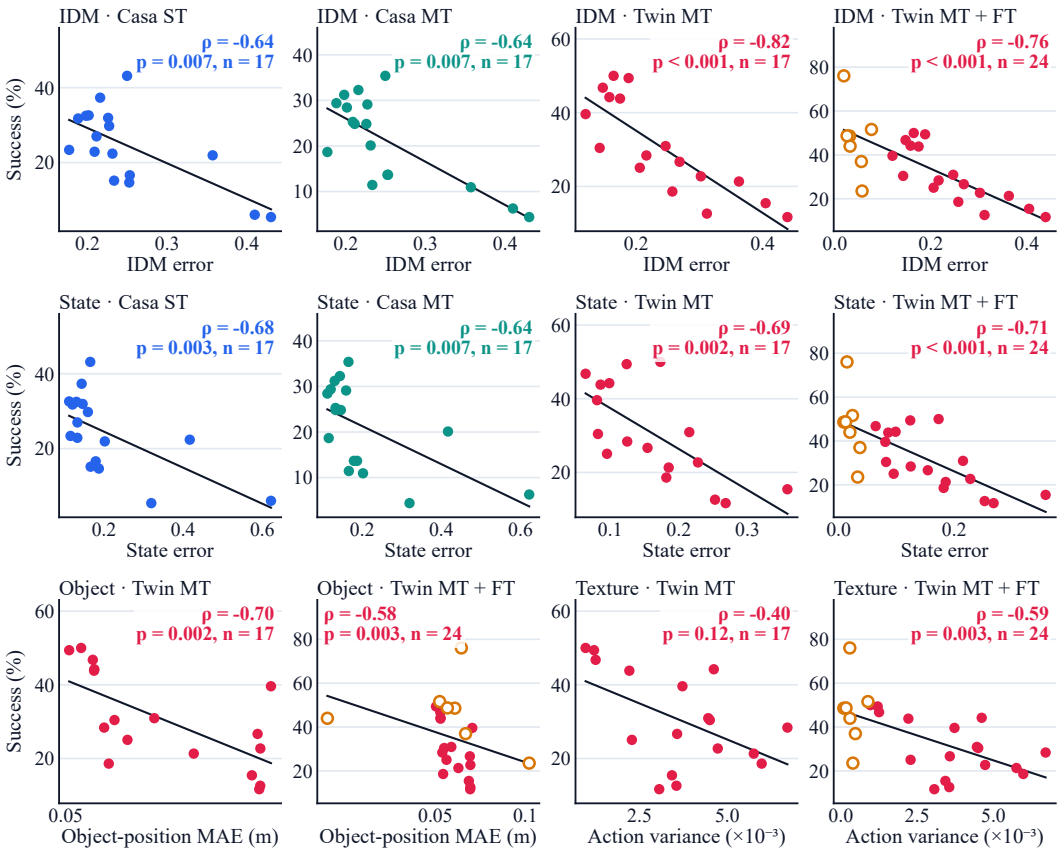


Figure 15: Readout probes and the policy diagnostic against success. Points are the 17 vision encoders; colors denote the cohort (blue: Casa ST, teal: Casa MT, rose: Twin MT), and hollow amber points in the Twin+FT panels are the fine-tuned RoboTwin checkpoints. Rows: IDM error, state error, and, on RoboTwin, object-position error and action sensitivity to texture. Each panel reports Spearman's ρ, its permutation p, and n; lines are least-squares fits.

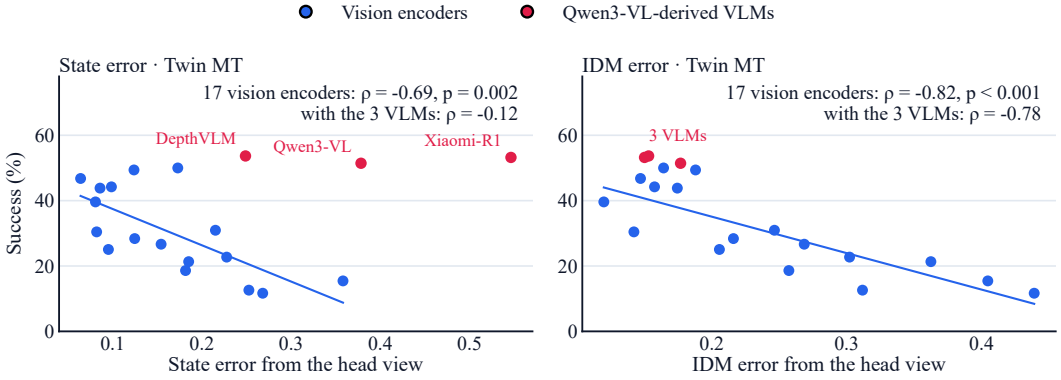


Figure 16: Readout probes with and without the VLMs on RoboTwin. Success against state error (left) and IDM error (right) for all 20 encoders. Blue points are vision encoders and rose points the three Qwen3-VL-derived VLMs; the line fits the 17 vision encoders, and the text compares the rank correlation with and without the VLMs.

coder with the lowest error, V-JEPA 2.1, ranks 9th and 11th in success, while the most successful encoder, VGGT-Ω, ranks 12th of 17 in IDM error.

Between families the readout associations are strong; within families they are weaker on RoboCasa. Averaging ranks within the five families that contain more than one vision encoder, family-level IDM error and family-level success correlate at $\rho = -0.90$ on all three cohorts. Af-

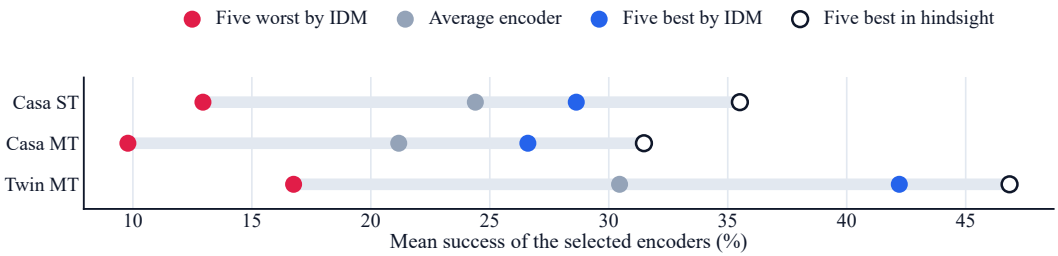


Figure 17: Selecting encoders by IDM error. For each frozen cohort (17 vision encoders), the mean success of the five encoders with the lowest and the highest IDM error, of all encoders, and of the five most successful encoders.

ter the family means are removed, IDM error ranks encoders within their families at -0.36 and -0.40 on RoboCasa and -0.80 on RoboTwin, and state error at -0.38, -0.43, and -0.64. A good part of each association therefore reflects differences between pretraining families. With five families these comparisons are exploratory.

The readouts are most informative on the tasks where encoders differ most. Within each cohort, the per-task correlation of IDM error with success grows with the spread of success across encoders on that task ($\rho = +0.20$, $+0.59$, and $+0.75$ on Casa ST, Casa MT, and Twin MT), and state error shows the same dependence ($+0.17$, $+0.71$, $+0.76$). Tasks that most encoders fail, or solve alike, carry little ranking information; where the encoders separate, the probes follow the separation.

Object-position errors are compressed into a narrow range. The object probe covers four RoboTwin tasks. Eleven vision encoders reach 5.0–6.0 cm mean absolute error per coordinate and the other six cluster between 6.5 and 6.6 cm, so the whole cohort spans 1.6 cm. Without a mean-predictor baseline we cannot tell how far any encoder is above chance. Its correlation with success (-0.70) appears in Table 13; it falls to -0.16 once the VLMs join, and fine-tuning raises object error in five of seven pairs, so no conclusion rests on it alone.

H.3 Token statistics

Token statistics need only frames, which makes them the cheapest screens. They describe how token norms, variance, and spatial frequencies are distributed; they do not test whether task variables can be decoded. Figure 18 shows the main statistics on every cohort.

Finding 7 in detail: Norm-distribution statistics keep their direction across frozen encoders but are weaker, and fine-tuning erases it. Norm-histogram entropy is positive on the three frozen cohorts ($\rho = +0.56$ [$+0.12$, $+0.86$], $+0.41$, $+0.43$), and its top tercile has the highest mean success on all three (Figure 19); only Casa ST reaches $p < 0.05$ ($p = 0.021$). The within/between-frame variance ratio is negative on the same cohorts (-0.22, -0.34, -0.41), and temporal norm-histogram entropy positive ($+0.39$, $+0.48$, $+0.34$). No other token statistic keeps one sign across the frozen cohorts with a magnitude above 0.2. Adding the seven fine-tuned checkpoints weakens all three ($+0.34$, -0.03, $+0.02$), while IDM error stays at -0.76. The one token statistic that becomes robust on Twin+FT, the within/between-trajectory variance ratio ($+0.54$, $q = 0.045$), has no association across frozen encoders (-0.05 to $+0.30$); it separates fine-tuned from frozen checkpoints rather than good from weak encoders. The entropy is computed with bins that span each frame's own range, so it is low both when the norms are nearly equal and when a few very large norms stretch the range. Three of the four lowest-entropy encoders, CLIP, RADIOv2.5, and InternViT, also have by far the largest norm variance, which fits the high-norm artifact tokens described for large vision transformers (Darcet et al., 2024). Whether such tokens affect the policy through the attention of Eq. (3) is a hypothesis; we measured neither attention weights nor the effect of removing the outliers.

The participation ratio has no monotone association with success. The participation ratio of the patch covariance, often called the effective rank, has been proposed as a label-free quality score (Garrido et al., 2023). Here its correlation with success stays between $+0.10$ and $+0.19$. On RoboCasa, the seven encoders with a ratio between 3.5 and 15 average 33.2% single-task and 28.6% multi-task

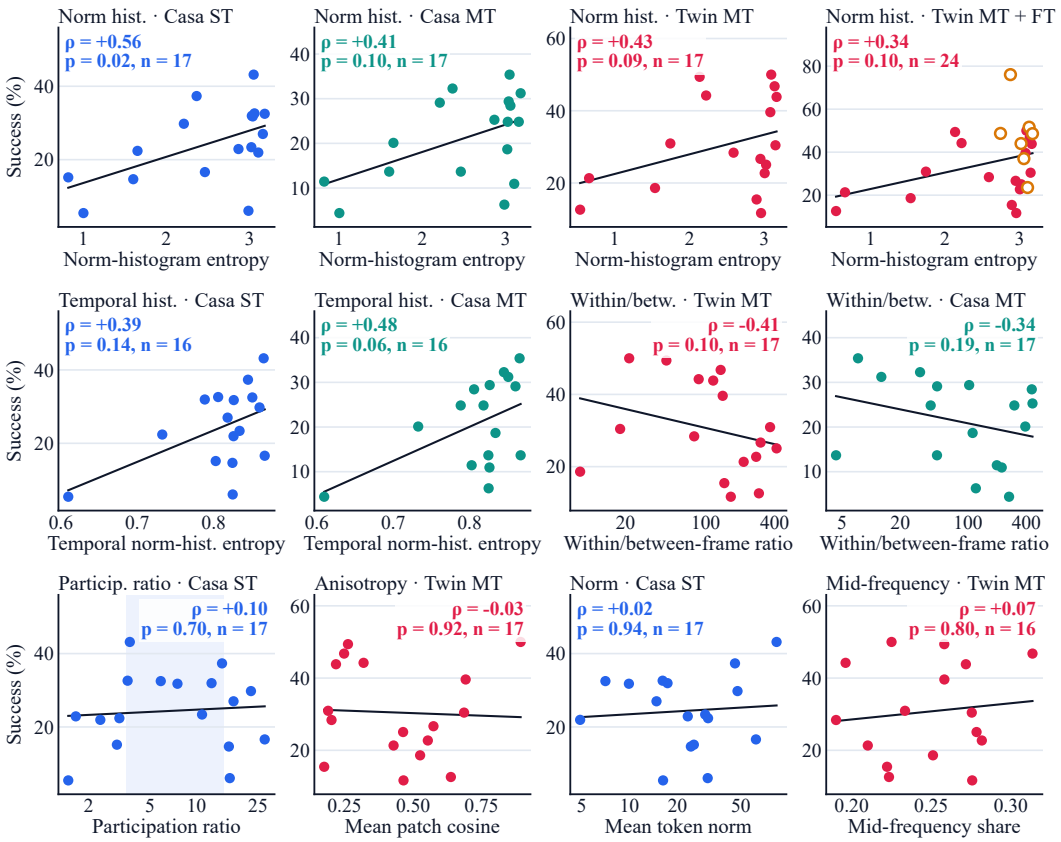


Figure 18: Token statistics against success. Same format as Figure 15. Top: norm-histogram entropy on every cohort; hollow amber points are the seven fine-tuned RoboTwin checkpoints, which cluster at high entropy across the whole range of success. Middle: the two statistics with the most consistent trends, temporal norm-histogram entropy and the within/between-frame ratio. Bottom: four widely used statistics without a consistent association; the shaded band on the participation ratio marks the exploratory RoboCasa range discussed in the text.

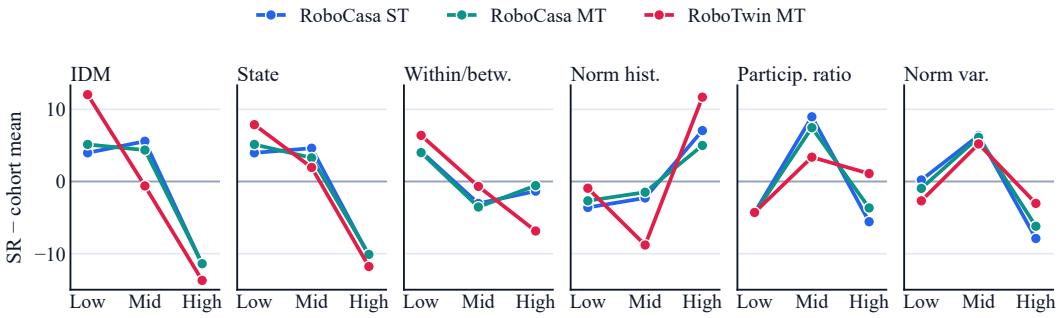


Figure 19: Mean success by diagnostic tercile. Encoders are split into thirds by each diagnostic; lines connect the mean success of the three thirds minus the cohort mean. Terciles contain five or six encoders.

success, against 18.2% and 16.0% for the other ten. The range was read off the RoboCasa data and does not separate the RoboTwin encoders (31.7% against 29.8%), so it is an exploratory grouping.

Stable statistics are not necessarily predictive. Several statistics order the encoders almost identically on both benchmarks: mean token norm has a cross-benchmark rank stability of 0.99, and the spectral statistics 0.97 to 0.98. Across the 49 diagnostics measured on both benchmarks, this stability is only weakly related to predictive strength ($\rho = 0.21$). Mean patch cosine, a common indicator of representational collapse, is weakly positive on RoboCasa and null on RoboTwin, and VGGT-Ω combines the most anisotropic tokens with the best RoboCasa success.

H.4 Standard vision benchmarks

Encoders are usually chosen by their scores on standard vision benchmarks. We collected published scores for the checkpoints we evaluated and kept one protocol per benchmark, so that the numbers within a benchmark are comparable. The LingBot-Vision paper (Fu et al., 2026) provides linear-probe ImageNet-1k classification, ADE20k and VOC segmentation, and NYUv2 depth; the DINOv3 paper (Siméoni et al., 2025) provides NAVI and SPair keypoint correspondence and DAVIS video segmentation; and the C-RADIOv4 report (Ranzinger et al., 2026) adds its correspondence scores. Nine to eleven checkpoints qualify: four sizes each of DINOv2 and DINOv3, SigLIP2, LingBot-Vision, and two sizes of C-RADIOv4. The other encoders lack scores under these protocols. We compare with multi-task success, the regime in which the model-scale study covers every size. Figure 20 shows three of the benchmarks, and Appendix I.2 lists all scores and correlations.

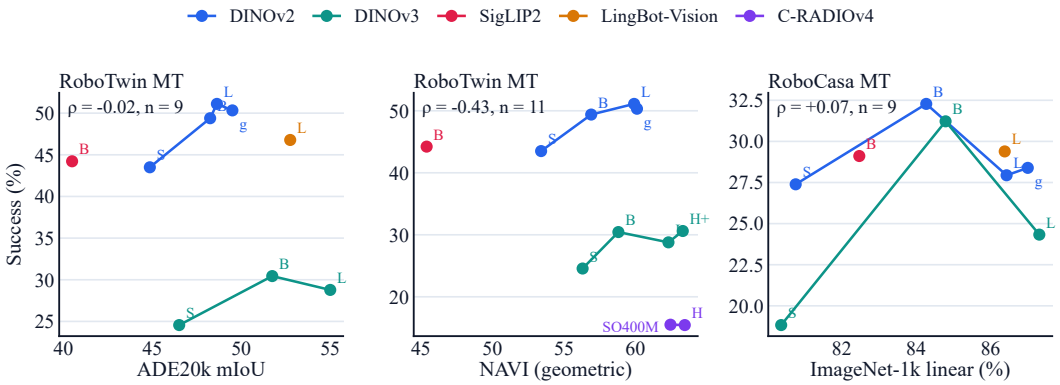


Figure 20: Standard vision benchmarks against multi-task success. Published scores of the evaluated checkpoints under one protocol per benchmark: ADE20k linear segmentation and ImageNet-1k linear classification from the LingBot-Vision paper, and NAVI geometric correspondence from the DINOv3 and C-RADIOv4 papers. Lines connect sizes of one family, and letters give the size. Each panel reports Spearman's ρ over the plotted checkpoints.

Standard vision benchmarks do not track multi-task success. No vision benchmark correlates with success at $p < 0.05$ on RoboCasa or RoboTwin (Table 17). ImageNet-1k accuracy is near zero on RoboCasa ($+0.07$ for the linear probe, $+0.02$ for k-NN) and weakly positive on RoboTwin ($+0.38$, $+0.35$). Segmentation and depth lie between -0.43 and $+0.10$, and the three correspondence and video benchmarks are negative on both (-0.27 to -0.55). With 9 to 11 checkpoints from mostly two families these correlations are exploratory, but none of them points to higher benchmark scores as a sign of higher success.

Benchmark gains from scale and from newer pretraining do not turn into success. Within DINOv2, ImageNet-1k accuracy rises from 80.8% to 87.0% from the small to the giant model, ADE20k from 44.9 to 49.5 mIoU, and NAVI from 53.4 to 60.1. RoboCasa multi-task success peaks at the base model (32.3%), and RoboTwin success stops rising after the large model (51.1% and 50.3%). DINOv3 beats DINOv2 at every matched size on ADE20k (by 1.7 to 6.4 mIoU), NYUv2, and NAVI (by 1.9 to 2.9), yet trails it by 19 to 22 points on RoboTwin and by 1 to 9 points on RoboCasa. C-RADIOv4 has NAVI and SPair scores among the highest in the table and the lowest success of all eleven checkpoints. The readout probes do not resolve the DINO comparison either. On RoboTwin, DINOv3 has lower IDM and state errors than DINOv2 (0.14 against 0.19, and 0.08 against 0.12) but lower success, consistent with the probes separating weak encoders better than they order strong ones. These benchmarks score single images of general scenes, and our results suggest they cannot stand in for a manipulation benchmark when choosing an encoder.

H.5 A trained-policy diagnostic

Action sensitivity to texture tracks success once fine-tuned checkpoints are included. On RoboTwin, policies whose predicted actions change less across background textures tend to succeed more, weakly over the 17 frozen encoders ($\rho = -0.40$, $p = 0.12$) and robustly with the fine-tuned

checkpoints (-0.59, $p = 0.003$, $q = 0.029$). The fine-tuned policies are both the most successful and the least sensitive, so the pooled value mostly reflects that group difference. The seven least sensitive frozen policies average 34.6% success and the other ten 27.6%. The diagnostic needs a trained policy and cannot be used to choose an encoder before training. A small change of the predicted chunk at the initial scene is not closed-loop robustness, and a policy that never moves would score as insensitive, so we read it only together with success.

H.6 Combining diagnostics

Combinations can serve three goals, which we evaluate separately: *prediction* of each encoder's success (leave-one-encoder-out regression on diagnostic ranks, reported as mean absolute error), *ranking* of encoders (Spearman's ρ between success and an unweighted average of diagnostic ranks), and *selection* of a few encoders (mean success of the top five). Table 14 uses the same 16 encoders in every row, those for which all of its diagnostics are defined.

Table 14: Predicting and ranking success from diagnostics. Left: leave-one-encoder-out mean absolute error (percentage points) of a linear model on rank-transformed diagnostics. Right: Spearman's ρ between success and the average of oriented diagnostic ranks. All rows use the same 16 encoders per cohort; bold marks the best value in each column. Differences between correlations were not tested. "All five" combines IDM, state, norm-histogram entropy, temporal norm-histogram entropy, and mid-frequency share.

	LOO error (pp) ↓			Rank-average ρ ↑		
Diagnostics	**Casa ST**	**Casa MT**	**Twin MT**	**Casa ST**	**Casa MT**	**Twin MT**
Cohort mean	9.36	9.09	11.99	–	–	–
IDM	6.99	**5.54**	**6.83**	0.65	0.66	**0.81**
Norm-histogram entropy	7.83	7.69	10.74	0.56	0.43	0.44
IDM + mid-frequency share	**6.80**	6.07	7.08	0.63	0.57	0.57
IDM + norm-histogram entropy	6.82	5.93	7.31	**0.75**	0.67	0.70
IDM + temporal norm-histogram entropy	7.36	5.91	7.17	0.68	**0.74**	0.64
IDM + within/between ratio	8.19	6.32	7.63	0.58	0.72	0.77
IDM + state	8.08	6.50	7.41	0.69	0.67	0.75
All five	9.36	8.00	9.07	0.74	0.70	0.63

Finding 8 in detail: No combination improves on a single readout probe consistently across cohorts. IDM alone lowers the prediction error below that of the cohort mean on every cohort, by 25% to 43%. A second diagnostic lowers it further only on Casa ST, and only slightly (from 6.99 to 6.80 points with the mid-frequency share); on Casa MT and Twin MT every pair is worse than IDM alone, and the five-diagnostic model is worse everywhere. Adding state to IDM does not help, because the two carry nearly the same ranking. For ranking, IDM paired with norm-histogram entropy or with state has the largest worst-case correlation over the three cohorts (0.67, against 0.65 for IDM alone), a difference well within the noise of 16 encoders. Selection on the same data inflates such gains, so we also held out one cohort at a time. Choosing the best single diagnostic on two cohorts and testing it on the third picks a readout probe every time, IDM once and state twice (mean held-out $\rho = 0.66$), and choosing the best pair in the same way gives 0.64. Median splits by IDM error and a partner diagnostic (Figure 21, right) show no benefit from the partner among encoders with low IDM error; with two to six encoders per cell we do not interpret the other cells.

H.7 Frozen properties and adaptation

We recompute the diagnostics on the fine-tuned RoboTwin checkpoints. Table 15 lists the paired changes, and Figure 22 shows how the checkpoints move.

Finding 5 in detail: Fine-tuning lowers IDM and state errors in every pair and compresses them into a narrow range. Fine-tuning lowers IDM error by 59–89% and state error by 80–92% in all seven RoboTwin pairs, and action sensitivity to texture falls in every pair as well, whereas object error moves in both directions. These pairs are the one comparison in which the representation changes while the rest of the setup stays fixed, and in every pair success rises as both probe errors

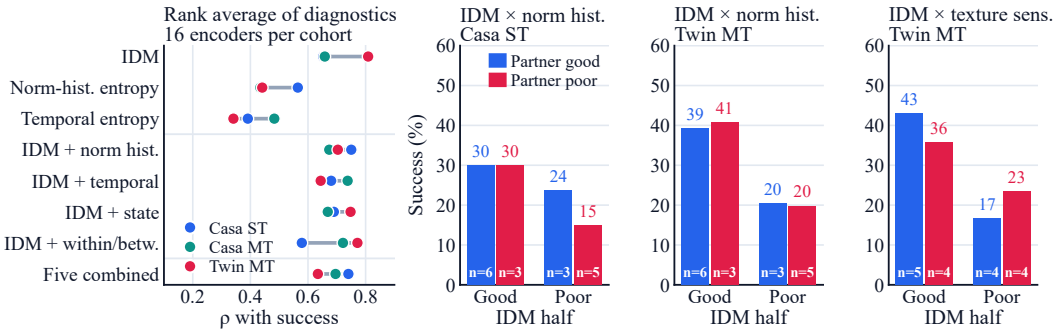


Figure 21: Combining diagnostics. Left: Spearman's ρ between success and the rank average of each diagnostic set on the three cohorts, for the 16 encoders of Table 14; the grey bar spans the lowest and highest cohort. Right: mean success in the four cells of median splits by IDM error and by a partner diagnostic, with the number of encoders in each cell. These are observed conditional means without an interaction test.

Table 15: Fine-tuning shifts on RoboTwin. Success is frozen → fine-tuned (%). Diagnostic entries are relative changes after fine-tuning; teal marks a decrease and rose an increase. Lower is better for every diagnostic in the table.

Encoder	Success	Δ (pp)	IDM	State	Texture	Object
VGGT-Ω	50.0 → 76.1	+26.1	−89%	−92%	−70%	+20%
SPA	11.7 → 37.0	+25.3	−87%	−86%	−85%	−90%
VC-1	25.1 → 44.0	+18.9	−85%	−80%	−87%	−60%
DINOv3	30.4 → 48.6	+18.2	−79%	−89%	−98%	+9%
C-RADIOv4	15.4 → 23.6	+8.1	−86%	−91%	−89%	+59%
SigLIP2	44.2 → 48.7	+4.5	−84%	−88%	−96%	+5%
DINOv2	49.4 → 51.6	+2.2	−59%	−81%	−27%	+3%

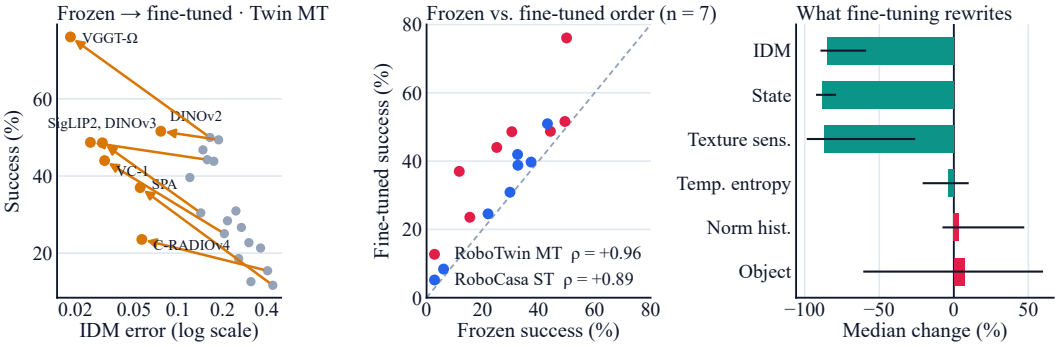


Figure 22: What changes with fine-tuning. Left: each arrow moves one RoboTwin encoder from its frozen to its fine-tuned checkpoint (log scale); grey points are the 17 frozen vision encoders. Middle: fine-tuned against frozen success for the seven pairs on RoboTwin multi-task and RoboCasa single-task; the dashed line marks equal success. Right: median relative change of each diagnostic across the RoboTwin pairs, with the range as a whisker; teal marks decreases and rose increases.

fall. After fine-tuning, all seven checkpoints have IDM errors between 0.018 and 0.076, below every frozen encoder, while their success spans 24% to 76%. Among the fine-tuned checkpoints alone, IDM error therefore ranks success only weakly ($\rho = -0.46$, $p = 0.30$), and the pooled Twin+FT correlation (-0.76) is carried mainly by the separation between frozen and fine-tuned checkpoints. The frozen order survives fine-tuning much better: fine-tuned success follows frozen success at $\rho = 0.96$ on RoboTwin multi-task and 0.89 on RoboCasa single-task (Figure 22, middle), and frozen IDM error correlates with fine-tuned success at -0.64 and -0.61. The size of the gain is a separate question. Frozen success does not predict it ($\rho = -0.14$ on RoboTwin, $+0.50$ on RoboCasa), and frozen norm-histogram entropy goes with it on both benchmarks ($+0.75$ and $+0.82$). With seven pairs per benchmark, all correlations in this paragraph are exploratory.

Properties that distinguish frozen encoders need not move in the same direction during adaptation. Temporal norm-histogram entropy, positive across frozen encoders, falls in six of the seven pairs, although success rises in all seven. The participation ratio converges: the frozen checkpoints span 3.2 to 23.6 and the fine-tuned ones 5.7 to 7.7. Mean patch norm falls in five pairs, from 45.8 to 10.0 for DINOv2 and from 90.2 to 26.6 for VGGT-Ω, and rises from 30.6 to 87.0 for C-RADIOv4. A property that separates good from weak frozen checkpoints is therefore not, by itself, a direction in which fine-tuning moves an encoder.

H.8 How general are the associations?

Which diagnostics matter changes little with the training regime and more with the benchmark. Figure 23 places each diagnostic by its correlation with success in two settings. Between RoboCasa single-task and multi-task the profiles nearly coincide (Pearson $r = 0.94$); between RoboCasa and RoboTwin they agree at $r = 0.44$. The success rankings of the 17 vision encoders agree at $\rho = 0.93$ across regimes and at 0.72 to 0.83 across benchmarks. IDM error, state error, the two norm-histogram entropies, and the within/between ratio keep their sign in both comparisons.

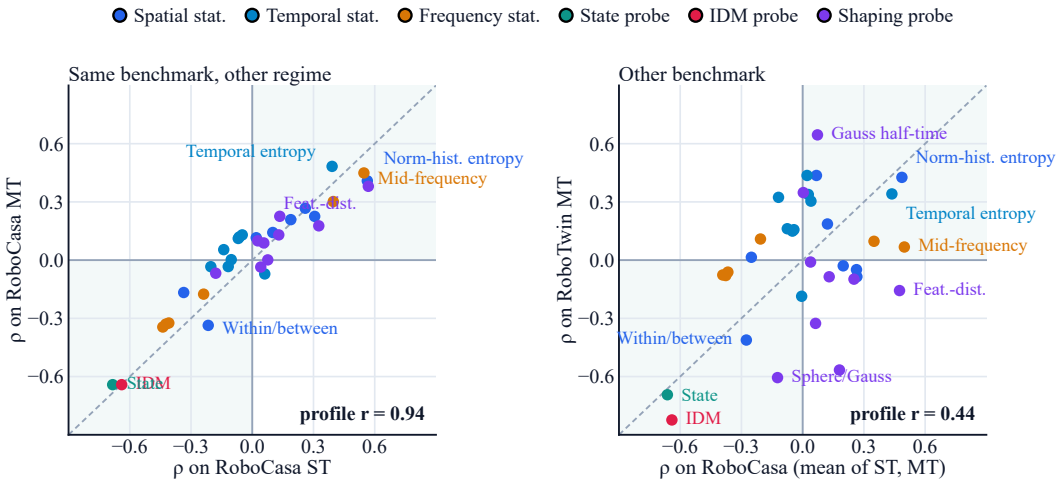


Figure 23: Association profiles compared across settings. Each point is one diagnostic, placed by its Spearman correlation with success in two settings and colored by diagnostic group. Shaded quadrants hold diagnostics whose sign agrees in both settings. Left: RoboCasa single-task versus multi-task. Right: RoboCasa versus RoboTwin.

The consistent diagnostics keep their direction task by task. Figure 24 recomputes each correlation within every task of the three frozen cohorts and orients it so that positive is the expected direction. IDM error and temporal norm-histogram entropy take the expected sign in all 36 task-level correlations; state error, norm-histogram entropy, and the within/between ratio in 35; and texture sensitivity on 11 of the 12 RoboTwin tasks. These counts measure direction, not strength, and they are not independent: the 24 RoboCasa correlations cover 12 tasks twice. In strength the readouts stand apart, with median per-task correlations of 0.60 for IDM and 0.59 for state, against 0.42 for norm-histogram entropy.

I Additional Analyses

I.1 Distribution-shaping probe and feature-distance preservation

The shaping probe fits a weak adapter that maps each frame to a point in $\mathbb{R}^3$ and pulls the resulting cloud toward either a standard Gaussian or a uniform sphere of radius $\sqrt{3}$ (Appendix D.6 and Figure 25). Each of its associations holds on one benchmark only, so we report them here rather than in Section H.

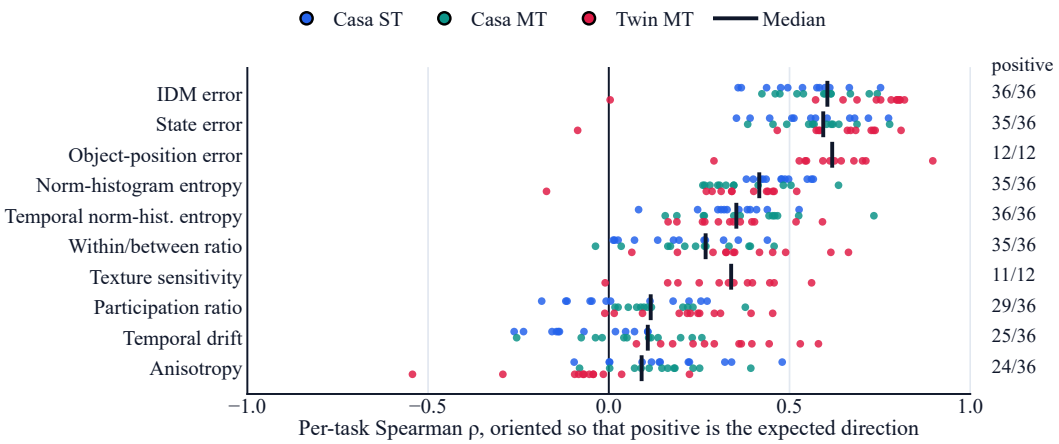


Figure 24: Per-task direction consistency. Each dot is the Spearman correlation between a diagnostic and success on one task, oriented so that positive is the expected direction. Colors denote the cohort, the black tick marks the median, and the right column counts correlations with a positive value.

Loss summaries. The two targets have different floors. The Gaussian discrepancy can approach zero, whereas the sphere's uniformity term has a reference value of 0.268 for points spread uniformly on the sphere, and the final sphere discrepancy on RoboTwin lies between 0.28 and 0.37 for 16 of the 17 vision encoders. Loss ratios are therefore not comparable across targets. Within a target, the Gaussian half-time ($\rho = +0.65$, $q = 0.08$) and the ratio of sphere to Gaussian loss area (-0.61, $p = 0.011$) are associated with RoboTwin success and null on RoboCasa. We report these associations without an interpretation and draw no conclusion about Gaussian regularization.

Feature-distance preservation. This summary is the Spearman correlation between pairwise distances of the pooled, randomly projected frame features and pairwise distances of the adapter's outputs after fitting the Gaussian target. It measures how much of the feature-space distance ranking a weak reshaping keeps, not 3D scene structure. It is computed on mean-pooled frame features, the reduction that removes most of the success in Section G.2, so it describes a pooled view of the encoder rather than the tokens the policy reads. It reaches $+0.57$ on Casa ST ($p = 0.019$) and $+0.38$ on Casa MT but -0.16 on Twin MT, and it is positive on every RoboCasa task and negative on every RoboTwin task. The benchmarks differ in cameras, embodiment, action space, tasks, and randomization, so the contrast cannot be traced to one factor.

Task sharing. On RoboCasa, encoders with higher feature-distance preservation lose more from single-task to multi-task training in absolute terms ($\rho = -0.64$, $p = 0.007$) and keep a smaller fraction of their single-task success ($\rho = -0.42$, $p = 0.09$). The six with the highest preservation keep 78% of their single-task success and the six with the lowest keep 90%, while single-task success itself does not predict the fraction kept ($\rho = 0.02$). Because the multi-task policy also receives fewer updates per task, this concerns sharing under an equal per-policy budget.

Fine-tuning. The tokens of every fine-tuned pair become faster to fit to both targets (the Gaussian loss area falls by 18% to 66%), although across frozen encoders faster fitting does not go with higher success on either benchmark. Feature-distance preservation falls in five of the seven pairs, most where the gains are largest ($\rho = -0.79$). Frozen preservation correlates with the gain at $+0.86$ on RoboTwin and $+0.46$ on RoboCasa; with seven pairs this is exploratory.

I.2 Standard vision benchmarks

Table 16 lists the published scores behind Section H.4, and Table 17 their correlations with multi-task success. Scores are copied from the cited papers. Within a column the evaluation protocol is the same, but the columns come from different evaluations and should not be compared with each other. Success values of non-default sizes come from the model-scale study (Tables 30 to 33).

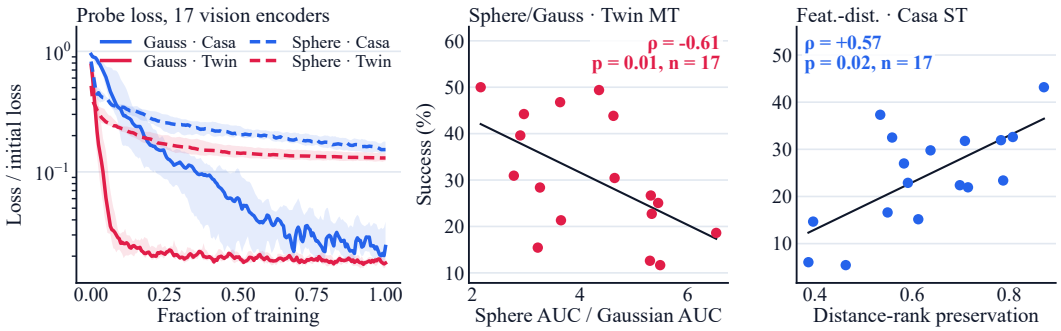


Figure 25: Distribution-shaping probe. Left: median loss relative to its initial value across the 17 vision encoders, with interquartile bands. The Gaussian and sphere targets have different floors, so their curves are not comparable in level. Middle: on RoboTwin, the ratio of sphere to Gaussian loss area against success. Right: on RoboCasa single-task, feature-distance preservation against success.

Table 16: Standard vision benchmarks of the evaluated checkpoints. Reported results of frozen encoders, taken from the literature under one protocol per column: ImageNet-1k, ADE20k, VOC, and NYUv2 from the linear-probe evaluation of the LingBot-Vision paper (Fu et al., 2026); NAVI, SPair, and DAVIS from the DINOv3 paper (Siméoni et al., 2025), with the C-RADIOv4 correspondence results from its own paper (Ranzinger et al., 2026). A dash marks a checkpoint without a result under that protocol. The last two columns are the multi-task success (%) of the same checkpoint in HIVE-Bench.

Encoder	Size	IN-1k lin.	IN-1k k-NN	ADE20k	VOC	NYUv2↓	NAVI	SPair	DAVIS	Casa MT	Twin MT
DINOv2	S	80.8	79.1	44.9	82.3	0.447	53.4	51.6	73.6	27.4	43.5
DINOv2	B	84.3	82.2	48.2	83.8	0.429	56.9	57.1	72.9	32.3	49.4
DINOv2	L	86.4	83.8	48.6	83.5	0.411	59.9	57.0	73.4	27.9	51.1
DINOv2	g	87.0	83.7	49.5	83.1	0.372	60.1	56.1	73.6	28.4	50.3
DINOv3	S	80.4	79.3	46.5	84.3	0.405	56.3	50.4	72.7	18.8	24.6
DINOv3	B	84.8	83.2	51.7	88.2	0.371	58.8	57.2	77.2	31.2	30.4
DINOv3	L	87.3	85.3	55.0	88.1	0.351	62.3	61.2	79.9	24.3	28.8
DINOv3	H+	–	–	–	–	–	63.3	56.3	79.3	24.8	30.6
SigLIP2	B	82.5	79.5	40.5	73.5	0.539	45.4	32.8	63.2	29.1	44.2
LingBot-Vision	L	86.4	83.6	52.8	87.4	0.310	–	–	–	29.4	46.8
C-RADIOv4	SO400M	–	–	–	–	–	62.4	60.0	–	4.8	15.5
C-RADIOv4	H	–	–	–	–	–	63.4	60.6	–	6.3	15.4

Table 17: Vision benchmarks against multi-task success. Spearman ρ between each benchmark (oriented so that higher is better) and suite success over the checkpoints of Table 16, with the two-sided permutation p and the number of checkpoints n; none reaches $p < 0.05$.

Benchmark	n	ρ Casa MT	p	ρ Twin MT	p
ImageNet-1k linear (%)	9	+0.07	0.88	+0.38	0.31
ImageNet-1k k-NN (%)	9	+0.02	0.98	+0.35	0.36
ADE20k mIoU	9	+0.10	0.81	−0.02	0.98
VOC mIoU	9	+0.10	0.81	−0.43	0.25
NYUv2 RMSE	9	+0.00	1.00	−0.18	0.64
NAVI (geometric)	11	−0.55	0.08	−0.43	0.19
SPair (semantic)	11	−0.31	0.36	−0.45	0.16
DAVIS $J\&F$	9	−0.27	0.48	−0.29	0.44

J Extended Discussion and Limitations

Within the scope of our experiments, the results support one central statement. For vision encoders whose policies read images without proprioception, readout probes that decode the robot's state or its commanded motion track success on every cohort, more closely than token statistics, whereas published vision-benchmark scores do not track it; fine-tuning raises every encoder but largely keeps the frozen order, and the size of its gain needs a separate measurement. This appendix extends Section 6 and does not add evidence. It discusses what the results imply for choosing encoders and building benchmarks, lists hypotheses that need interventions, and states its limits.

J.1 How the interface changes what a diagnostic means

A diagnostic describes tokens, but success depends on how the policy reads them. Three results make this concrete. First, the same frozen DINOv2 and DINOv3 features lose most of their success when each view is averaged into one token (Finding 2), so evaluations built on pooled embeddings measure a different interface from the one patch-conditioned policies use. Second, the three Qwen3-VL-derived VLMs lead RoboTwin while decoding the robot's state poorly from the head view, so the state probe misranks them, although it ranks the vision encoders well (Section H.2). Their policy tokens mix the three views through the language model's attention, which a head-view probe does not see. Third, reading an intermediate VLM layer changes success by up to eleven points in some settings and by less than one in others (Section G.6). The vision-benchmark comparison points the same way from the other side: scores measured on single images under linear probes did not track success, even where they improved steadily with scale (Section H.4). A representation claim should therefore name the layer, the token interface, and the cameras it was measured under, and a diagnostic should be read against the interface of the policy it is meant to screen for.

J.2 Selection is not optimization

The diagnostics answer a selection question: which frozen encoder is likely to control well under a fixed interface. The readout probes answer it best, and mostly by exclusion. The five vision encoders with the highest IDM error fall 11 to 14 points below average, whereas the five with the lowest error exceed it by only 4 to 12 points (Section H.2). Adaptation is a different question. Fine-tuning moves every checkpoint to low IDM and state errors, so these errors measured after fine-tuning barely order the checkpoints, and some properties that distinguish frozen encoders move the other way during successful adaptation (Finding 5). The gains rarely reorder the seven encoders (Finding 3), so frozen evaluation says which encoders are likely to end best, while the size of each gain has to be measured. A correlate of success across checkpoints is also not a training target; turning one into an objective requires an intervention showing that the property itself matters.

J.3 What the results imply for choosing encoders

The results suggest an order of work rather than a validated protocol. The interface comes first: keep separate patch tokens, fix the cameras, and for VLMs try an intermediate layer as well as the final one. Among frozen candidates, a readout probe trained on head-camera frames of the target data is the most useful cheap measurement, mainly for setting aside encoders with high error. It orders the strongest candidates poorly, and its state version misranks the VLMs. Token statistics are weaker and better used to flag anomalies, such as very low norm-histogram entropy, than as filters; DINOv2 and SigLIP2 would fail a strict entropy threshold. Combining several diagnostics did not beat a single readout probe under held-out selection (Finding 8), and published vision-benchmark scores did not help at all. The final choice still needs closed-loop runs on a shortlist, and whether an encoder is worth fine-tuning needs fine-tuning runs, because frozen success does not predict the gain, which ranged from 1 to 26 points across our pairs.

J.4 Hypotheses worth testing

The associations suggest several hypotheses that require interventions this study did not perform.

Command readouts beyond state. The IDM probe ranks encoders like the state probe, so our data cannot tell whether decoding the change between frames matters beyond reading the current state. If it does, pretraining objectives that tie observation differences to actions or latent actions should help. Latent inverse and forward dynamics over 3D features (Liang et al., 2026), interleaved action–effect memory (Zhou et al., 2026b), and transition-aware targets in a video-pretrained latent space (Lin et al., 2026) are such objectives, and reports that video prediction helps world-action models mainly through the representation it shapes (Yuan et al., 2026), and that semantic latents recover actions better than reconstruction latents (Nilaksh et al., 2026), fit the idea. A direct test would compare the IDM probe with its single-frame and repeated-frame versions and train encoders that differ only in such an objective.

Proprioception. No policy in HIVE-Bench receives proprioception, so reading the robot's configuration from images is part of every policy's job. The state association may weaken once proprioception is supplied, because the policy then no longer needs the images for it. Retraining a subset of encoders with proprioception, and comparing their ranking with the current one, would test this.

High-norm tokens. The encoders with the lowest norm-histogram entropy also have the largest norm variance, consistent with the high-norm artifact tokens of large vision transformers (Darcet et al., 2024), and suppressing task-irrelevant high-norm tokens has been reported to help robot learning (Wang et al., 2026c). Whether these tokens hurt a patch-conditioned policy could be tested by measuring attention weights in Eq. (3) and by removing or renormalizing the outliers of one encoder while holding everything else fixed.

VLMs on RoboTwin. The VLMs lead RoboTwin although they decode the robot's state poorly from the head view. Their policies may rely on the wrist views, on the cross-view attention inside the language model, or on information the state probe does not capture. A camera ablation within one benchmark, with and without wrist views and with views encoded separately, would separate these explanations on the same tasks.

J.5 Implications for benchmark design

Four results bear on how representation benchmarks for manipulation are built. Mid-ranked encoders reorder between suites while the extremes do not (Finding 1), so small differences on one suite should not be read as a general order. The readout probes track success most closely on the tasks where encoders differ most (up to $\rho = +0.75$ between the per-task strength of IDM or state decoding and the spread of success), so suites dominated by tasks that every encoder fails or solves understate differences between representations. Because a single mean embedding supports far less success than separate patch tokens, benchmarks should evaluate encoders through the interface that the target policies use. And because standard vision benchmarks did not track success, they cannot replace closed-loop evaluation.

J.6 Scope and limitations

The evidence comes from two simulated suites of twelve tasks each, twenty default encoders, of which the 17 vision encoders enter the diagnostic analyses, and seven fine-tuned pairs per benchmark. All policies share one action head, one recipe per benchmark, and evaluation of the final checkpoint, so we do not know how the rankings depend on the learner or on training length. Apart from the mean-pooling control and the paired fine-tuning changes, every result is a correlation across encoders, and the encoders are not independent samples: they share families, teachers, and data, and the three VLMs share one base model. The two benchmarks differ in embodiment, action space, cameras, tasks, randomization, and how much of each chunk is executed, so differences between them are not attributed to any single factor. The policies receive no proprioception, the probes read only the head camera, readout probes differ in capacity, and probe errors are validation errors on the split that also selected the epoch. Token statistics describe the first 100 steps of each sampled demonstration, and the norm-histogram entropy partly measures outliers because its bins follow each frame's range. The vision-benchmark scores are taken from the literature and cover 9 to 11 checkpoints from few families. Each configuration is trained once, so the reported deviations cover evaluation noise but not the variation between training runs, and suite uncertainties are propagated from task-level deviations rather than computed from per-seed suite scores. Real-robot transfer, other embodiments, and the

interventions above are the natural next steps, and HIVE-Bench provides the shared policy, protocols, and diagnostics needed to run them on new encoders and benchmarks.

K Complete Results

This section lists every closed-loop result and every diagnostic value behind Sections G and H. All tables and figures are generated from one result export. A suite score is the twelve-task mean of the exported task cells; every exported suite average agrees with its task cells to rounding.

Table 18: Task identifiers. RoboCasa-GR1 tasks (C) and RoboTwin 2.0 tasks (W) used in every per-task table. "PnP" denotes pick-and-place; the RoboCasa single-task data-scale sweep uses C1–C4.

ID	RoboCasa-GR1 task	ID	RoboTwin 2.0 task
C1	PnP Bottle To Cabinet Close	W1	Adjust Bottle
C2	PnP Can To Drawer Close	W2	Beat Block Hammer
C3	PnP Milk To Microwave Close	W3	Click Alarmclock
C4	PnP Novel From Cuttingboard To Basket	W4	Handover Block
C5	PnP Novel From Cuttingboard To Pan	W5	Move Playingcard Away
C6	PnP Novel From Placemat To Basket	W6	Open Laptop
C7	PnP Novel From Placemat To Plate	W7	Place Can Basket
C8	PnP Novel From Plate To Cardboardbox	W8	Rotate QRcode
C9	PnP Novel From Plate To Plate	W9	Stamp Seal
C10	PnP Novel From Tray To Plate	W10	Turn Switch
C11	PnP Novel From Tray To Pot	W11	Place Burger Fries
C12	PnP Novel From Tray To Tieredbasket	W12	Lift Pot

K.1 Task-level success

Tables 19, 21, and 23 give per-task success for every frozen encoder, every layer-16 VLM variant, every fine-tuned encoder, and the global-average-pooling readout discussed in Section G.2. To keep the page format, each table is split in two: tasks 1–6, then tasks 7–12 with the suite average.

Table 19: RoboCasa-GR1 single-task success per task. Entries are mean success (%) over three evaluation seeds, with the standard deviation across them in small type; task IDs follow Table 18. Avg. is the twelve-task mean; its ± term propagates the per-task deviations as $\sqrt{\sum_k s_k^2}/12$, which assumes independent task errors. **First**, **second**, and **third** are marked among the frozen encoders in each column. Encoder names are colored by family as in Table 2. This part lists tasks C1–C6; Table 20 continues with the remaining tasks and the suite average.

Encoder	C1	C2	C3	C4	C5	C6
Frozen encoders						
VGGT-Ω	48.0 ±6.5	**74.0 ±4.3**	**50.0 ±5.9**	**39.3 ±2.5**	**59.3 ±2.5**	**32.0 ±4.3**
LingBot-Vision	49.3 ±3.8	**68.7 ±2.5**	42.0 ±4.9	**30.0 ±3.3**	38.0 ±1.6	**19.3 ±0.9**
SPA	36.7 ±5.7	45.3 ±7.4	46.7 ±10.4	6.7 ±1.9	15.3 ±0.9	6.0 ±0.0
DINOv2	**60.7 ±5.0**	61.3 ±3.4	**49.3 ±6.2**	**26.7 ±0.9**	**42.7 ±3.4**	**16.0 ±1.6**
DINOv3	53.3 ±2.5	60.0 ±5.7	47.3 ±6.6	17.3 ±5.7	33.3 ±7.4	15.3 ±3.4
MAE	**55.3 ±6.2**	64.7 ±0.9	40.7 ±4.1	14.0 ±1.6	5.3 ±5.0	1.3 ±0.9
V-JEPA 2.1	44.0 ±4.3	54.7 ±0.9	34.7 ±4.1	1.3 ±0.9	30.7 ±3.4	9.3 ±3.4
Xiaomi-Robotics-1	40.7 ±2.5	63.3 ±8.1	44.0 ±8.6	14.0 ±2.8	17.3 ±3.4	5.3 ±2.5
DepthVLM	48.7 ±8.2	52.7 ±2.5	37.3 ±3.4	22.7 ±4.1	30.7 ±8.2	14.0 ±4.3
Qwen3-VL	33.3 ±0.9	55.3 ±7.4	24.7 ±7.4	12.0 ±1.6	28.0 ±9.1	8.0 ±4.3
SigLIP2	48.0 ±1.6	61.3 ±6.8	44.0 ±6.5	15.3 ±3.4	38.7 ±8.1	8.7 ±5.2
InternViT	43.3 ±0.9	52.0 ±1.6	47.3 ±6.8	12.0 ±2.8	20.7 ±3.8	6.7 ±1.9
SigLIP	32.7 ±4.1	39.3 ±2.5	28.7 ±6.6	8.0 ±1.6	23.3 ±4.1	5.3 ±2.5
CLIP	28.7 ±9.0	39.3 ±10.0	26.7 ±4.1	6.0 ±4.3	14.0 ±2.8	6.7 ±1.9
VC-1	**59.3 ±4.1**	**71.3 ±5.7**	**56.7 ±2.5**	12.0 ±1.6	41.3 ±0.9	8.0 ±3.3
Voltron	48.0 ±6.5	60.0 ±4.3	43.3 ±7.7	21.3 ±4.7	**48.7 ±3.4**	**19.3 ±2.5**
Theia	46.7 ±6.8	58.0 ±4.3	36.7 ±3.4	19.3 ±0.9	24.7 ±4.1	12.7 ±1.9
C-RADIOv4	14.0 ±5.7	27.3 ±3.4	16.7 ±6.6	0.0 ±0.0	3.3 ±3.4	0.0 ±0.0
RADIOv2.5	8.7 ±4.1	18.7 ±3.4	13.3 ±6.2	3.3 ±1.9	4.7 ±2.5	0.7 ±0.9
ViT	29.3 ±1.9	33.3 ±9.4	37.3 ±4.1	4.7 ±0.9	11.3 ±0.9	3.3 ±0.9
VLM layer 16						
Xiaomi-Robotics-1 L16	52.7 ±3.8	68.7 ±6.2	40.7 ±6.2	19.3 ±5.0	44.0 ±8.6	16.0 ±5.7
DepthVLM L16	44.0 ±3.3	60.7 ±6.8	48.7 ±6.2	27.3 ±6.6	44.7 ±8.2	24.0 ±5.9
Qwen3-VL L16	48.7 ±3.8	61.3 ±5.0	34.0 ±7.1	28.7 ±2.5	40.7 ±9.8	20.7 ±5.7
Fine-tuned encoders						
VGGT-Ω FT	72.7 ±3.8	78.0 ±4.3	70.0 ±9.9	44.7 ±3.4	58.0 ±2.8	43.3 ±0.9
DINOv2 FT	64.7 ±3.8	76.7 ±3.4	46.7 ±8.4	40.0 ±4.3	46.0 ±2.8	32.0 ±2.8
DINOv3 FT	71.3 ±2.5	68.0 ±1.6	50.7 ±5.2	34.7 ±4.1	55.3 ±5.0	32.0 ±4.9
SigLIP2 FT	43.3 ±7.7	65.3 ±0.9	41.3 ±2.5	20.0 ±2.8	42.7 ±4.1	16.7 ±2.5
VC-1 FT	54.0 ±5.7	70.0 ±8.6	46.7 ±2.5	37.3 ±0.9	57.3 ±0.9	29.3 ±3.4
SPA FT	50.7 ±8.1	40.7 ±5.7	40.0 ±5.7	28.0 ±5.7	38.7 ±2.5	10.0 ±1.6
C-RADIOv4 FT	29.3 ±5.7	28.7 ±6.6	18.7 ±3.4	0.0 ±0.0	9.3 ±0.9	1.3 ±1.9

Table 20: Continued from Table 19. Tasks C7–C12 and the suite average.

Encoder	C7	C8	C9	C10	C11	C12	Avg.
Frozen encoders							
VGGT-Ω	**14.7** ±1.9	14.0 ±2.8	**57.3** ±6.2	**49.3** ±6.8	**52.0** ±3.3	**28.0** ±7.5	**43.17** ±1.4
LingBot-Vision	10.7 ±2.5	9.3 ±5.7	38.0 ±1.6	32.0 ±3.3	30.0 ±0.0	14.0 ±3.3	31.78 ±0.9
SPA	5.3 ±3.4	**16.0** ±3.3	18.0 ±4.3	28.7 ±0.9	26.0 ±1.6	12.7 ±4.7	21.94 ±1.4
DINOv2	14.0 ±0.0	**21.3** ±0.9	**41.3** ±3.4	**51.3** ±2.5	**43.3** ±0.9	**20.0** ±4.3	**37.33** ±0.9
DINOv3	**14.7** ±5.2	**16.7** ±3.4	32.0 ±7.1	**50.7** ±2.5	35.3 ±1.9	14.0 ±2.8	32.50 ±1.4
MAE	13.3 ±6.6	3.3 ±2.5	11.3 ±6.6	33.3 ±0.9	10.7 ±1.9	**21.3** ±0.9	22.89 ±1.1
V-JEPA 2.1	9.3 ±3.4	8.0 ±5.9	28.7 ±5.2	28.0 ±1.6	19.3 ±3.8	12.7 ±1.9	23.39 ±1.0
Xiaomi-Robotics-1	14.0 ±2.8	9.3 ±2.5	28.7 ±8.4	28.0 ±5.9	21.3 ±5.0	12.0 ±5.7	24.83 ±1.6
DepthVLM	10.7 ±3.4	8.7 ±6.8	36.0 ±2.8	39.3 ±9.0	23.3 ±4.1	14.7 ±1.9	28.22 ±1.6
Qwen3-VL	**16.7** ±5.2	12.0 ±5.9	31.3 ±8.4	24.7 ±7.7	22.0 ±1.6	17.3 ±0.9	23.78 ±1.7
SigLIP2	8.0 ±4.9	2.7 ±0.9	**43.3** ±0.9	38.7 ±9.3	**36.7** ±6.6	12.0 ±4.3	29.78 ±1.6
InternViT	8.7 ±2.5	8.0 ±5.9	19.3 ±2.5	22.7 ±0.9	18.7 ±4.7	9.3 ±3.4	22.39 ±1.0
SigLIP	10.0 ±1.6	3.3 ±2.5	10.7 ±5.2	18.0 ±5.9	14.7 ±0.9	5.3 ±4.1	16.61 ±1.1
CLIP	10.0 ±5.9	8.7 ±4.1	10.7 ±5.0	25.3 ±4.1	2.0 ±1.6	4.0 ±3.3	15.17 ±1.5
VC-1	**24.0** ±4.3	12.0 ±1.6	18.7 ±5.2	42.0 ±7.5	33.3 ±3.4	12.7 ±2.5	**32.61** ±1.2
Voltron	12.0 ±4.3	14.7 ±3.4	36.0 ±3.3	34.0 ±3.3	26.7 ±3.4	19.3 ±0.9	31.94 ±1.2
Theia	6.7 ±2.5	10.0 ±1.6	34.0 ±4.3	30.7 ±6.2	28.0 ±5.7	16.7 ±6.6	27.00 ±1.3
C-RADIOv4	2.7 ±0.9	1.3 ±0.9	2.0 ±1.6	2.0 ±1.6	2.7 ±0.9	0.7 ±0.9	6.06 ±0.9
RADIOv2.5	1.3 ±0.9	3.3 ±1.9	4.7 ±2.5	4.7 ±2.5	1.3 ±0.9	0.7 ±0.9	5.44 ±0.8
ViT	5.3 ±0.9	5.3 ±2.5	10.0 ±2.8	14.0 ±2.8	16.0 ±4.3	6.0 ±4.3	14.67 ±1.1
VLM layer 16							
Xiaomi-Robotics-1 L16	15.3 ±3.4	12.7 ±3.4	34.7 ±1.9	36.7 ±5.2	36.0 ±1.6	20.7 ±6.8	33.11 ±1.5
DepthVLM L16	18.7 ±0.9	12.7 ±3.8	42.0 ±7.5	34.7 ±3.4	30.7 ±3.8	22.0 ±1.6	34.17 ±1.5
Qwen3-VL L16	14.0 ±4.3	12.7 ±1.9	38.7 ±10.5	43.3 ±2.5	30.0 ±6.5	19.3 ±5.0	32.67 ±1.7
Fine-tuned encoders							
VGGT-Ω FT	26.7 ±4.1	14.0 ±1.6	70.7 ±3.4	48.7 ±3.4	56.7 ±1.9	28.0 ±7.5	50.94 ±1.3
DINOv2 FT	8.0 ±1.6	17.3 ±5.0	37.3 ±6.2	35.3 ±12.5	39.3 ±4.7	33.3 ±7.7	39.72 ±1.7
DINOv3 FT	17.3 ±2.5	14.0 ±5.7	53.3 ±5.0	42.7 ±10.9	43.3 ±4.7	20.7 ±0.9	41.94 ±1.5
SigLIP2 FT	12.0 ±3.3	14.7 ±1.9	35.3 ±7.7	37.3 ±6.6	34.0 ±4.3	8.0 ±2.8	30.89 ±1.3
VC-1 FT	18.0 ±4.3	9.3 ±0.9	42.0 ±7.5	30.0 ±1.6	40.0 ±4.3	31.3 ±4.1	38.78 ±1.3
SPA FT	11.3 ±0.9	9.3 ±2.5	8.7 ±3.4	22.0 ±4.3	25.3 ±8.1	10.0 ±1.6	24.56 ±1.4
C-RADIOv4 FT	2.7 ±2.5	2.0 ±0.0	2.7 ±0.9	0.7 ±0.9	4.7 ±2.5	0.7 ±0.9	8.39 ±0.9

Table 21: RoboCasa-GR1 multi-task success per task. Same format as Table 19. The global average pooling (GAP) rows condition the same action head on one mean-pooled token per view instead of the patch tokens. This part lists tasks C1–C6; Table 22 continues with the remaining tasks and the suite average.

Encoder	C1	C2	C3	C4	C5	C6
Frozen encoders						
VGGT-Ω	29.3 ±5.7	38.0 ±1.6	34.0 ±4.3	**39.3 ±2.5**	**52.0 ±1.6**	**32.0 ±4.3**
LingBot-Vision	34.0 ±8.2	**49.3 ±9.0**	34.7 ±5.7	27.3 ±3.4	44.0 ±2.8	12.7 ±2.5
SPA	10.0 ±1.6	17.3 ±0.9	12.7 ±5.2	3.3 ±1.9	16.7 ±3.8	10.7 ±4.1
DINOv2	**41.3 ±5.2**	37.3 ±2.5	**38.0 ±4.3**	**36.7 ±4.1**	**49.3 ±2.5**	20.7 ±3.4
DINOv3	**50.0 ±4.3**	34.0 ±1.6	**38.7 ±6.2**	20.7 ±5.2	41.3 ±5.2	**22.0 ±3.3**
MAE	36.7 ±7.4	39.3 ±5.0	26.7 ±0.9	24.0 ±1.6	40.7 ±6.6	16.0 ±4.3
V-JEPA 2.1	26.7 ±5.7	26.7 ±2.5	30.7 ±7.5	14.0 ±0.0	36.7 ±5.0	4.7 ±2.5
Xiaomi-Robotics-1	35.3 ±6.8	38.0 ±1.6	22.7 ±7.4	25.3 ±3.8	32.7 ±3.4	16.0 ±4.3
DepthVLM	27.3 ±2.5	22.7 ±2.5	6.7 ±2.5	22.0 ±2.8	22.7 ±3.8	16.7 ±3.4
Qwen3-VL	24.7 ±5.0	24.7 ±2.5	11.3 ±2.5	20.0 ±7.1	29.3 ±5.2	13.3 ±2.5
SigLIP2	35.3 ±5.7	32.0 ±4.3	**43.3 ±5.7**	**29.3 ±4.7**	46.0 ±10.2	20.0 ±3.3
InternViT	39.3 ±3.4	24.7 ±0.9	27.3 ±6.8	13.3 ±0.9	16.7 ±0.9	6.7 ±1.9
SigLIP	19.3 ±7.5	20.0 ±4.3	16.7 ±5.2	6.7 ±2.5	19.3 ±4.1	5.3 ±1.9
CLIP	16.7 ±0.9	12.0 ±1.6	14.0 ±5.9	8.0 ±4.9	12.7 ±2.5	2.7 ±0.9
VC-1	**40.7 ±5.0**	**47.3 ±6.6**	26.7 ±5.7	17.3 ±7.7	**48.7 ±3.4**	**24.7 ±6.6**
Voltron	27.3 ±4.1	**45.3 ±3.4**	20.7 ±1.9	18.0 ±1.6	42.7 ±5.0	**22.0 ±1.6**
Theia	30.0 ±3.3	36.7 ±1.9	35.3 ±2.5	20.7 ±0.9	34.0 ±1.6	**22.0 ±4.3**
C-RADIOv4	10.7 ±2.5	8.0 ±2.8	18.7 ±1.9	4.7 ±0.9	6.7 ±0.9	0.7 ±0.9
RADIOv2.5	4.7 ±5.2	10.7 ±4.1	13.3 ±0.9	3.3 ±1.9	3.3 ±0.9	0.7 ±0.9
ViT	13.3 ±2.5	21.3 ±4.7	24.0 ±7.5	9.3 ±3.4	18.7 ±2.5	8.0 ±3.3
VLM layer 16						
Qwen3-VL L16	43.3 ±5.2	52.7 ±1.9	30.7 ±3.4	32.0 ±8.5	37.3 ±6.2	17.3 ±3.4
Global average pooling						
DINOv2 GAP	11.3 ±6.8	20.0 ±4.9	16.0 ±1.6	7.3 ±2.5	7.3 ±3.4	5.3 ±2.5
DINOv3 GAP	16.0 ±5.9	12.7 ±0.9	4.0 ±1.6	2.0 ±1.6	2.7 ±0.9	1.3 ±0.9

Table 22: Continued from Table 21. Tasks C7–C12 and the suite average.

Encoder	C7	C8	C9	C10	C11	C12	Avg.
Frozen encoders							
VGGT-Ω	**21.3** ±2.5	**16.0** ±4.9	**44.0** ±8.6	**46.0** ±3.3	**45.3** ±5.7	**27.3** ±5.0	**35.39** ±1.3
LingBot-Vision	10.7 ±1.9	**14.7** ±3.4	**34.7** ±6.2	34.7 ±5.2	35.3 ±2.5	**20.7** ±5.2	29.39 ±1.5
SPA	12.0 ±0.0	5.3 ±3.4	9.3 ±2.5	18.7 ±1.9	15.3 ±0.9	0.0 ±0.0	10.94 ±0.8
DINOv2	**28.7** ±2.5	14.0 ±3.3	**34.0** ±2.8	**46.0** ±13.4	**38.7** ±3.4	2.7 ±0.9	**32.28** ±1.5
DINOv3	**23.3** ±7.4	**25.3** ±0.9	26.0 ±4.3	**46.7** ±5.2	34.7 ±2.5	12.0 ±4.9	**31.22** ±1.3
MAE	10.7 ±6.8	8.0 ±4.3	30.0 ±5.7	26.7 ±5.2	29.3 ±2.5	15.3 ±6.6	25.28 ±1.5
V-JEPA 2.1	13.3 ±1.9	5.3 ±3.4	16.0 ±1.6	12.7 ±8.2	22.0 ±1.6	15.3 ±0.9	18.67 ±1.2
Xiaomi-Robotics-1	9.3 ±8.2	10.0 ±0.0	32.0 ±4.3	34.7 ±4.1	28.7 ±7.5	17.3 ±10.4	25.17 ±1.7
DepthVLM	10.0 ±3.3	10.7 ±2.5	22.0 ±2.8	26.0 ±0.0	21.3 ±9.0	14.0 ±3.3	18.50 ±1.1
Qwen3-VL	11.3 ±1.9	13.3 ±0.9	22.0 ±5.9	26.0 ±2.8	26.0 ±4.3	12.7 ±4.1	19.56 ±1.2
SigLIP2	18.7 ±5.2	12.0 ±1.6	26.7 ±2.5	33.3 ±6.8	**47.3** ±5.7	5.3 ±0.9	29.11 ±1.5
InternViT	9.3 ±2.5	11.3 ±5.0	21.3 ±2.5	28.0 ±2.8	26.0 ±1.6	17.3 ±5.0	20.11 ±1.0
SigLIP	12.0 ±1.6	10.7 ±1.9	14.7 ±3.8	15.3 ±7.7	20.0 ±4.9	4.0 ±1.6	13.67 ±1.3
CLIP	8.0 ±1.6	10.0 ±3.3	12.0 ±2.8	16.0 ±4.3	18.0 ±4.3	7.3 ±3.4	11.44 ±1.0
VC-1	16.0 ±4.9	14.0 ±5.9	31.3 ±0.9	**36.0** ±3.3	36.7 ±0.9	2.0 ±1.6	28.44 ±1.4
Voltron	8.0 ±1.6	7.3 ±0.9	26.7 ±6.6	28.7 ±3.4	33.3 ±0.9	18.0 ±4.9	24.83 ±1.0
Theia	6.7 ±2.5	9.3 ±2.5	24.0 ±8.2	29.3 ±3.4	31.3 ±5.7	**18.7** ±0.9	24.83 ±1.1
C-RADIOv4	3.3 ±2.5	0.7 ±0.9	7.3 ±3.4	3.3 ±3.4	9.3 ±1.9	2.0 ±1.6	6.28 ±0.6
RADIOv2.5	2.0 ±1.6	1.3 ±0.9	3.3 ±4.7	4.0 ±3.3	5.3 ±0.9	0.7 ±0.9	4.39 ±0.8
ViT	10.0 ±2.8	8.0 ±2.8	18.0 ±7.5	14.0 ±2.8	16.7 ±5.0	2.7 ±0.9	13.67 ±1.2
VLM layer 16							
Qwen3-VL L16	16.7 ±3.4	14.7 ±4.1	29.3 ±5.0	38.0 ±5.9	26.0 ±1.6	24.7 ±2.5	30.22 ±1.3
Global average pooling							
DINOv2 GAP	10.0 ±4.3	1.3 ±0.9	6.7 ±2.5	13.3 ±1.9	12.7 ±2.5	8.0 ±2.8	9.94 ±1.0
DINOv3 GAP	3.3 ±3.4	2.7 ±0.9	8.0 ±1.6	6.7 ±2.5	7.3 ±2.5	10.7 ±0.9	6.44 ±0.7

Table 23: RoboTwin 2.0 multi-task success per task. Same format as Table 19. VGGT-Ω receives the three views as one joint input. This part lists tasks W1–W6; Table 24 continues with the remaining tasks and the suite average.

Encoder	W1	W2	W3	W4	W5	W6
Frozen encoders						
VGGT-Ω	90.0 ±4.3	**74.0 ±7.5**	83.3 ±6.8	18.0 ±1.6	44.7 ±3.4	74.7 ±5.7
LingBot-Vision	91.3 ±4.1	62.0 ±5.9	84.7 ±3.4	**26.0 ±3.3**	**54.7 ±6.8**	84.0 ±8.2
SPA	39.3 ±6.8	11.3 ±2.5	20.0 ±1.6	0.0 ±0.0	3.3 ±0.9	36.7 ±5.0
DINOv2	**96.7 ±0.9**	**63.3 ±0.9**	**94.7 ±0.9**	14.7 ±6.2	44.7 ±8.4	**90.7 ±3.4**
DINOv3	72.0 ±1.6	42.0 ±5.9	63.3 ±6.8	15.3 ±5.0	16.0 ±7.1	59.3 ±2.5
MAE	77.3 ±2.5	32.0 ±3.3	60.7 ±10.4	0.0 ±0.0	12.0 ±4.3	69.3 ±1.9
V-JEPA 2.1	**94.7 ±2.5**	50.7 ±1.9	**92.0 ±1.6**	2.7 ±0.9	24.7 ±5.2	80.0 ±7.1
Xiaomi-Robotics-1	92.7 ±2.5	54.7 ±5.2	88.0 ±1.6	9.3 ±1.9	**66.0 ±5.7**	**95.3 ±3.4**
DepthVLM	92.7 ±4.7	60.7 ±0.9	**97.3 ±0.9**	**32.0 ±5.9**	**46.7 ±5.2**	82.0 ±4.3
Qwen3-VL	**94.7 ±3.4**	**68.0 ±4.3**	90.7 ±1.9	**32.7 ±8.1**	38.7 ±6.6	83.3 ±1.9
SigLIP2	**95.3 ±1.9**	56.7 ±6.6	**94.7 ±0.9**	1.3 ±0.9	46.0 ±8.6	79.3 ±8.2
InternViT	64.7 ±7.7	22.0 ±4.3	76.7 ±6.8	2.0 ±1.6	32.7 ±3.4	59.3 ±6.2
SigLIP	78.7 ±5.7	27.3 ±6.2	56.0 ±5.7	4.7 ±3.4	10.0 ±4.3	50.7 ±3.4
CLIP	62.0 ±2.8	25.3 ±6.6	40.0 ±1.6	4.0 ±2.8	16.0 ±1.6	46.7 ±0.9
VC-1	44.0 ±5.9	28.0 ±4.3	53.3 ±2.5	0.0 ±0.0	15.3 ±4.7	66.7 ±6.6
Voltron	70.0 ±2.8	34.0 ±1.6	49.3 ±4.1	0.0 ±0.0	13.3 ±2.5	42.0 ±4.3
Theia	93.3 ±0.9	50.7 ±0.9	89.3 ±0.9	3.3 ±1.9	40.0 ±6.5	**86.7 ±3.4**
C-RADIOv4	50.7 ±3.4	9.3 ±5.0	28.7 ±9.3	0.0 ±0.0	8.0 ±4.3	40.7 ±6.8
RADIOv2.5	40.7 ±8.4	14.0 ±2.8	13.3 ±2.5	0.0 ±0.0	6.7 ±2.5	36.7 ±5.2
ViT	46.0 ±5.9	17.3 ±0.9	26.7 ±5.0	2.0 ±1.6	11.3 ±0.9	53.3 ±5.0
VLM layer 16						
Xiaomi-Robotics-1 L16	98.0 ±1.6	60.0 ±6.5	85.3 ±4.1	20.7 ±6.6	66.0 ±5.9	92.7 ±2.5
Qwen3-VL L16	88.7 ±2.5	60.7 ±2.5	96.0 ±0.0	13.3 ±4.7	49.3 ±3.8	79.3 ±0.9
Fine-tuned encoders						
VGGT-Ω FT	99.3 ±0.9	79.3 ±5.7	99.3 ±0.9	76.0 ±5.9	95.3 ±0.9	97.3 ±2.5
DINOv2 FT	96.0 ±2.8	22.0 ±9.1	64.0 ±1.6	3.3 ±2.5	68.7 ±5.2	96.0 ±1.6
DINOv3 FT	58.0 ±6.5	49.3 ±0.9	93.3 ±4.7	26.0 ±5.9	59.3 ±1.9	82.0 ±4.3
SigLIP2 FT	95.3 ±0.9	24.7 ±3.4	58.7 ±7.5	4.0 ±2.8	68.7 ±0.9	91.3 ±2.5
VC-1 FT	84.7 ±3.4	35.3 ±3.4	94.7 ±2.5	0.7 ±0.9	39.3 ±4.1	92.7 ±3.4
SPA FT	90.0 ±1.6	35.3 ±10.5	90.0 ±2.8	0.0 ±0.0	23.3 ±5.2	91.3 ±2.5
C-RADIOv4 FT	61.3 ±8.1	8.7 ±1.9	19.3 ±1.9	0.0 ±0.0	18.0 ±3.3	78.7 ±3.4

Table 24: Continued from Table 23. Tasks W7–W12 and the suite average.

Encoder	W7	W8	W9	W10	W11	W12	Avg.
Frozen encoders							
VGGT-Ω	**48.0** ±1.6	**47.3** ±3.8	14.0 ±4.3	**57.3** ±0.9	48.0 ±7.1	0.7 ±0.9	50.00 ±1.3
LingBot-Vision	**41.3** ±3.4	40.7 ±6.6	10.0 ±2.8	36.0 ±1.6	30.7 ±4.1	0.0 ±0.0	46.78 ±1.4
SPA	2.7 ±1.9	5.3 ±0.9	0.7 ±0.9	14.7 ±2.5	5.3 ±2.5	0.7 ±0.9	11.67 ±0.8
DINOv2	29.3 ±3.4	**51.3** ±1.9	**18.7** ±0.9	47.3 ±1.9	39.3 ±2.5	2.0 ±2.8	49.39 ±1.0
DINOv3	9.3 ±0.9	25.3 ±6.6	3.3 ±0.9	31.3 ±7.7	26.7 ±5.2	1.3 ±0.9	30.44 ±1.4
MAE	5.3 ±1.9	22.7 ±6.6	1.3 ±0.9	33.3 ±12.3	4.7 ±1.9	1.3 ±0.9	26.67 ±1.6
V-JEPA 2.1	8.0 ±0.0	30.0 ±1.6	1.3 ±0.9	**53.3** ±9.6	38.0 ±7.1	0.0 ±0.0	39.61 ±1.3
Xiaomi-Robotics-1	22.0 ±4.9	**58.0** ±2.8	**24.0** ±4.3	**54.0** ±9.1	**74.7** ±5.0	0.0 ±0.0	**53.22** ±1.3
DepthVLM	**31.3** ±5.0	43.3 ±0.9	**23.3** ±9.0	48.0 ±2.8	**84.0** ±1.6	**2.7** ±0.9	**53.67** ±1.2
Qwen3-VL	**31.3** ±3.4	42.0 ±1.6	7.3 ±3.4	49.3 ±4.1	**77.3** ±2.5	2.0 ±1.6	**51.44** ±1.2
SigLIP2	20.0 ±1.6	30.7 ±2.5	9.3 ±2.5	49.3 ±2.5	44.0 ±2.8	**4.0** ±0.0	44.22 ±1.2
InternViT	18.7 ±0.9	27.3 ±5.2	4.7 ±2.5	34.7 ±3.8	26.0 ±2.8	**2.7** ±0.9	30.94 ±1.3
SigLIP	12.7 ±4.1	28.0 ±4.9	0.0 ±0.0	42.0 ±5.9	24.0 ±7.1	**6.7** ±0.9	28.39 ±1.4
CLIP	6.7 ±0.9	16.7 ±0.9	1.3 ±0.9	17.3 ±5.0	20.0 ±2.8	0.0 ±0.0	21.33 ±0.8
VC-1	12.0 ±1.6	28.7 ±7.4	2.7 ±0.9	39.3 ±10.6	10.0 ±2.8	0.7 ±0.9	25.06 ±1.5
Voltron	4.7 ±3.4	17.3 ±4.1	3.3 ±2.5	30.0 ±7.5	8.7 ±3.4	0.0 ±0.0	22.72 ±1.0
Theia	20.0 ±5.9	44.7 ±2.5	10.7 ±0.9	52.7 ±0.9	32.7 ±1.9	2.0 ±1.6	43.83 ±0.9
C-RADIOv4	7.3 ±0.9	5.3 ±1.9	0.7 ±0.9	20.0 ±1.6	12.0 ±1.6	**2.7** ±0.9	15.44 ±1.2
RADIOv2.5	4.0 ±1.6	6.7 ±4.1	0.0 ±0.0	19.3 ±5.7	8.7 ±0.9	1.3 ±1.9	12.61 ±1.1
ViT	10.7 ±5.0	13.3 ±1.9	0.0 ±0.0	30.0 ±4.3	12.7 ±2.5	0.0 ±0.0	18.61 ±1.0
VLM layer 16							
Xiaomi-Robotics-1 L16	16.7 ±0.9	50.7 ±4.7	18.7 ±5.7	58.7 ±5.0	66.0 ±5.9	0.0 ±0.0	52.78 ±1.3
Qwen3-VL L16	28.7 ±6.8	47.3 ±6.6	20.7 ±3.4	45.3 ±10.0	77.3 ±4.7	2.0 ±1.6	50.72 ±1.4
Fine-tuned encoders							
VGGT-Ω FT	58.7 ±6.2	82.0 ±5.9	77.3 ±2.5	78.7 ±2.5	69.3 ±5.7	0.0 ±0.0	76.06 ±1.2
DINOv2 FT	46.0 ±5.7	70.0 ±4.3	40.7 ±2.5	76.0 ±7.1	36.0 ±1.6	0.7 ±0.9	51.61 ±1.3
DINOv3 FT	38.0 ±4.9	39.3 ±9.6	29.3 ±0.9	72.0 ±3.3	36.0 ±1.6	0.7 ±0.9	48.61 ±1.3
SigLIP2 FT	32.0 ±5.9	52.7 ±10.5	25.3 ±5.7	67.3 ±3.4	64.7 ±3.8	0.0 ±0.0	48.72 ±1.4
VC-1 FT	13.3 ±4.7	62.7 ±5.0	7.3 ±0.9	61.3 ±10.0	36.0 ±2.8	0.0 ±0.0	44.00 ±1.2
SPA FT	7.3 ±0.9	38.0 ±4.3	7.3 ±0.9	38.7 ±5.7	21.3 ±2.5	1.3 ±0.9	37.00 ±1.2
C-RADIOv4 FT	0.0 ±0.0	38.7 ±2.5	3.3 ±0.9	50.7 ±6.8	4.0 ±2.8	0.0 ±0.0	23.56 ±1.0

K.2 Demonstration and model scale

Tables 25 to 33 give the per-task values behind Figures 12 and 13.

Table 25: Demonstration scale, RoboCasa-GR1 single-task. Four matched tasks (C1–C4) at 50–1,000 demonstrations per task; the 1,000 row reuses the corresponding tasks of Table 19.

Encoder	Setting	C1	C2	C3	C4	Avg.
DINOv2	50	15.3 ±5.2	24.0 ±1.6	21.3 ±6.6	3.3 ±1.9	16.00 ±2.2
	100	36.7 ±6.6	40.7 ±6.8	24.7 ±0.9	6.0 ±2.8	27.00 ±2.5
	200	50.7 ±11.5	55.3 ±7.4	40.0 ±10.2	14.0 ±1.6	40.00 ±4.3
	500	57.3 ±5.2	60.0 ±5.7	44.0 ±1.6	21.3 ±2.5	45.67 ±2.1
	1,000	60.7 ±5.0	61.3 ±3.4	49.3 ±6.2	26.7 ±0.9	49.50 ±2.2
DINOv3	50	13.3 ±5.0	20.0 ±1.6	12.0 ±1.6	2.0 ±1.6	11.83 ±1.4
	100	38.0 ±3.3	32.0 ±2.8	29.3 ±6.8	2.0 ±1.6	25.33 ±2.1
	200	43.3 ±5.0	43.3 ±2.5	31.3 ±5.2	12.0 ±3.3	32.50 ±2.1
	500	49.3 ±3.8	56.0 ±10.2	48.0 ±4.3	25.3 ±5.2	44.67 ±3.2
	1,000	53.3 ±2.5	60.0 ±5.7	47.3 ±6.6	17.3 ±5.7	44.50 ±2.7
SigLIP2	50	12.7 ±4.7	26.7 ±5.0	8.0 ±3.3	1.3 ±0.9	12.17 ±1.9
	100	31.3 ±2.5	36.7 ±0.9	23.3 ±11.5	2.7 ±2.5	23.50 ±3.0
	200	44.7 ±2.5	44.7 ±3.4	36.0 ±2.8	1.3 ±0.9	31.67 ±1.3
	500	49.3 ±4.7	57.7 ±2.0	39.3 ±5.7	11.3 ±2.5	39.42 ±2.0
	1,000	48.0 ±1.6	61.3 ±6.8	44.0 ±6.5	15.3 ±3.4	42.16 ±2.5
V-JEPA 2.1	50	10.7 ±7.4	14.7 ±5.0	14.0 ±2.8	0.7 ±0.9	10.00 ±2.3
	100	24.0 ±6.5	21.3 ±7.5	24.7 ±7.7	1.3 ±0.9	17.83 ±3.2
	200	36.7 ±4.1	34.7 ±3.8	31.3 ±2.5	1.3 ±0.9	26.00 ±1.5
	500	42.0 ±7.1	52.0 ±3.3	33.3 ±1.9	1.3 ±0.9	32.17 ±2.0
	1,000	44.0 ±4.3	54.7 ±0.9	34.7 ±4.1	1.3 ±0.9	33.67 ±1.5

Table 26: Demonstration scale, RoboCasa-GR1 multi-task. All twelve tasks; the 1,000 row reuses Table 21. This part lists tasks C1–C6; Table 27 continues with the remaining tasks and the suite average.

Encoder	Setting	C1	C2	C3	C4	C5	C6
DINOv2	50	8.0 ±4.3	4.0 ±1.6	5.3 ±3.4	6.0 ±2.8	7.3 ±0.9	3.3 ±2.5
	100	7.3 ±3.4	4.7 ±1.9	4.0 ±1.6	12.0 ±4.3	7.3 ±2.5	8.0 ±3.3
	200	14.0 ±3.3	15.3 ±5.2	16.7 ±0.9	16.0 ±2.8	11.3 ±0.9	10.0 ±0.0
	500	24.7 ±4.1	22.7 ±2.5	28.7 ±6.6	25.3 ±5.2	26.7 ±5.0	16.0 ±4.3
	1,000	41.3 ±5.2	37.3 ±2.5	38.0 ±4.3	36.7 ±4.1	49.3 ±2.5	20.7 ±3.4
DINOv3	50	2.0 ±1.6	2.0 ±2.8	9.3 ±2.5	4.0 ±3.3	2.7 ±0.9	0.7 ±0.9
	100	11.3 ±3.8	6.7 ±1.9	8.0 ±0.0	2.7 ±2.5	6.7 ±3.4	3.3 ±3.4
	200	8.7 ±3.4	8.7 ±1.9	18.0 ±4.3	12.7 ±3.4	24.0 ±7.5	7.3 ±0.9
	500	26.0 ±3.3	30.7 ±5.2	26.0 ±5.9	16.7 ±0.9	44.0 ±4.9	9.3 ±3.4
	1,000	50.0 ±4.3	34.0 ±1.6	38.7 ±6.2	20.7 ±5.2	41.3 ±5.2	22.0 ±3.3
SigLIP2	50	4.7 ±2.5	4.7 ±3.4	4.0 ±1.6	4.0 ±0.0	5.3 ±3.4	3.3 ±0.9
	100	10.0 ±4.9	8.7 ±2.5	7.3 ±3.4	8.7 ±3.8	7.3 ±1.9	6.0 ±0.0
	200	17.3 ±0.9	14.7 ±1.9	13.0 ±6.5	14.7 ±4.1	16.0 ±4.3	7.3 ±4.1
	500	24.0 ±5.9	26.0 ±4.3	26.0 ±1.6	22.7 ±9.6	32.7 ±5.2	14.0 ±5.9
	1,000	35.3 ±5.7	32.0 ±4.3	43.3 ±5.7	29.3 ±4.7	46.0 ±10.2	20.0 ±3.3
SPA	50	1.3 ±1.9	0.0 ±0.0	4.7 ±0.9	0.0 ±0.0	2.0 ±2.8	0.7 ±0.9
	100	1.3 ±0.9	0.0 ±0.0	4.0 ±2.8	1.3 ±1.9	2.0 ±0.0	0.7 ±0.9
	200	2.0 ±0.0	2.7 ±0.9	6.7 ±3.4	2.0 ±1.6	0.7 ±0.9	1.3 ±0.9
	500	6.7 ±3.4	14.0 ±2.8	14.7 ±2.5	4.0 ±2.8	10.7 ±0.9	3.3 ±3.4
	1,000	10.0 ±1.6	17.3 ±0.9	12.7 ±5.2	3.3 ±1.9	16.7 ±3.8	10.7 ±4.1
VC-1	50	4.0 ±0.0	4.0 ±1.6	5.3 ±0.9	2.0 ±1.6	8.0 ±1.6	1.3 ±1.9
	100	8.7 ±0.9	4.7 ±0.9	9.3 ±2.5	4.0 ±3.3	13.3 ±4.1	2.0 ±0.0
	200	12.7 ±6.6	21.3 ±5.0	15.3 ±5.0	8.0 ±1.6	18.0 ±3.3	5.3 ±3.4
	500	29.3 ±5.7	32.0 ±8.6	29.3 ±3.8	18.7 ±2.5	32.0 ±9.1	8.7 ±2.5
	1,000	40.7 ±5.0	47.3 ±6.6	26.7 ±5.7	17.3 ±7.7	48.7 ±3.4	24.7 ±6.6
C-RADIOv4	50	0.0 ±0.0	0.0 ±0.0	2.0 ±2.8	0.7 ±0.9	0.0 ±0.0	0.7 ±0.9
	100	0.0 ±0.0	1.3 ±0.9	5.3 ±1.9	0.7 ±0.9	2.0 ±1.6	2.0 ±1.6
	200	0.0 ±0.0	4.0 ±2.8	6.7 ±2.5	1.3 ±1.9	0.7 ±0.9	0.7 ±0.9
	500	0.7 ±0.9	4.0 ±1.6	9.3 ±1.9	2.0 ±1.6	2.7 ±2.5	1.3 ±0.9
	1,000	10.7 ±2.5	8.0 ±2.8	18.7 ±1.9	4.7 ±0.9	6.7 ±0.9	0.7 ±0.9

Table 27: Continued from Table 26. Tasks C7–C12 and the suite average.

Encoder	Setting	C7	C8	C9	C10	C11	C12	Avg.
DINOv2	50	4.7 ±3.4	2.0 ±0.0	5.3 ±0.9	11.3 ±2.5	7.3 ±1.9	1.3 ±1.9	5.50 ±0.7
	100	8.7 ±4.1	4.7 ±2.5	13.3 ±4.1	11.3 ±1.9	15.3 ±1.9	6.0 ±2.8	8.56 ±0.9
	200	14.0 ±4.3	11.3 ±0.9	18.0 ±2.8	24.0 ±5.7	14.0 ±6.5	10.0 ±4.3	14.56 ±1.1
	500	18.0 ±6.5	6.7 ±0.9	24.0 ±3.3	26.0 ±3.3	22.7 ±9.0	14.7 ±0.9	21.33 ±1.4
	1,000	28.7 ±2.5	14.0 ±3.3	34.0 ±2.8	46.0 ±13.4	38.7 ±3.4	2.7 ±0.9	32.28 ±1.5
DINOv3	50	2.0 ±1.6	1.3 ±1.9	2.7 ±0.9	6.0 ±5.7	3.3 ±0.9	0.7 ±0.9	3.06 ±0.7
	100	6.7 ±1.9	4.0 ±0.0	9.3 ±0.9	12.0 ±3.3	8.0 ±1.6	4.0 ±1.6	6.89 ±0.7
	200	9.3 ±5.0	8.7 ±1.9	14.0 ±7.1	19.3 ±6.2	14.0 ±7.1	6.0 ±4.3	12.56 ±1.4
	500	16.7 ±5.7	7.3 ±1.9	18.7 ±1.9	31.3 ±3.8	26.7 ±6.6	14.0 ±1.6	22.28 ±1.2
	1,000	23.3 ±7.4	25.3 ±0.9	26.0 ±4.3	46.7 ±5.2	34.7 ±2.5	12.0 ±4.9	31.22 ±1.3
SigLIP2	50	0.7 ±0.9	0.0 ±0.0	6.7 ±0.9	9.3 ±4.1	7.3 ±4.7	0.0 ±0.0	4.17 ±0.7
	100	13.3 ±3.4	3.3 ±1.9	7.3 ±1.9	10.7 ±4.1	8.7 ±0.9	5.3 ±0.9	8.06 ±0.8
	200	14.0 ±3.3	7.3 ±3.4	16.0 ±4.9	16.0 ±1.6	18.7 ±1.9	6.7 ±8.1	13.47 ±1.2
	500	13.3 ±0.9	14.7 ±7.5	28.7 ±0.9	23.3 ±0.9	26.7 ±4.1	11.3 ±3.8	21.94 ±1.4
	1,000	18.7 ±5.2	12.0 ±1.6	26.7 ±2.5	33.3 ±6.8	47.3 ±5.7	5.3 ±0.9	29.11 ±1.5
SPA	50	1.3 ±0.9	1.3 ±1.9	0.7 ±0.9	0.0 ±0.0	2.0 ±1.6	1.3 ±0.9	1.28 ±0.4
	100	4.0 ±3.3	0.7 ±0.9	2.0 ±1.6	2.7 ±0.9	0.7 ±0.9	0.0 ±0.0	1.61 ±0.5
	200	1.3 ±0.9	1.3 ±0.9	0.0 ±0.0	3.3 ±0.9	0.7 ±0.9	0.0 ±0.0	1.83 ±0.4
	500	5.3 ±1.9	3.3 ±0.9	1.3 ±1.9	6.0 ±1.6	6.0 ±3.3	0.7 ±0.9	6.33 ±0.7
	1,000	12.0 ±0.0	5.3 ±3.4	9.3 ±2.5	18.7 ±1.9	15.3 ±0.9	0.0 ±0.0	10.94 ±0.8
VC-1	50	5.3 ±0.9	3.3 ±2.5	4.0 ±1.6	4.0 ±3.3	5.3 ±3.4	1.3 ±0.9	4.00 ±0.6
	100	5.3 ±0.9	3.3 ±0.9	6.7 ±2.5	12.7 ±3.8	9.3 ±6.2	2.7 ±2.5	6.83 ±0.8
	200	8.7 ±6.6	2.0 ±1.6	12.0 ±4.3	18.7 ±3.4	20.0 ±1.6	8.7 ±3.8	12.56 ±1.2
	500	12.0 ±5.7	4.7 ±0.9	15.3 ±3.4	22.0 ±4.3	29.3 ±5.2	14.0 ±0.0	20.61 ±1.5
	1,000	16.0 ±4.9	14.0 ±5.9	31.3 ±0.9	36.0 ±3.3	36.7 ±0.9	2.0 ±1.6	28.44 ±1.4
C-RADIOv4	50	0.7 ±0.9	0.0 ±0.0	0.7 ±0.9	1.3 ±1.9	1.3 ±1.9	0.0 ±0.0	0.61 ±0.4
	100	0.7 ±0.9	0.7 ±0.9	2.7 ±0.9	0.0 ±0.0	1.3 ±0.9	0.0 ±0.0	1.39 ±0.3
	200	0.7 ±0.9	2.0 ±1.6	0.7 ±0.9	0.7 ±0.9	2.0 ±1.6	0.0 ±0.0	1.61 ±0.4
	500	0.7 ±0.9	0.7 ±0.9	0.0 ±0.0	2.0 ±1.6	2.0 ±1.6	1.3 ±0.9	2.22 ±0.4
	1,000	3.3 ±2.5	0.7 ±0.9	7.3 ±3.4	3.3 ±3.4	9.3 ±1.9	2.0 ±1.6	6.28 ±0.6

Table 28: Demonstration scale, RoboTwin 2.0 multi-task. All twelve tasks; the 500 row is the full budget of Table 23. This part lists tasks W1–W6; Table 29 continues with the remaining tasks and the suite average.

Encoder	Setting	W1	W2	W3	W4	W5	W6
DINOv2	50	60.7 ±5.2	10.0 ±1.6	56.7 ±4.1	0.0 ±0.0	4.0 ±2.8	48.7 ±3.4
	100	56.7 ±5.2	22.0 ±2.8	72.0 ±2.8	0.7 ±0.9	14.7 ±3.8	70.0 ±1.6
	200	86.0 ±4.9	35.3 ±4.1	73.3 ±7.4	0.0 ±0.0	16.0 ±4.3	72.0 ±1.6
	500	96.7 ±0.9	63.3 ±0.9	94.7 ±0.9	14.7 ±6.2	44.7 ±8.4	90.7 ±3.4
DINOv3	50	52.0 ±1.6	13.3 ±2.5	34.7 ±1.9	0.7 ±0.9	3.3 ±2.5	44.7 ±4.1
	100	58.7 ±3.4	26.7 ±0.9	50.7 ±5.0	6.0 ±0.0	6.7 ±2.5	54.7 ±4.1
	200	60.7 ±2.5	37.3 ±4.1	53.3 ±1.9	10.0 ±1.6	8.7 ±0.9	54.7 ±7.4
	500	72.0 ±1.6	42.0 ±5.9	63.3 ±6.8	15.3 ±5.0	16.0 ±7.1	59.3 ±2.5
SigLIP2	50	29.3 ±8.2	2.0 ±1.6	22.0 ±3.3	0.0 ±0.0	3.3 ±1.9	24.7 ±10.6
	100	46.0 ±4.9	7.3 ±3.4	41.3 ±2.5	0.0 ±0.0	15.3 ±3.8	27.3 ±4.7
	200	81.3 ±5.0	27.3 ±5.0	65.3 ±3.8	2.0 ±1.6	35.3 ±5.0	50.7 ±2.5
	500	95.3 ±1.9	56.7 ±6.6	94.7 ±0.9	1.3 ±0.9	46.0 ±8.6	79.3 ±8.2
V-JEPA 2.1	50	34.0 ±3.3	5.3 ±0.9	28.0 ±8.6	0.0 ±0.0	4.0 ±4.3	33.3 ±4.1
	100	38.7 ±5.7	13.3 ±5.2	22.0 ±3.3	0.7 ±0.9	7.3 ±2.5	48.7 ±0.9
	200	70.7 ±5.0	24.7 ±5.7	54.0 ±5.9	1.3 ±0.9	10.0 ±1.6	67.3 ±3.8
	500	94.7 ±2.5	50.7 ±1.9	92.0 ±1.6	2.7 ±0.9	24.7 ±5.2	80.0 ±7.1

Table 29: Continued from Table 28. Tasks W7–W12 and the suite average.

Encoder	Setting	W7	W8	W9	W10	W11	W12	Avg.
DINOv2	50	0.7 ±0.9	4.0 ±0.0	0.0 ±0.0	22.7 ±9.0	10.7 ±3.4	0.0 ±0.0	18.17 ±1.1
	100	2.7 ±0.9	12.0 ±6.5	2.7 ±2.5	37.3 ±3.8	10.0 ±2.8	0.0 ±0.0	25.06 ±1.0
	200	4.0 ±3.3	22.7 ±4.1	6.7 ±1.9	48.0 ±3.3	18.0 ±3.3	0.7 ±0.9	31.89 ±1.1
	500	29.3 ±3.4	51.3 ±1.9	18.7 ±0.9	47.3 ±1.9	39.3 ±2.5	2.0 ±2.8	49.39 ±1.0
DINOv3	50	0.0 ±0.0	6.0 ±4.3	0.0 ±0.0	12.7 ±2.5	4.7 ±2.5	0.0 ±0.0	14.33 ±0.7
	100	2.0 ±1.6	9.3 ±3.4	0.0 ±0.0	10.7 ±1.9	9.3 ±0.9	0.0 ±0.0	19.56 ±0.7
	200	6.7 ±1.9	27.3 ±3.4	0.7 ±0.9	14.0 ±1.6	18.7 ±4.7	0.7 ±0.9	24.39 ±0.9
	500	9.3 ±0.9	25.3 ±6.6	3.3 ±0.9	31.3 ±7.7	26.7 ±5.2	1.3 ±0.9	30.44 ±1.4
SigLIP2	50	2.0 ±0.0	1.3 ±0.9	0.0 ±0.0	23.3 ±5.0	3.3 ±3.4	0.0 ±0.0	9.28 ±1.3
	100	2.0 ±1.6	8.0 ±1.6	3.3 ±0.9	31.3 ±2.5	18.0 ±0.0	0.7 ±0.9	16.72 ±0.8
	200	6.7 ±4.1	27.3 ±1.9	5.3 ±3.4	40.7 ±6.2	30.7 ±1.9	1.3 ±0.9	31.17 ±1.1
	500	20.0 ±1.6	30.7 ±2.5	9.3 ±2.5	49.3 ±2.5	44.0 ±2.8	4.0 ±0.0	44.22 ±1.2
V-JEPA 2.1	50	1.3 ±0.9	1.3 ±1.9	0.0 ±0.0	11.3 ±1.9	3.3 ±0.9	0.7 ±0.9	10.22 ±1.0
	100	8.7 ±0.9	6.7 ±3.4	0.0 ±0.0	16.0 ±6.5	2.0 ±1.6	0.0 ±0.0	13.67 ±1.0
	200	16.7 ±0.9	12.7 ±3.4	0.7 ±0.9	24.7 ±0.9	4.7 ±0.9	1.3 ±0.9	24.06 ±0.9
	500	8.0 ±0.0	30.0 ±1.6	1.3 ±0.9	53.3 ±9.6	38.0 ±7.1	0.0 ±0.0	39.61 ±1.3

Table 30: Model scale, RoboCasa-GR1 multi-task. Every row is a frozen checkpoint; the default row is the one used in Table 2. This part lists tasks C1–C6; Table 31 continues with the remaining tasks and the suite average.

Encoder	Setting	C1	C2	C3	C4	C5	C6
DINOv2	S	32.0 ±7.5	40.0 ±2.8	24.0 ±1.6	25.3 ±2.5	44.0 ±4.3	13.3 ±7.5
	B (default)	41.3 ±5.2	37.3 ±2.5	38.0 ±4.3	36.7 ±4.1	49.3 ±2.5	20.7 ±3.4
	L	31.3 ±4.1	44.7 ±3.4	22.7 ±4.1	28.0 ±7.5	50.7 ±11.5	29.3 ±6.6
	g	35.3 ±7.7	36.0 ±5.9	34.0 ±7.5	30.7 ±3.4	42.7 ±2.5	21.3 ±2.5
DINOv3	S	26.0 ±3.3	28.7 ±7.7	17.3 ±1.9	14.0 ±1.6	33.3 ±4.1	9.3 ±5.0
	B (default)	50.0 ±4.3	34.0 ±1.6	38.7 ±6.2	20.7 ±5.2	41.3 ±5.2	22.0 ±3.3
	L	47.3 ±6.6	28.7 ±6.8	35.3 ±5.0	21.3 ±3.8	32.7 ±3.8	15.3 ±4.1
	H+	44.7 ±4.1	23.3 ±1.9	24.0 ±1.6	24.0 ±3.3	32.0 ±1.6	18.7 ±3.4
V-JEPA 2.1	B (default)	26.7 ±5.7	26.7 ±2.5	30.7 ±7.5	14.0 ±0.0	36.7 ±5.0	4.7 ±2.5
	L	40.7 ±9.3	44.0 ±2.8	34.0 ±3.3	38.0 ±2.8	53.3 ±7.4	32.0 ±7.5
	g	44.0 ±13.0	46.0 ±4.9	37.3 ±2.5	36.7 ±3.8	56.0 ±2.8	21.3 ±1.9
VC-1	B (default)	40.7 ±5.0	47.3 ±6.6	26.7 ±5.7	17.3 ±7.7	48.7 ±3.4	24.7 ±6.6
	L	40.0 ±4.3	57.3 ±8.1	31.3 ±2.5	27.3 ±6.6	44.0 ±7.1	22.0 ±1.6
SPA	B (default)	10.0 ±1.6	17.3 ±0.9	12.7 ±5.2	3.3 ±1.9	16.7 ±3.8	10.7 ±4.1
	L	12.7 ±1.9	16.7 ±1.9	11.3 ±2.5	6.7 ±0.9	10.0 ±1.6	5.3 ±1.9
C-RADIOv4	SO400M	4.7 ±3.8	11.3 ±2.5	8.0 ±2.8	4.0 ±3.3	4.0 ±3.3	1.3 ±1.9
	H (default)	10.7 ±2.5	8.0 ±2.8	18.7 ±1.9	4.7 ±0.9	6.7 ±0.9	0.7 ±0.9

Table 31: Continued from Table 30. Tasks C7–C12 and the suite average.

Encoder	Setting	C7	C8	C9	C10	C11	C12	Avg.
DINOv2	S	14.7 ±3.4	16.7 ±9.8	30.0 ±4.3	27.3 ±3.4	42.0 ±4.9	19.3 ±2.5	27.39 ±1.5
	B (default)	28.7 ±2.5	14.0 ±3.3	34.0 ±2.8	46.0 ±13.4	38.7 ±3.4	2.7 ±0.9	32.28 ±1.5
	L	14.0 ±5.9	9.3 ±0.9	30.0 ±8.5	30.0 ±5.9	24.7 ±3.4	20.7 ±8.2	27.94 ±1.9
	g	19.3 ±4.1	12.0 ±0.0	33.3 ±9.8	26.7 ±5.2	24.0 ±0.0	25.3 ±1.9	28.39 ±1.5
DINOv3	S	12.7 ±4.7	6.7 ±2.5	18.0 ±3.3	27.3 ±6.8	21.3 ±2.5	11.3 ±2.5	18.83 ±1.2
	B (default)	23.3 ±7.4	25.3 ±0.9	26.0 ±4.3	46.7 ±5.2	34.7 ±2.5	12.0 ±4.9	31.22 ±1.3
	L	13.3 ±5.2	10.7 ±3.8	21.3 ±2.5	26.0 ±5.9	22.7 ±2.5	17.3 ±3.4	24.33 ±1.3
	H+	16.7 ±1.9	14.7 ±1.9	22.7 ±1.9	25.3 ±7.5	31.3 ±4.1	20.0 ±3.3	24.78 ±1.0
V-JEPA 2.1	B (default)	13.3 ±1.9	5.3 ±3.4	16.0 ±1.6	12.7 ±8.2	22.0 ±1.6	15.3 ±0.9	18.67 ±1.2
	L	17.3 ±5.7	20.0 ±8.2	39.3 ±9.0	43.3 ±1.9	40.0 ±8.6	22.0 ±4.3	35.33 ±1.9
	g	22.7 ±2.5	22.0 ±4.9	36.0 ±3.3	42.0 ±1.6	40.7 ±5.7	30.0 ±4.9	36.22 ±1.5
VC-1	B (default)	16.0 ±4.9	14.0 ±5.9	31.3 ±0.9	36.0 ±3.3	36.7 ±0.9	2.0 ±1.6	28.44 ±1.4
	L	16.7 ±7.7	8.7 ±3.4	30.0 ±4.3	41.3 ±3.4	39.3 ±3.4	20.7 ±7.7	31.56 ±1.6
SPA	B (default)	12.0 ±0.0	5.3 ±3.4	9.3 ±2.5	18.7 ±1.9	15.3 ±0.9	0.0 ±0.0	10.94 ±0.8
	L	9.3 ±1.9	4.7 ±0.9	9.3 ±2.5	10.0 ±2.8	16.7 ±6.8	4.7 ±1.9	9.78 ±0.8
C-RADIOv4	SO400M	1.3 ±1.9	0.7 ±0.9	4.0 ±0.0	11.3 ±0.9	6.0 ±2.8	1.3 ±0.9	4.83 ±0.7
	H (default)	3.3 ±2.5	0.7 ±0.9	7.3 ±3.4	3.3 ±3.4	9.3 ±1.9	2.0 ±1.6	6.28 ±0.6

Table 32: Model scale, RoboTwin 2.0 multi-task. Every row is a frozen checkpoint; the default row is the one used in Table 2. This part lists tasks W1–W6; Table 33 continues with the remaining tasks and the suite average.

Encoder	Setting	W1	W2	W3	W4	W5	W6
DINOv2	S	84.0 ±8.2	32.7 ±4.1	90.0 ±0.0	18.7 ±6.2	46.0 ±7.1	82.0 ±3.3
	B (default)	96.7 ±0.9	63.3 ±0.9	94.7 ±0.9	14.7 ±6.2	44.7 ±8.4	90.7 ±3.4
	L	92.0 ±0.0	71.3 ±3.4	93.3 ±1.9	28.7 ±0.9	33.3 ±2.5	96.0 ±2.8
	g	80.7 ±2.5	60.0 ±3.3	93.3 ±0.9	34.0 ±4.3	34.7 ±2.5	90.7 ±0.9
DINOv3	S	52.7 ±8.2	31.3 ±3.8	46.0 ±6.5	8.0 ±1.6	21.3 ±6.6	54.7 ±2.5
	B (default)	72.0 ±1.6	42.0 ±5.9	63.3 ±6.8	15.3 ±5.0	16.0 ±7.1	59.3 ±2.5
	L	73.3 ±6.2	29.3 ±5.0	58.7 ±2.5	15.3 ±5.2	21.3 ±0.9	63.3 ±5.0
	H+	75.3 ±1.9	40.0 ±7.5	72.0 ±7.5	20.7 ±5.0	22.7 ±5.2	50.7 ±5.0
V-JEPA 2.1	B (default)	94.7 ±2.5	50.7 ±1.9	92.0 ±1.6	2.7 ±0.9	24.7 ±5.2	80.0 ±7.1
	L	73.3 ±2.5	12.0 ±0.0	34.7 ±0.9	4.7 ±3.4	40.7 ±3.8	84.0 ±7.5
	g	97.3 ±1.9	41.3 ±3.8	96.7 ±0.9	5.3 ±0.9	67.3 ±5.0	96.7 ±2.5
VC-1	B (default)	44.0 ±5.9	28.0 ±4.3	53.3 ±2.5	0.0 ±0.0	15.3 ±4.7	66.7 ±6.6
	L	77.3 ±6.2	25.3 ±1.9	64.7 ±0.9	0.0 ±0.0	24.7 ±8.2	74.0 ±3.3
SPA	B (default)	39.3 ±6.8	11.3 ±2.5	20.0 ±1.6	0.0 ±0.0	3.3 ±0.9	36.7 ±5.0
	L	38.0 ±10.7	8.7 ±1.9	13.3 ±1.9	0.0 ±0.0	3.3 ±1.9	34.7 ±5.7
C-RADIOv4	SO400M	50.7 ±2.5	10.0 ±1.6	14.7 ±6.2	0.0 ±0.0	6.0 ±4.3	52.0 ±4.3
	H (default)	50.7 ±3.4	9.3 ±5.0	28.7 ±9.3	0.0 ±0.0	8.0 ±4.3	40.7 ±6.8

Table 33: Continued from Table 32. Tasks W7–W12 and the suite average.

Encoder	Setting	W7	W8	W9	W10	W11	W12	Avg.
DINOv2	S	18.7 ±5.7	45.3 ±7.4	9.3 ±2.5	50.0 ±5.9	39.3 ±11.1	6.0 ±1.6	43.50 ±1.7
	B (default)	29.3 ±3.4	51.3 ±1.9	18.7 ±0.9	47.3 ±1.9	39.3 ±2.5	2.0 ±2.8	49.39 ±1.0
	L	34.7 ±6.6	49.3 ±4.1	18.0 ±5.7	34.0 ±9.1	62.0 ±1.6	0.7 ±0.9	51.11 ±1.2
	g	22.0 ±4.9	50.0 ±3.3	15.3 ±5.2	59.3 ±2.5	64.0 ±4.3	0.0 ±0.0	50.33 ±1.0
DINOv3	S	12.0 ±1.6	16.7 ±0.9	0.0 ±0.0	29.3 ±2.5	22.0 ±1.6	0.7 ±0.9	24.56 ±1.1
	B (default)	9.3 ±0.9	25.3 ±6.6	3.3 ±0.9	31.3 ±7.7	26.7 ±5.2	1.3 ±0.9	30.44 ±1.4
	L	14.0 ±4.9	28.7 ±2.5	2.0 ±1.6	26.0 ±2.8	13.3 ±3.4	0.0 ±0.0	28.78 ±1.1
	H+	15.3 ±4.7	26.7 ±4.1	0.7 ±0.9	27.3 ±2.5	15.3 ±3.4	0.7 ±0.9	30.61 ±1.3
V-JEPA 2.1	B (default)	8.0 ±0.0	30.0 ±1.6	1.3 ±0.9	53.3 ±9.6	38.0 ±7.1	0.0 ±0.0	39.61 ±1.3
	L	2.0 ±1.6	57.3 ±7.7	0.0 ±0.0	61.3 ±3.4	82.0 ±1.6	0.0 ±0.0	37.67 ±1.1
	g	36.0 ±3.3	64.7 ±3.4	18.7 ±0.9	61.3 ±7.4	83.3 ±2.5	2.0 ±0.0	55.89 ±1.0
VC-1	B (default)	12.0 ±1.6	28.7 ±7.4	2.7 ±0.9	39.3 ±10.6	10.0 ±2.8	0.7 ±0.9	25.06 ±1.5
	L	12.0 ±1.6	30.0 ±5.7	3.3 ±2.5	43.3 ±8.4	55.3 ±5.0	0.0 ±0.0	34.17 ±1.3
SPA	B (default)	2.7 ±1.9	5.3 ±0.9	0.7 ±0.9	14.7 ±2.5	5.3 ±2.5	0.7 ±0.9	11.67 ±0.8
	L	6.0 ±2.8	4.0 ±0.0	1.3 ±0.9	13.3 ±3.4	3.3 ±0.9	0.0 ±0.0	10.50 ±1.1
C-RADIOv4	SO400M	4.0 ±1.6	10.7 ±3.8	0.0 ±0.0	24.0 ±3.3	14.0 ±3.3	0.0 ±0.0	15.50 ±0.9
	H (default)	7.3 ±0.9	5.3 ±1.9	0.7 ±0.9	20.0 ±1.6	12.0 ±1.6	2.7 ±0.9	15.44 ±1.2

K.3 Diagnostic values

Tables 34 to 39 list every diagnostic of every encoder, as defined in Appendix D.

Table 34: Spatial token statistics. Encoder-level averages over frames, videos, and tasks (Appendix D) for the 17 vision encoders. FT rows are the seven fine-tuned RoboTwin checkpoints. A dash for MAE marks a statistic that depends on patch positions; the released MAE implementation permutes its output tokens, so these values are not reported. PR is the participation ratio of the token covariance; NormHist is the norm-histogram entropy.

Encoder	Aniso.	Norm	Nbr.	PR	NormHist	NormVar	Glob.	W/B
RoboCasa-GR1								
VGGT-Ω	0.919	83.8	0.987	3.70	3.05	11.1	0.959	7.30
LingBot-Vision	0.254	9.90	0.751	7.52	3.03	3.77	0.503	103
SPA	0.201	4.89	0.722	2.38	3.10	1.21	0.434	223
DINOv2	0.248	45.9	0.609	14.6	2.36	11.4	0.500	31.9
DINOv3	0.656	7.05	0.904	5.86	3.18	0.084	0.811	12.7
MAE	0.173	23.2	–	1.65	2.86	65.2	0.359	463
V-JEPA 2.1	0.723	29.8	0.853	10.8	3.02	0.217	0.851	111
SigLIP2	0.338	47.6	0.493	22.5	2.21	20.1	0.583	47.9
InternViT	0.228	31.1	0.440	3.16	1.65	1,214	0.396	390
SigLIP	0.205	62.2	0.357	27.6	2.46	55.7	0.456	47.8
CLIP	0.411	25.3	0.541	3.05	0.821	343	0.637	198
VC-1	0.145	16.1	0.661	3.58	3.06	8.51	0.376	455
Voltron	0.441	17.3	0.763	12.5	3.02	3.40	0.666	301
Theia	0.205	14.8	0.546	17.3	3.16	19.0	0.451	41.1
C-RADIOv4	0.190	30.9	0.851	16.4	2.98	3.41	0.436	120
RADIOv2.5	0.407	16.2	0.640	1.48	1.00	521	0.497	265
ViT	0.502	24.3	0.703	16.2	1.60	10.6	0.709	4.33
RoboTwin 2.0								
VGGT-Ω	0.892	90.2	0.983	3.15	3.09	17.7	0.945	21.6
LingBot-Vision	0.248	8.75	0.752	11.1	3.13	0.464	0.501	135
SPA	0.465	3.87	0.784	6.22	2.95	0.665	0.675	166
DINOv2	0.262	45.8	0.634	13.2	2.13	9.50	0.513	45.5
DINOv3	0.686	7.13	0.918	4.91	3.15	0.067	0.829	18.0
MAE	0.574	15.8	–	6.32	2.95	14.7	0.750	301
V-JEPA 2.1	0.692	30.0	0.831	16.2	3.08	0.278	0.833	141
SigLIP2	0.319	47.9	0.477	23.6	2.22	21.2	0.567	86.9
InternViT	0.190	32.4	0.380	2.57	1.74	1,191	0.335	364
SigLIP	0.202	62.4	0.352	29.4	2.59	64.7	0.453	79.4
CLIP	0.429	24.7	0.564	2.77	0.657	345	0.651	214
VC-1	0.463	14.6	0.749	8.95	3.02	4.89	0.672	415
Voltron	0.554	16.7	0.820	16.6	3.00	2.13	0.745	275
Theia	0.218	14.9	0.570	18.5	3.16	23.5	0.464	116
C-RADIOv4	0.175	30.6	0.858	15.7	2.89	3.49	0.418	145
RADIOv2.5	0.638	9.81	0.816	2.40	0.537	115	0.771	291
ViT	0.525	24.5	0.701	17.1	1.54	7.86	0.725	8.05
VGGT-Ω FT	0.135	26.6	0.491	5.85	2.88	24.1	0.368	289
DINOv2 FT	0.262	10.0	0.804	6.83	3.13	3.75	0.509	181
DINOv3 FT	0.305	5.16	0.910	5.67	3.17	0.392	0.549	364
SigLIP2 FT	0.135	26.6	0.515	6.49	2.74	133	0.337	961
VC-1 FT	0.270	7.35	0.840	7.04	3.02	1.25	0.518	127
SPA FT	0.396	6.75	0.866	7.08	3.06	3.18	0.621	84.0
C-RADIOv4 FT	0.601	87.0	0.769	7.74	3.11	809	0.770	52.1

Table 35: Spatial-frequency token statistics. Same rows and conventions as Table 34.

Encoder	Low	Mid	High	SpEnt	Cent.	BW
RoboCasa-GR1						
VGGT-Ω	0.450	0.250	0.300	0.641	0.257	0.198
LingBot-Vision	0.356	0.344	0.300	0.716	0.286	0.187
SPA	0.296	0.299	0.406	0.761	0.336	0.203
DINOv2	0.247	0.276	0.478	0.825	0.364	0.229
DINOv3	0.450	0.315	0.235	0.677	0.253	0.178
MAE	–	–	–	–	–	–
V-JEPA 2.1	0.250	0.262	0.488	0.829	0.375	0.221
SigLIP2	0.140	0.204	0.656	0.893	0.464	0.225
InternViT	0.101	0.253	0.646	0.896	0.440	0.213
SigLIP	0.131	0.203	0.666	0.898	0.468	0.224
CLIP	0.138	0.206	0.657	0.891	0.456	0.233
VC-1	0.258	0.282	0.460	0.787	0.361	0.211
Voltron	0.323	0.285	0.392	0.759	0.330	0.212
Theia	0.230	0.275	0.496	0.841	0.380	0.216
C-RADIOv4	0.617	0.227	0.156	0.650	0.187	0.175
RADIOv2.5	0.180	0.200	0.621	0.864	0.442	0.224
ViT	0.228	0.256	0.516	0.839	0.392	0.223
RoboTwin 2.0						
VGGT-Ω	0.476	0.225	0.299	0.654	0.259	0.201
LingBot-Vision	0.333	0.314	0.354	0.765	0.310	0.199
SPA	0.299	0.276	0.425	0.781	0.342	0.207
DINOv2	0.261	0.258	0.480	0.816	0.362	0.230
DINOv3	0.466	0.275	0.258	0.677	0.259	0.185
MAE	–	–	–	–	–	–
V-JEPA 2.1	0.231	0.258	0.510	0.843	0.386	0.221
SigLIP2	0.139	0.196	0.665	0.892	0.467	0.224
InternViT	0.084	0.233	0.683	0.904	0.467	0.216
SigLIP	0.129	0.190	0.681	0.899	0.473	0.223
CLIP	0.130	0.210	0.660	0.895	0.454	0.229
VC-1	0.232	0.278	0.489	0.824	0.375	0.212
Voltron	0.333	0.282	0.385	0.759	0.325	0.210
Theia	0.237	0.272	0.492	0.837	0.376	0.215
C-RADIOv4	0.616	0.222	0.162	0.653	0.188	0.176
RADIOv2.5	0.220	0.223	0.557	0.847	0.409	0.221
ViT	0.222	0.251	0.527	0.852	0.395	0.223
VGGT-Ω FT	–	–	–	–	–	–
DINOv2 FT	0.453	0.288	0.260	0.686	0.248	0.190
DINOv3 FT	0.569	0.289	0.142	0.579	0.204	0.145
SigLIP2 FT	0.269	0.263	0.468	0.809	0.359	0.211
VC-1 FT	0.462	0.277	0.261	0.683	0.263	0.187
SPA FT	0.464	0.279	0.257	0.684	0.261	0.189
C-RADIOv4 FT	0.276	0.230	0.494	0.848	0.361	0.235

Table 36: Temporal token statistics. Same rows and conventions as Table 34. EigEnt is the entropy of the eigenvalues of the covariance of pooled tokens over time, not a Fourier spectrum; TrajLen and Drift both sum frame-to-frame changes and are near-duplicates.

Encoder	Drift	CosShift	Var	PR	EigEnt	AC1	TrajLen	PatchDrift	NormHist	TrajW/B
RoboCasa-GR1										
VGGT-Ω	7.41	6.46×10^{-3}	0.041	2.93	1.51	0.330	7.25	10.4	0.865	0.658
LingBot-Vision	0.573	7.43×10^{-3}	8.03×10^{-4}	4.32	2.08	0.641	0.560	1.98	0.826	0.536
SPA	0.178	4.63×10^{-3}	1.57×10^{-4}	3.03	1.61	0.665	0.174	0.779	0.826	0.094
DINOv2	5.53	0.034	0.055	5.59	2.38	0.587	5.41	19.5	0.845	0.966
DINOv3	0.877	0.013	3.92×10^{-3}	6.08	2.45	0.607	0.858	1.50	0.850	0.573
MAE	0.512	2.22×10^{-3}	2.02×10^{-3}	2.23	1.30	0.647	0.501	–	–	0.063
V-JEPA 2.1	1.08	1.03×10^{-3}	3.27×10^{-3}	5.75	2.41	0.604	1.05	5.35	0.834	0.432
SigLIP2	4.59	0.014	0.043	6.71	2.58	0.551	4.49	24.9	0.860	0.902
InternViT	2.16	0.015	6.16×10^{-3}	4.58	2.17	0.461	2.12	13.2	0.733	1.44
SigLIP	6.80	0.030	0.089	7.37	2.68	0.540	6.66	39.8	0.867	0.871
CLIP	1.58	5.53×10^{-3}	3.82×10^{-3}	7.07	2.62	0.564	1.54	11.6	0.802	0.831
VC-1	0.395	2.38×10^{-3}	8.88×10^{-4}	3.19	1.73	0.638	0.387	2.98	0.806	0.091
Voltron	0.481	1.07×10^{-3}	9.19×10^{-4}	4.25	2.09	0.668	0.470	3.63	0.788	0.127
Theia	1.60	0.033	6.72×10^{-3}	5.93	2.42	0.617	1.57	5.15	0.818	0.665
C-RADIOv4	1.59	8.47×10^{-3}	6.02×10^{-3}	4.02	2.03	0.695	1.55	7.56	0.825	0.495
RADIOv2.5	1.41	0.011	3.74×10^{-3}	2.19	1.47	0.485	1.38	7.72	0.610	0.363
ViT	6.89	0.085	0.093	8.93	2.80	0.549	6.74	12.4	0.824	1.11
RoboTwin 2.0										
VGGT-Ω	5.33	5.75×10^{-3}	0.106	2.32	1.17	0.608	5.15	8.72	0.738	0.618
LingBot-Vision	0.538	0.011	1.23×10^{-3}	2.84	1.49	0.650	0.520	1.91	0.808	0.411
SPA	0.199	5.69×10^{-3}	3.53×10^{-4}	1.94	1.00	0.631	0.192	0.646	0.792	0.136
DINOv2	5.34	0.035	0.097	4.06	1.90	0.581	5.16	16.6	0.804	0.571
DINOv3	0.757	0.011	4.38×10^{-3}	3.66	1.80	0.624	0.731	1.31	0.837	0.359
MAE	0.509	1.78×10^{-3}	3.26×10^{-3}	1.83	0.893	0.656	0.492	–	–	0.137
V-JEPA 2.1	1.10	1.29×10^{-3}	5.42×10^{-3}	3.40	1.70	0.620	1.06	5.74	0.809	0.286
SigLIP2	3.72	0.011	0.043	4.11	1.94	0.572	3.59	21.6	0.832	0.587
InternViT	2.15	0.013	8.20×10^{-3}	3.03	1.59	0.498	2.08	13.8	0.698	0.599
SigLIP	5.54	0.023	0.086	4.52	2.05	0.556	5.35	34.1	0.828	0.588
CLIP	1.60	6.54×10^{-3}	5.50×10^{-3}	4.15	1.95	0.554	1.55	13.8	0.740	0.699
VC-1	0.509	4.52×10^{-3}	2.70×10^{-3}	2.27	1.15	0.638	0.491	2.77	0.769	0.080
Voltron	0.596	2.50×10^{-3}	2.66×10^{-3}	2.69	1.34	0.657	0.575	3.21	0.770	0.123
Theia	1.17	0.022	6.76×10^{-3}	3.38	1.68	0.650	1.13	4.41	0.792	0.491
C-RADIOv4	2.05	0.018	0.011	3.31	1.65	0.624	1.98	7.96	0.798	0.432
RADIOv2.5	0.617	6.22×10^{-3}	1.98×10^{-3}	2.49	1.32	0.643	0.596	2.02	0.701	0.236
ViT	5.36	0.064	0.095	4.58	2.00	0.594	5.18	10.4	0.734	0.699
VGGT-Ω FT	1.68	0.031	0.016	2.59	1.23	0.776	1.63	8.85	0.807	3.15
DINOv2 FT	0.744	0.022	8.02×10^{-3}	2.31	1.13	0.752	0.719	1.87	0.731	1.93
DINOv3 FT	0.460	0.032	4.20×10^{-3}	2.56	1.16	0.795	0.445	0.914	0.734	4.05
SigLIP2 FT	1.55	0.018	0.047	2.51	1.16	0.793	1.50	5.14	0.663	3.19
VC-1 FT	0.692	0.035	6.72×10^{-3}	2.73	1.27	0.768	0.670	1.47	0.742	2.51
SPA FT	0.710	0.026	4.41×10^{-3}	2.49	1.29	0.713	0.686	1.38	0.767	0.871
C-RADIOv4 FT	8.05	0.010	0.175	3.39	1.66	0.618	7.78	28.7	0.771	0.720

Table 37: Readout probes and the texture diagnostic. Held-out normalized mean squared error for IDM and state; per-coordinate mean absolute object-position error in meters, with no mean-predictor baseline; texture-induced variance of the normalized action chunk of a trained policy. Lower is better throughout. FT rows are the fine-tuned RoboTwin encoders; RoboCasa probes were not rerun after fine-tuning.

	RoboCasa-GR1		RoboTwin 2.0			
Encoder	**IDM**	**State**	**IDM**	**State**	**Object (m)**	**Texture**
VGGT-Ω	0.249	0.165	0.164	0.173	0.051	9.63×10^{-4}
LingBot-Vision	0.188	0.120	0.147	0.064	0.052	1.25×10^{-3}
SPA	0.357	0.202	0.439	0.268	0.065	3.06×10^{-3}
DINOv2	0.216	0.143	0.188	0.124	0.050	1.20×10^{-3}
DINOv3	0.198	0.130	0.142	0.082	0.053	4.50×10^{-3}
MAE	0.209	0.133	0.269	0.154	0.065	3.57×10^{-3}
V-JEPA 2.1	0.177	0.115	0.120	0.081	0.066	3.72×10^{-3}
Xiaomi-Robotics-1	0.158	0.450	0.150	0.546	0.065	2.31×10^{-3}
DepthVLM	0.255	0.466	0.153	0.249	0.066	1.86×10^{-3}
Qwen3-VL	0.278	0.481	0.177	0.378	0.065	4.49×10^{-3}
SigLIP2	0.227	0.159	0.158	0.099	0.052	4.61×10^{-3}
InternViT	0.231	0.417	0.246	0.215	0.057	4.45×10^{-3}
SigLIP	0.253	0.179	0.216	0.125	0.053	6.71×10^{-3}
CLIP	0.233	0.166	0.362	0.185	0.060	5.75×10^{-3}
VC-1	0.201	0.112	0.206	0.095	0.054	2.28×10^{-3}
Voltron	0.226	0.146	0.302	0.228	0.065	4.73×10^{-3}
Theia	0.211	0.133	0.174	0.086	0.052	2.21×10^{-3}
C-RADIOv4	0.410	0.622	0.405	0.358	0.065	3.42×10^{-3}
RADIOv2.5	0.430	0.319	0.312	0.253	0.065	3.55×10^{-3}
ViT	0.252	0.187	0.257	0.182	0.053	5.98×10^{-3}
VGGT-Ω FT	–	–	0.018	0.014	0.061	2.89×10^{-4}
DINOv2 FT	–	–	0.076	0.024	0.052	8.83×10^{-4}
DINOv3 FT	–	–	0.030	8.83×10^{-3}	0.058	9.28×10^{-5}
SigLIP2 FT	–	–	0.025	0.012	0.055	1.65×10^{-4}
VC-1 FT	–	–	0.032	0.019	0.022	2.94×10^{-4}
SPA FT	–	–	0.055	0.037	0.063	4.71×10^{-4}
C-RADIOv4 FT	–	–	0.057	0.033	0.103	3.90×10^{-4}

Table 38: Shape probe, Gaussian target. AUC is the area under the loss curve normalized by its initial value; t_{50}, t_{25}, and t_{10} are the fractions of training needed to reach 50, 25, and 10% of the initial loss; Residual is the final target discrepancy; Dist. pres. (feature-distance preservation) is the Spearman correlation between pairwise distances before and after the fit. FT rows are fine-tuned RoboTwin checkpoints.

	Gaussian target					
Encoder	**AUC**	t_{50}	t_{25}	t_{10}	**Residual**	**Dist. pres.**
RoboCasa-GR1						
VGGT-Ω	0.322	0.167	0.537	0.933	0.032	0.871
LingBot-Vision	0.136	0.087	0.146	0.321	0.013	0.709
SPA	0.143	0.025	0.154	0.521	0.013	0.715
DINOv2	0.085	0.058	0.087	0.150	5.09×10^{-3}	0.534
DINOv3	0.100	0.067	0.096	0.129	3.79×10^{-3}	0.559
MAE	0.148	0.037	0.150	0.483	0.014	0.591
V-JEPA 2.1	0.227	0.204	0.237	0.362	0.032	0.787
Xiaomi-Robotics-1	0.248	0.225	0.246	0.283	8.64×10^{-3}	0.635
DepthVLM	0.104	0.083	0.113	0.138	0.016	0.655
Qwen3-VL	0.143	0.121	0.154	0.212	8.03×10^{-3}	0.630
SigLIP2	0.155	0.125	0.183	0.292	7.04×10^{-3}	0.638
InternViT	0.178	0.146	0.200	0.388	9.03×10^{-3}	0.698
SigLIP	0.133	0.108	0.146	0.237	0.012	0.549
CLIP	0.201	0.179	0.221	0.358	0.013	0.613
VC-1	0.122	0.042	0.129	0.371	0.010	0.807
Voltron	0.180	0.067	0.233	0.575	0.013	0.783
Theia	0.098	0.075	0.100	0.138	5.78×10^{-3}	0.583
C-RADIOv4	0.123	0.062	0.087	0.179	0.012	0.386
RADIOv2.5	0.139	0.067	0.121	0.404	6.94×10^{-3}	0.463
ViT	0.077	0.054	0.079	0.113	4.51×10^{-3}	0.396
RoboTwin 2.0						
VGGT-Ω	0.106	0.049	0.082	0.146	6.12×10^{-3}	0.858
LingBot-Vision	0.039	0.017	0.023	0.048	1.39×10^{-3}	0.622
SPA	0.034	5.32×10^{-3}	0.012	0.031	3.93×10^{-3}	0.723
DINOv2	0.037	0.016	0.023	0.040	2.13×10^{-3}	0.472
DINOv3	0.037	0.015	0.022	0.041	5.13×10^{-3}	0.524
MAE	0.033	0.011	0.016	0.022	3.21×10^{-3}	0.748
V-JEPA 2.1	0.056	0.033	0.043	0.093	4.75×10^{-3}	0.705
Xiaomi-Robotics-1	0.080	0.054	0.062	0.104	3.82×10^{-3}	0.567
DepthVLM	0.044	0.020	0.027	0.053	3.49×10^{-3}	0.632
Qwen3-VL	0.060	0.024	0.071	0.094	5.43×10^{-3}	0.583
SigLIP2	0.052	0.032	0.041	0.060	4.13×10^{-3}	0.617
InternViT	0.051	0.029	0.038	0.068	5.83×10^{-3}	0.680
SigLIP	0.049	0.022	0.040	0.068	3.15×10^{-3}	0.544
CLIP	0.056	0.041	0.050	0.063	4.03×10^{-3}	0.570
VC-1	0.031	9.57×10^{-3}	0.013	0.017	1.86×10^{-3}	0.743
Voltron	0.030	0.012	0.018	0.027	2.75×10^{-3}	0.762
Theia	0.036	0.018	0.024	0.031	2.16×10^{-3}	0.473
C-RADIOv4	0.045	9.57×10^{-3}	0.038	0.057	4.58×10^{-3}	0.646
RADIOv2.5	0.041	0.014	0.029	0.052	5.53×10^{-3}	0.745
ViT	0.033	0.012	0.017	0.036	2.10×10^{-3}	0.336
VGGT-Ω FT	0.036	0.014	0.020	0.044	2.09×10^{-3}	0.489
DINOv2 FT	0.029	7.45×10^{-3}	0.013	0.022	2.16×10^{-3}	0.536
DINOv3 FT	0.029	7.45×10^{-3}	0.012	0.029	2.24×10^{-3}	0.616
SigLIP2 FT	0.032	0.011	0.017	0.031	2.66×10^{-3}	0.584
VC-1 FT	0.025	6.38×10^{-3}	9.57×10^{-3}	0.013	1.70×10^{-3}	0.574
SPA FT	0.028	4.26×10^{-3}	0.013	0.028	2.80×10^{-3}	0.545
C-RADIOv4 FT	0.032	0.014	0.018	0.029	1.80×10^{-3}	0.502

Table 39: Shape probe, hard-sphere target. Same conventions as Table 38. Tail is the coefficient of variation of the loss over the last 10% of training, and S/G AUC is the ratio of sphere AUC to Gaussian AUC. No sphere run reaches 10% of its initial loss, so t_{10} is omitted for this target.

	Hard-sphere target						
Encoder	**AUC**	t_{50}	t_{25}	**Residual**	**Dist. pres.**	**Tail**	**S/G AUC**
RoboCasa-GR1							
VGGT-Ω	0.297	0.017	0.646	0.654	0.875	0.032	0.924
LingBot-Vision	0.201	0.021	0.217	0.368	0.589	0.028	1.48
SPA	0.198	0.033	0.104	0.401	0.787	0.024	1.38
DINOv2	0.251	0.021	0.254	0.394	0.392	0.043	2.97
DINOv3	0.248	0.017	0.275	0.386	0.534	0.028	2.48
MAE	0.218	0.033	0.125	0.437	0.650	0.048	1.48
V-JEPA 2.1	0.289	0.013	0.483	0.409	0.573	0.036	1.28
Xiaomi-Robotics-1	0.281	0.017	0.604	0.595	0.578	0.019	1.13
DepthVLM	0.250	0.013	0.300	0.403	0.442	0.038	2.41
Qwen3-VL	0.249	0.017	0.346	0.433	0.426	0.041	1.74
SigLIP2	0.235	0.021	0.312	0.365	0.397	0.030	1.51
InternViT	0.265	0.017	0.438	0.574	0.636	0.025	1.49
SigLIP	0.237	0.025	0.250	0.407	0.597	0.037	1.78
CLIP	0.308	0.017	0.496	0.526	0.574	0.029	1.53
VC-1	0.217	0.037	0.067	0.442	0.816	0.032	1.78
Voltron	0.223	0.017	0.225	0.390	0.746	0.029	1.24
Theia	0.247	0.029	0.208	0.444	0.434	0.038	2.53
C-RADIOv4	0.209	0.025	0.146	0.359	0.499	0.033	1.70
RADIOv2.5	0.247	0.017	0.346	0.313	0.382	0.015	1.78
ViT	0.290	0.021	0.596	0.449	0.429	0.036	3.78
RoboTwin 2.0							
VGGT-Ω	0.229	4.26×10^{-3}	0.141	0.561	0.739	0.033	2.15
LingBot-Vision	0.140	5.32×10^{-3}	0.046	0.288	0.363	8.94×10^{-3}	3.63
SPA	0.189	4.26×10^{-3}	0.043	0.342	0.499	0.026	5.49
DINOv2	0.159	6.38×10^{-3}	0.041	0.298	0.220	0.012	4.35
DINOv3	0.173	4.26×10^{-3}	0.083	0.298	0.344	0.011	4.64
MAE	0.174	4.26×10^{-3}	0.055	0.331	0.558	0.022	5.32
V-JEPA 2.1	0.161	3.19×10^{-3}	0.098	0.301	0.463	0.012	2.89
Xiaomi-Robotics-1	0.167	4.26×10^{-3}	0.196	0.303	0.105	0.014	2.10
DepthVLM	0.149	4.26×10^{-3}	0.050	0.308	0.353	0.015	3.42
Qwen3-VL	0.148	5.32×10^{-3}	0.088	0.304	0.300	0.013	2.47
SigLIP2	0.155	5.32×10^{-3}	0.090	0.306	0.295	0.015	2.96
InternViT	0.141	5.32×10^{-3}	0.067	0.285	0.396	0.010	2.77
SigLIP	0.159	8.51×10^{-3}	0.074	0.319	0.333	0.018	3.26
CLIP	0.203	4.26×10^{-3}	0.203	0.319	0.262	0.018	3.65
VC-1	0.168	4.26×10^{-3}	0.035	0.324	0.650	0.018	5.45
Voltron	0.159	4.26×10^{-3}	0.032	0.317	0.513	0.018	5.33
Theia	0.168	6.38×10^{-3}	0.056	0.299	0.359	0.013	4.62
C-RADIOv4	0.145	5.32×10^{-3}	0.044	0.287	0.467	0.042	3.22
RADIOv2.5	0.216	0.026	0.132	0.279	0.386	0.023	5.30
ViT	0.215	6.38×10^{-3}	0.216	0.367	0.215	0.023	6.53
VGGT-Ω FT	0.136	7.45×10^{-3}	0.032	0.276	0.307	0.010	3.76
DINOv2 FT	0.131	8.51×10^{-3}	0.028	0.276	0.462	7.60×10^{-3}	4.57
DINOv3 FT	0.135	6.38×10^{-3}	0.027	0.275	0.400	9.79×10^{-3}	4.65
SigLIP2 FT	0.122	9.57×10^{-3}	0.027	0.271	0.473	7.78×10^{-3}	3.78
VC-1 FT	0.148	6.38×10^{-3}	0.023	0.277	0.343	9.45×10^{-3}	5.88
SPA FT	0.154	6.38×10^{-3}	0.027	0.280	0.523	9.19×10^{-3}	5.50
C-RADIOv4 FT	0.134	0.012	0.034	0.277	0.280	9.19×10^{-3}	4.13

K.4 ALL ASSOCIATIONS

Tables 40 and 41 report, for every diagnostic, the association with suite success together with its uncertainty and false-discovery control within each cohort, grouped by diagnostic family.

Table 40: All RoboCasa-GR1 associations. Spearman ρ between each diagnostic and suite success, 95% bootstrap interval, two-sided permutation p, and Benjamini–Hochberg q within the cohort. Bold ρ marks $p < 0.05$; bold q marks $q < 0.10$. Every diagnostic uses the 17 vision encoders, or 16 for position-dependent statistics, which leave out MAE.

	RoboCasa single-task				RoboCasa multi-task			
Diagnostic	ρ	95% CI	p	q	ρ	95% CI	p	q
Spatial								
Anisotropy	+0.19	[-0.38, +0.68]	0.470	0.914	+0.21	[-0.33, +0.68]	0.417	0.971
Mean token norm	+0.02	[-0.53, +0.53]	0.943	0.943	+0.12	[-0.53, +0.64]	0.658	0.971
Neighbor cosine	+0.31	[-0.25, +0.74]	0.248	0.867	+0.23	[-0.32, +0.71]	0.398	0.971
Participation ratio	+0.10	[-0.46, +0.61]	0.702	0.914	+0.14	[-0.42, +0.66]	0.585	0.971
Norm-histogram entropy	**+0.56**	[+0.12, +0.86]	0.021	0.255	+0.41	[-0.11, +0.81]	0.104	0.971
Token-norm variance	-0.34	[-0.74, +0.16]	0.187	0.758	-0.17	[-0.64, +0.35]	0.521	0.971
Token-to-global cosine	+0.26	[-0.29, +0.68]	0.312	0.914	+0.27	[-0.25, +0.66]	0.297	0.971
Within/between-image var.	-0.22	[-0.75, +0.38]	0.403	0.914	-0.34	[-0.77, +0.23]	0.187	0.971
Temporal								
Temporal drift	-0.06	[-0.62, +0.49]	0.811	0.914	+0.12	[-0.43, +0.62]	0.644	0.971
Temporal cosine shift	-0.14	[-0.59, +0.32]	0.593	0.914	+0.05	[-0.37, +0.46]	0.836	0.971
Temporal variance	-0.07	[-0.53, +0.48]	0.796	0.914	+0.11	[-0.44, +0.56]	0.668	0.971
Temporal participation ratio	-0.10	[-0.67, +0.48]	0.694	0.914	+0.00	[-0.55, +0.55]	0.995	0.998
Temporal eigenvalue entropy	-0.12	[-0.67, +0.44]	0.651	0.914	-0.03	[-0.58, +0.51]	0.901	0.971
Lag-1 autocorrelation	+0.06	[-0.50, +0.61]	0.818	0.914	-0.07	[-0.62, +0.52]	0.783	0.971
Trajectory length	-0.06	[-0.63, +0.51]	0.810	0.914	+0.12	[-0.44, +0.62]	0.648	0.971
Patch temporal drift	-0.20	[-0.66, +0.28]	0.450	0.914	-0.03	[-0.60, +0.44]	0.902	0.971
Temporal norm-histogram entropy	+0.39	[-0.19, +0.81]	0.136	0.606	+0.48	[-0.05, +0.83]	0.059	0.970
Within/between-trajectory var.	-0.05	[-0.51, +0.44]	0.856	0.914	+0.13	[-0.37, +0.56]	0.615	0.971
Frequency								
Low-frequency share	+0.40	[-0.16, +0.80]	0.131	0.606	+0.30	[-0.28, +0.79]	0.254	0.971
Mid-frequency share	**+0.55**	[+0.07, +0.85]	0.030	0.294	+0.45	[-0.12, +0.85]	0.082	0.971
High-frequency share	-0.41	[-0.80, +0.15]	0.117	0.606	-0.32	[-0.79, +0.26]	0.221	0.971
Spectral entropy	-0.44	[-0.83, +0.11]	0.092	0.606	-0.34	[-0.81, +0.26]	0.191	0.971
Spectral centroid	-0.42	[-0.84, +0.15]	0.104	0.606	-0.33	[-0.84, +0.26]	0.210	0.971
Spectral bandwidth	-0.24	[-0.78, +0.36]	0.376	0.914	-0.18	[-0.79, +0.44]	0.511	0.971
Readout probes								
IDM error	**-0.64**	[-0.90, -0.14]	0.007	0.174	**-0.64**	[-0.93, -0.13]	0.006	0.170
State error	**-0.68**	[-0.86, -0.28]	0.003	0.163	**-0.64**	[-0.87, -0.18]	0.007	0.170
Object-position error	–	–	–	–	–	–	–	–
Trained policy								
Texture sensitivity	–	–	–	–	–	–	–	–
Shaping probe								
Gauss AUC	+0.04	[-0.54, +0.57]	0.876	0.914	-0.04	[-0.57, +0.49]	0.893	0.971
Gauss t_{50}	+0.06	[-0.44, +0.50]	0.830	0.914	+0.09	[-0.45, +0.54]	0.736	0.971
Gauss t_{25}	+0.21	[-0.42, +0.69]	0.415	0.914	+0.10	[-0.50, +0.59]	0.710	0.971
Gauss t_{10}	+0.12	[-0.48, +0.64]	0.652	0.914	-0.04	[-0.59, +0.51]	0.879	0.971
Gauss residual	+0.08	[-0.47, +0.61]	0.774	0.914	+0.00	[-0.56, +0.55]	0.998	0.998
Gauss distance preservation	**+0.57**	[-0.01, +0.90]	0.019	0.255	+0.38	[-0.23, +0.78]	0.133	0.971
Gauss adapter norm	-0.11	[-0.57, +0.42]	0.678	0.914	-0.11	[-0.59, +0.40]	0.680	0.971
Gauss early loss	+0.04	[-0.55, +0.57]	0.869	0.914	+0.01	[-0.55, +0.51]	0.965	0.998
Gauss rebound	-0.17	[-0.65, +0.39]	0.507	0.914	-0.08	[-0.63, +0.52]	0.755	0.971
Gauss tail oscillation	+0.18	[-0.38, +0.72]	0.477	0.914	+0.26	[-0.37, +0.79]	0.307	0.971
Gauss relative drop	-0.12	[-0.65, +0.45]	0.644	0.914	+0.04	[-0.51, +0.58]	0.874	0.971
Sphere AUC	+0.03	[-0.52, +0.53]	0.921	0.941	+0.10	[-0.48, +0.60]	0.702	0.971
Sphere t_{50}	-0.04	[-0.50, +0.42]	0.868	0.914	-0.04	[-0.51, +0.45]	0.876	0.971
Sphere t_{25}	-0.08	[-0.64, +0.47]	0.748	0.914	+0.03	[-0.53, +0.55]	0.911	0.971
Sphere t_{10}	–	–	–	–	–	–	–	–
Sphere residual	+0.13	[-0.49, +0.66]	0.622	0.914	+0.13	[-0.51, +0.67]	0.615	0.971
Sphere distance preservation	+0.33	[-0.31, +0.80]	0.201	0.758	+0.18	[-0.42, +0.69]	0.497	0.971
Sphere adapter norm	+0.13	[-0.43, +0.67]	0.618	0.914	+0.12	[-0.42, +0.63]	0.637	0.971
Sphere early loss	-0.13	[-0.61, +0.43]	0.622	0.914	-0.07	[-0.61, +0.48]	0.803	0.971
Sphere rebound	+0.41	[-0.08, +0.73]	0.104	0.606	+0.21	[-0.25, +0.60]	0.410	0.971
Sphere tail oscillation	+0.13	[-0.37, +0.59]	0.606	0.914	+0.23	[-0.35, +0.68]	0.377	0.971
Sphere relative drop	-0.04	[-0.58, +0.52]	0.869	0.914	-0.12	[-0.62, +0.49]	0.644	0.971
Sphere/Gauss residual ratio	-0.04	[-0.57, +0.47]	0.870	0.914	+0.09	[-0.44, +0.60]	0.737	0.971
Sphere/Gauss AUC ratio	-0.18	[-0.72, +0.42]	0.490	0.914	-0.07	[-0.58, +0.47]	0.798	0.971

Table 41: All RoboTwin 2.0 associations. Same format as Table 40. Twin+FT adds the seven fine-tuned checkpoints, 24 encoders in total (23 for position-dependent token statistics).

	RoboTwin multi-task				RoboTwin + fine-tuned			
Diagnostic	ρ	95% CI	p	q	ρ	95% CI	p	q
Spatial								
Anisotropy	-0.03	[-0.57, +0.52]	0.915	0.972	-0.36	[-0.72, +0.09]	0.083	0.271
Mean token norm	+0.44	[-0.14, +0.83]	0.080	0.417	+0.14	[-0.30, +0.53]	0.518	0.755
Neighbor cosine	-0.09	[-0.57, +0.44]	0.758	0.921	-0.05	[-0.45, +0.36]	0.805	0.993
Participation ratio	+0.19	[-0.33, +0.62]	0.473	0.823	-0.03	[-0.45, +0.38]	0.896	0.993
Norm-histogram entropy	+0.43	[-0.11, +0.79]	0.089	0.417	+0.34	[-0.08, +0.67]	0.102	0.271
Token-norm variance	+0.01	[-0.45, +0.51]	0.958	0.975	-0.04	[-0.41, +0.35]	0.857	0.993
Token-to-global cosine	-0.05	[-0.57, +0.48]	0.854	0.947	-0.35	[-0.70, +0.08]	0.090	0.271
Within/between-image var.	-0.41	[-0.77, +0.10]	0.102	0.434	-0.03	[-0.44, +0.37]	0.891	0.993
Temporal								
Temporal drift	+0.34	[-0.19, +0.76]	0.183	0.472	+0.12	[-0.31, +0.54]	0.560	0.772
Temporal cosine shift	+0.16	[-0.34, +0.60]	0.546	0.823	+0.38	[-0.04, +0.71]	0.067	0.261
Temporal variance	+0.44	[-0.17, +0.87]	0.081	0.417	+0.34	[-0.11, +0.70]	0.102	0.271
Temporal participation ratio	+0.15	[-0.40, +0.61]	0.565	0.823	-0.09	[-0.54, +0.35]	0.683	0.893
Temporal eigenvalue entropy	+0.16	[-0.37, +0.63]	0.535	0.823	-0.19	[-0.59, +0.26]	0.379	0.603
Lag-1 autocorrelation	-0.19	[-0.60, +0.26]	0.471	0.823	+0.33	[-0.10, +0.67]	0.113	0.273
Trajectory length	+0.34	[-0.18, +0.78]	0.182	0.472	+0.13	[-0.31, +0.54]	0.556	0.772
Patch temporal drift	+0.32	[-0.21, +0.74]	0.220	0.488	+0.02	[-0.39, +0.43]	0.941	0.994
Temporal norm-histogram entropy	+0.34	[-0.17, +0.72]	0.196	0.472	+0.02	[-0.39, +0.44]	0.916	0.994
Within/between-trajectory var.	+0.30	[-0.23, +0.76]	0.233	0.496	**+0.54**	[+0.17, +0.77]	0.006	**0.045**
Frequency								
Low-frequency share	+0.10	[-0.44, +0.61]	0.723	0.921	+0.25	[-0.19, +0.62]	0.253	0.431
Mid-frequency share	+0.07	[-0.46, +0.54]	0.805	0.929	+0.31	[-0.13, +0.66]	0.164	0.362
High-frequency share	-0.06	[-0.58, +0.46]	0.820	0.929	-0.28	[-0.63, +0.14]	0.202	0.381
Spectral entropy	-0.08	[-0.60, +0.44]	0.777	0.921	-0.30	[-0.65, +0.13]	0.171	0.362
Spectral centroid	-0.08	[-0.61, +0.47]	0.771	0.921	-0.29	[-0.67, +0.15]	0.189	0.381
Spectral bandwidth	+0.11	[-0.47, +0.65]	0.687	0.921	-0.18	[-0.59, +0.26]	0.419	0.648
Readout probes								
IDM error	**-0.82**	[-0.94, -0.51]	<0.001	**0.002**	**-0.76**	[-0.92, -0.43]	<0.001	**0.005**
State error	**-0.69**	[-0.90, -0.26]	0.003	**0.042**	**-0.71**	[-0.89, -0.35]	<0.001	**0.005**
Object-position error	**-0.70**	[-0.93, -0.22]	0.002	**0.042**	**-0.58**	[-0.79, -0.25]	0.003	**0.029**
Trained policy								
Texture sensitivity	-0.40	[-0.82, +0.20]	0.115	0.434	**-0.59**	[-0.78, -0.27]	0.003	**0.029**
Shaping probe								
Gauss AUC	+0.35	[-0.16, +0.73]	0.171	0.472	-0.09	[-0.46, +0.33]	0.680	0.893
Gauss t_{50}	**+0.65**	[+0.18, +0.89]	0.006	**0.078**	+0.15	[-0.26, +0.54]	0.487	0.731
Gauss t_{25}	+0.39	[-0.12, +0.79]	0.119	0.434	-0.00	[-0.42, +0.42]	0.986	0.995
Gauss t_{10}	+0.33	[-0.14, +0.70]	0.192	0.472	+0.01	[-0.36, +0.41]	0.956	0.994
Gauss residual	-0.01	[-0.57, +0.53]	0.975	0.975	-0.25	[-0.63, +0.18]	0.236	0.416
Gauss distance preservation	-0.16	[-0.69, +0.39]	0.544	0.823	-0.26	[-0.65, +0.22]	0.227	0.413
Gauss adapter norm	-0.24	[-0.78, +0.34]	0.347	0.708	-0.34	[-0.70, +0.08]	0.102	0.271
Gauss early loss	+0.42	[-0.07, +0.78]	0.090	0.417	-0.00	[-0.40, +0.40]	0.995	0.995
Gauss rebound	-0.33	[-0.71, +0.21]	0.203	0.472	**-0.41**	[-0.70, -0.05]	0.045	0.193
Gauss tail oscillation	+0.15	[-0.42, +0.61]	0.553	0.823	-0.05	[-0.46, +0.37]	0.823	0.993
Gauss relative drop	+0.19	[-0.39, +0.73]	0.473	0.823	+0.07	[-0.39, +0.52]	0.761	0.970
Sphere AUC	-0.33	[-0.85, +0.29]	0.202	0.472	**-0.49**	[-0.85, -0.03]	0.016	0.102
Sphere t_{50}	-0.01	[-0.53, +0.53]	0.974	0.975	+0.22	[-0.23, +0.63]	0.307	0.505
Sphere t_{25}	+0.03	[-0.50, +0.60]	0.897	0.972	-0.35	[-0.66, +0.08]	0.097	0.271
Sphere t_{10}	–	–	–	–	–	–	–	–
Sphere residual	-0.09	[-0.75, +0.51]	0.744	0.921	**-0.42**	[-0.79, +0.04]	0.041	0.188
Sphere distance preservation	-0.10	[-0.65, +0.47]	0.707	0.921	-0.03	[-0.47, +0.39]	0.876	0.993
Sphere adapter norm	**-0.49**	[-0.82, +0.04]	0.048	0.349	**-0.61**	[-0.83, -0.27]	0.002	**0.029**
Sphere early loss	-0.11	[-0.64, +0.51]	0.678	0.921	**-0.44**	[-0.75, +0.01]	0.031	0.174
Sphere rebound	-0.17	[-0.76, +0.40]	0.501	0.823	-0.31	[-0.72, +0.15]	0.136	0.314
Sphere tail oscillation	**-0.57**	[-0.91, +0.02]	0.019	0.165	**-0.58**	[-0.88, -0.15]	0.003	**0.029**
Sphere relative drop	+0.34	[-0.24, +0.84]	0.177	0.472	**+0.43**	[-0.01, +0.78]	0.035	0.181
Sphere/Gauss residual ratio	+0.21	[-0.36, +0.63]	0.409	0.802	+0.27	[-0.15, +0.61]	0.198	0.381
Sphere/Gauss AUC ratio	**-0.61**	[-0.87, -0.17]	0.011	0.113	-0.34	[-0.65, +0.05]	0.106	0.271

L Extended Related Work

L.1 Pretrained visual representations

Different pretraining signals give a representation different biases. Contrastive learning favors invariance, masked reconstruction keeps recoverable image content, and self-distillation learns local correspondences; MAE, iBOT, and DINOv2/v3 produce patch features that transfer widely (He et al., 2020; 2022; Zhou et al., 2022; Oquab et al., 2024; Siméoni et al., 2025). Geometric models such as Depth Anything, DUSt3R, SPA, and VGGT-Ω learn depth, correspondence, and scene structure, and AFRO learns 3D representations through latent dynamics (Yang et al., 2024c; Wang et al., 2024b; Zhu et al., 2025; Wang et al., 2026a; Liang et al., 2026). CLIP and SigLIP align images with language, and Theia and RADIO distill several teachers into one student (Radford et al., 2021; Tschannen et al., 2025; Shang et al., 2024; Ranzinger et al., 2024). Robot-oriented models such as R3M, MVP, VIP, Voltron, and MCR emphasize activity, goals, or manipulation relevance (Nair et al., 2022; Xiao et al., 2022; Ma et al., 2023; Karamcheti et al., 2023; Jiang et al., 2025a), while VideoMAE, V-JEPA, and AEM add temporal prediction or action-conditioned history (Tong et al., 2022; Mur-Labadia et al., 2026; Zhou et al., 2026b). PaliGemma and Qwen-VL carry visual features into language-conditioned VLM representations (Beyer et al., 2024; Bai et al., 2025). The objective decides what a representation learns; the layer that is read and the spatial readout decide what a controller actually receives, and both belong in any report that compares encoders.

L.2 From tabletop arms to egocentric bimanual manipulation

RoboMimic and DROID established demonstration learning for tabletop arms observed by external and wrist cameras (Mandlekar et al., 2021; Khazatsky et al., 2024). ALOHA and ACT advanced fine-grained bimanual skills, and Mobile ALOHA extended them to whole-body movement with onboard top and wrist views (Zhao et al., 2023; Fu et al., 2024). UMI made wrist-view demonstrations portable across scenes and embodiments (Chi et al., 2024). DexMimicGen and GR00T broadened dual-arm manipulation to multi-fingered hands and humanoids, which makes resolving contact and coordinating both hands more demanding (Jiang et al., 2025b; Bjorck et al., 2025). Human egocentric datasets such as Ego4D and Ego-Exo4D record skilled hand–object interaction (Grauman et al., 2022; 2024); they do not specify robot actions, but they tie representation learning to everyday interaction. Together these directions point to first-person bimanual control as the setting to evaluate, because moving viewpoints, mutual occlusion, and fine contact geometry all stress the visual representation in ways that third-person benchmarks do not.

L.3 From global features to spatial and predictive policies

Global embeddings made pretrained vision easy to reuse in control (Parisi et al., 2022; Nair et al., 2022; Majumdar et al., 2023), but pooling can erase the distinctions that grasping and alignment depend on. Policies differ in how much spatial detail they keep: RT-1 compresses features with TokenLearner, ACT and Octo keep spatial inputs, and Patch Policy consumes dense pretrained tokens directly (Brohan et al., 2023b; Zhao et al., 2023; Octo Model Team et al., 2024; Zhou et al., 2026a). VLA systems such as RT-2, OpenVLA, and the $\pi_0/\pi_{0.5}$ family integrate vision and language with action prediction (Brohan et al., 2023a; Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025), and StarVLA makes the backbone and action head interchangeable (StarVLA Community, 2026). World-action models such as DreamZero and OpenWAM connect video-generative priors with action learning, while Fast-WAM keeps video co-training but skips future generation at inference (Ye et al., 2026; Wang et al., 2026d; Yuan et al., 2026). HIVE-Bench holds the action architecture fixed so that only the encoder and its token interface change between two compared policies.

L.4 Evaluating representations for control

Evaluation has moved from ranking encoders toward explaining why they transfer. Studies have examined adaptation (Majumdar et al., 2023), downstream learners (Hu et al., 2023), pretraining data (Dasari et al., 2023), and objectives (Jing et al., 2023). Others relate success to segmentation (Burns et al., 2023), objectness (Wen et al., 2025), manipulation relevance (Jiang et al., 2025a), feature statistics (Shang et al., 2024), and attention allocation (Wang et al., 2026c). Further work

Table 42: Coverage of representation evaluation. Works are ordered by first arXiv release (Year), not conference year. Ego: onboard manipulation, including mixed views; Bi: dual-arm; Patch: spatial readout, including learned pooling; F/FT: frozen and policy-fine-tuned encoders; S/M: single-/multi-task policies; Stat: feature/attention statistics; State: physical-state/pose decoding; Act: action recovery from visual transitions; Corr: metric–success analysis. ✓: reported; ×: not established; †: qualified coverage (Appendix N). Checks denote study-level coverage, not every encoder/protocol combination. The complete 23-study comparison appears in Table 44.

Study		Setting / input			Training		Diagnostics			
Work	**Year**	**Ego**	**Bi**	**Patch**	**F/FT**	**S/M**	**Stat**	**State**	**Act**	**Corr**
VC-1	2023	✓	×	✓	✓	×	×	×	×	×
Hu	2023	×	×	×	×	×	×	✓	×	✓
Burns	2023	✓	✓	✓†	×	×	×	×	×	✓
Theia	2024	✓	×	✓	✓	×	✓	×	×	✓
SPA	2024	✓	✓	✓	×	✓	×	✓	×	✓
MCR	2024	×	×	×	×	×	×	×	×	✓
Temporal Trap	2025	×	×	×	×	✓	✓	×	×	✓
DCR	2025	×	×	✓	×	×	×	×	×	✓
VER	2025	✓	×	✓	×	✓†	✓	×	×	✓
AFA	2025	×	×	✓	×	×	✓	×	×	✓
CVESC	2026	×	×	×	✓	×	×	✓	×	✓
Patch Policy	2026	✓	×	✓	×	✓	×	×	×	×
HIVE-Bench	2026	✓	✓	✓	✓†	✓	✓	✓	✓†	✓

probes physical structure (Zhu et al., 2025; Dong et al., 2026), task progression (Tsagkas et al., 2026), aggregation and local readouts (Tsagkas et al., 2025; Zhou et al., 2026a), latent geometry, action recoverability, object structure, and visual robustness (Qi et al., 2025; Nilaksh et al., 2026; Qian et al., 2024; Chapin et al., 2025; Xie et al., 2024). Each study measures one aspect of representation quality under its own policy. HIVE-Bench measures four families of diagnostics under one patch-level policy, across task sharing and adaptation, and grades each association by its evidence. Table 42 compares twelve key studies, and Appendix N documents all 23 with the evidence and the qualifications behind each entry.

M Extended Review of Visual Representations and Evaluation

We review the literature available through September 2026. The organizing unit is a representation's training signal and deployment interface. A model family, a particular checkpoint, and a feature extraction rule are different objects: changing any one can change the representation seen by a robot.

Explicit 3D inputs and temporal pretraining. PointNet, PointNet++, and Point Transformer encode unordered points using order invariance, local neighborhoods, and spatial attention (Qi et al., 2017a;b; Zhao et al., 2021). DP3, PerAct, RVT, and 3D Diffuser Actor illustrate point, voxel, rendered-view, and 3D action interfaces (Ze et al., 2024; Shridhar et al., 2022; Goyal et al., 2023; Ke et al., 2024; Zhu et al., 2024); neural fields and equivariance provide further geometric structure (Mildenhall et al., 2020; Yang et al., 2024a). These policy and architecture choices should be distinguished from pretraining. AFRO learns 3D representations without action labels or geometric reconstruction targets, using latent transition objectives (Liang et al., 2026). AEM learns from interleaved visual and action histories and exposes a single history vector (Zhou et al., 2026b). It therefore addresses temporal context without making a claim about dense spatial readout.

Representations inside world action models. DreamZero transfers a pretrained video diffusion backbone to joint future-video and action modeling (Ye et al., 2026). RepWAM instead emphasizes semantic visual–action tokenization over reconstruction-oriented video tokenizers (Wang et al., 2026b). JEPA-WAM uses pretrained V-JEPA features and a shared transition/action predictor, retaining dense current–future patch correspondence without rendering future pixels (Lin et al., 2026). These designs motivate studying the information carried by visual features, but their full-policy

results are not controlled comparisons of standalone encoders. Similarly, VGGT-Ω supports both dense geometric prediction and compact scene registers (Wang et al., 2026a); its native multi-view reconstruction setting differs from extracting image features for a shared manipulation head.

M.1 From recognition backbones to reusable visual features

Supervised recognition and architectural priors. VGG and ResNet made pretrained image classifiers practical sources of visual features (Simonyan & Zisserman, 2014; He et al., 2016). Their convolutional structure favors locality and translation sharing, and their spatial feature maps remain useful even when the pretraining loss supervises only an image label. ConvNeXt revisited this design with modern training and architectural choices (Liu et al., 2022). Vision Transformers instead expose an explicit sequence of patch embeddings (Dosovitskiy et al., 2021); DeiT improved data-efficient training, while Swin introduced hierarchical local attention (Touvron et al., 2021; Liu et al., 2021). These distinctions affect spatial resolution, receptive fields, token count, and computation. Architecture labels alone are thus an incomplete taxonomy for transfer: feature layer, stride, normalization, and the policy's readout are equally consequential.

Image self-supervision. Contrastive methods such as MoCo and SimCLR learn by distinguishing augmented views of the same image from other images (He et al., 2020; Chen et al., 2020); MoCo v3 extends this line to transformer training (Chen et al., 2021). BYOL and SimSiam show alternatives based on prediction and architectural asymmetry (Grill et al., 2020; Chen & He, 2021). Barlow Twins and VICReg regulate cross-view redundancy or feature variance and covariance (Zbontar et al., 2021; Bardes et al., 2022). These methods replace category labels with invariance assumptions. For control, the question is whether those assumptions preserve small changes in position, articulation, and contact.

Masked pretraining offers a different signal. BEiT predicts discrete visual tokens, while MAE reconstructs masked image content (Bao et al., 2022; He et al., 2022). ConvNeXt V2 joins masked learning with convolutional design (Woo et al., 2023). Reconstruction encourages retention of appearance and structure, but its loss does not identify which details are needed for action. An encoder can preserve nuisance texture or predictable background without improving manipulation. Our MAE evaluation uses unmasked feature extraction; mask ratio at pretraining and at deployment must not be conflated.

DINO, iBOT, DINOv2, and DINOv3 develop self-distillation into a strong source of global and dense visual representations (Caron et al., 2021; Zhou et al., 2022; Oquab et al., 2024; Siméoni et al., 2025). Register tokens, patch objectives, data curation, and dense-feature stabilization affect the spatial output. We remove non-spatial tokens where the wrapper specifies them instead of counting class or register tokens as patches. LingBot-Vision extends dense spatial pretraining (Fu et al., 2026). Its relevance is the possibility that pretraining favorable to dense perception transfers differently from an equally strong global image descriptor read by the same policy.

Predictive latent objectives and distribution regularization. I-JEPA predicts latent representations of image regions (Assran et al., 2023); it need not reconstruct every pixel. LeJEPA develops a self-supervised formulation with a distributional regularizer (Balestriero & LeCun, 2025). Such objectives motivate studying learned feature distributions, but do not validate a post-hoc Gaussian fitting score as a robot-selection criterion. A regularizer can help through optimization, stabilization, or information allocation, while its final scalar statistic remains weakly associated with control across heterogeneous checkpoints.

M.2 Language alignment, dense geometry, and temporal experience

Image–text supervision. CLIP and ALIGN learn visual semantics from image–text pairs (Radford et al., 2021; Jia et al., 2021); SigLIP changes the alignment objective, and SigLIP 2 extends the training recipe and representation capabilities (Zhai et al., 2023; Tschannen et al., 2025). These models expose patch features even when their best-known interface is a global embedding. Language alignment may identify task-relevant objects, but local token quality is not completely specified by image–text retrieval accuracy.

BLIP and BLIP-2 illustrate increasingly modular connections between vision encoders and language models (Li et al., 2022; 2023). InternVL, Qwen3-VL, and PaliGemma further broaden the available interfaces (Chen et al., 2024; Bai et al., 2025; Beyer et al., 2024). A raw visual-tower feature and a

visual token after multimodal transformer layers differ in both supervision and input dependence. Contextual features can carry instruction information, additional computation, and language-model priors. Comparing them with a standalone image encoder is a system comparison unless those factors are controlled. Intermediate-layer extraction is relevant because later layers may specialize for the original output interface rather than a downstream action head.

Geometry and spatial supervision. Depth Anything and its successors study transferable depth prediction (Yang et al., 2024b;c; Lin et al., 2025). DUSt3R and MASt3R learn geometric relations and correspondence, while VGGT and VGGT-Ω develop feed-forward geometric representations (Wang et al., 2024b; Leroy et al., 2024; Wang et al., 2025; 2026a). SPA uses spatially grounded pretraining for embodied representation learning (Zhu et al., 2025); DepthVLM connects dense metric depth with vision–language models (Yu et al., 2026). These approaches motivate better localization and spatial organization. They do not make every extracted feature geometrically equivalent: a single-frame slice of a geometry model does not reproduce its native multi-view reconstruction pipeline.

Related dense tasks provide complementary biases. LoFTR studies local correspondence, RAFT optical flow, SAM and SAM 2 segmentation, and Grounding DINO grounded object detection (Sun et al., 2021; Teed & Deng, 2020; Kirillov et al., 2023; Ravi et al., 2025; Liu et al., 2024). Their heads and labels differ from robot requirements, yet illustrate how spatial supervision shapes features. Inclusion in this taxonomy does not imply a completed HIVE-Bench policy evaluation.

Video and human experience. VideoMAE and VideoMAE V2 extend masked learning to video; V-JEPA, V-JEPA 2, and V-JEPA 2.1 study predictive latent representations over temporal observations (Tong et al., 2022; Wang et al., 2023; Bardes et al., 2024; Assran et al., 2025; Mur-Labadia et al., 2026). VideoPrism pursues broad video understanding (Zhao et al., 2024). Egocentric human recordings, exemplified by Ego4D, bring hands, objects, and camera motion into pretraining (Grauman et al., 2022); EgoVideo and multimodal approaches such as ImageBind broaden the signals available (Pei et al., 2024; Girdhar et al., 2023). Human video still differs from robot video in embodiment, viewpoint, action annotation, and contact dynamics.

Temporal pretraining and temporal inference should be reported separately. A video-pretrained encoder deployed on a repeated or single image does not receive the evidence available in its native setting. The current V-JEPA wrapper uses image-mode extraction. Its performance measures transfer under that interface, not the full value of video-conditioned control.

M.3 Representations learned for robots

R3M uses human video and language-related signals for manipulation representations (Nair et al., 2022). MVP and Real-World Robot Learning with Masked Visual Pre-training investigate masked learning for embodied tasks (Xiao et al., 2022; Radosavovic et al., 2022). VIP frames pretraining through goal-conditioned value structure (Ma et al., 2023). Voltron combines language-conditioned reconstruction and contrastive learning (Karamcheti et al., 2023). VC-1, introduced with CortexBench, studies a broad visual foundation model for embodied intelligence (Majumdar et al., 2023). These approaches differ in objective and in the future behavior their features make accessible to a downstream learner.

Manipulation-centric and heterogeneous robot pretraining add supervision from robot interaction and multiple embodiments (Jiang et al., 2025a; Wang et al., 2024a). MCR aligns visual observations with proprioceptive state–action dynamics, combining this alignment with action prediction and temporal contrastive learning on DROID (Jiang et al., 2025a). BridgeData V2, DROID, and Open X-Embodiment expand the tasks, viewpoints, and platforms available for learning (Walke et al., 2023; Khazatsky et al., 2024; O'Neill et al., 2024). Dataset scale must retain its unit: images, video hours, trajectories, transitions, and robot embodiments cannot be added into a meaningful common data size. Correlated frames also make nominal counts different from independent visual experience.

Distillation offers another route. Theia distills multiple vision foundation models into a robot-oriented representation (Shang et al., 2024); AM-RADIO, RADIOv2.5, and C-RADIOv4 combine complementary teachers (Ranzinger et al., 2024; 2026; NVIDIA, 2026). A student's data budget describes only its own training exposure. Its teachers may encode supervision from much

larger and overlapping corpora. We distinguish student training data from inherited teacher information when documenting these models.

M.4 The policy interface is part of the representation problem

Action Chunking with Transformers and Diffusion Policy illustrate influential transformer and generative action interfaces (Zhao et al., 2023; Chi et al., 2023). RT-1 and RT-2 connect large-scale robot learning to transformer and vision–language architectures (Brohan et al., 2023b;a). Octo, OpenVLA, π_0, and GR00T develop increasingly general policy systems (Octo Model Team et al., 2024; Kim et al., 2024; Black et al., 2024; Bjorck et al., 2025); Xiaomi-Robotics-1 and StarVLA extend the ecosystem (Xiaomi Robotics Team et al., 2026; StarVLA Community, 2026). Their results combine visual pretraining, language processing, robot data, action parameterization, and adaptation. A high-performing complete policy does not establish that its isolated visual tower is the best encoder for a different controller, and the same caution applies to the VLMs evaluated here.

Patch Policy provides especially relevant evidence that pretrained dense tokens can improve a compact control architecture (Zhou et al., 2026a). It is prior work on patch-level control. HIVE-Bench asks a complementary question: across robot-centric observations and adaptation regimes, which properties of those tokens are informative about closed-loop performance? Our global-average-pooling control quantifies the value of the patch interface itself: replacing the patch grid of DINOv2 and DINOv3 with one pooled token per view lowers RoboCasa multi-task success from 32.3% to 9.9% and from 31.2% to 6.4% (Section G.2).

M.5 From transfer benchmarks to representation diagnosis

Closed-loop benchmarks establish whether a policy works; representation diagnosis asks which properties make it work and under what conditions. Meta-World, RLBench, robosuite, robomimic, CALVIN, and LIBERO supply complementary task and generalization settings (Yu et al., 2020; James et al., 2020; Zhu et al., 2020; Mandlekar et al., 2021; Mees et al., 2022; Liu et al., 2023). Simpler investigates simulation as a tool for informing real deployment (Li et al., 2024), while RoboCasa and RoboTwin 2.0 expand environmental and manipulation diversity (Nasiriany et al., 2024; Chen et al., 2025b). These suites provide outcomes against which a diagnostic can be tested; a benchmark score alone is not a diagnostic.

We include a study in the central comparison when it relates control to an explicit representation measurement, probes recoverable information, or isolates a representation-relevant factor such as pretraining data, feature hierarchy, or aggregation. A method paper can satisfy this criterion: Theia, MCR, and AFA contain analyses that are directly relevant even though they also propose new representations or interfaces (Shang et al., 2024; Jiang et al., 2025a; Tsagkas et al., 2025). Conversely, the existence of an encoder leaderboard is not sufficient. Table 43 distinguishes factor studies, diagnostic studies, and methods with diagnostic evidence. These are literature comparisons, not claims that every method has been reproduced in HIVE-Bench.

M.6 Pretraining and the learner condition the meaning of quality

Parisi et al. isolate pretraining objectives, augmentations, and feature hierarchies across control domains (Parisi et al., 2022). Their study establishes that the extraction level and training recipe matter alongside the pretrained model name. Hu et al. make the downstream learner an explicit variable: their comparison of 14 visual models across reinforcement learning, behavior cloning, and visual-reward-based learning finds that pretraining effectiveness depends strongly on the learning method (Hu et al., 2023). Consequently, a representation ranking cannot be interpreted independently of the learner that produced it.

Data-centric studies identify a second source of conditionality. Dasari et al. compare pretraining distributions and evaluate transfer in simulation and on a real robot, finding that image distribution can matter more than nominal dataset size and that simulated rankings need not predict real-world rankings (Dasari et al., 2023). Jing et al. separate dataset, architecture, and learning objective, and investigate semantic and temporal supervision (Jing et al., 2023). Their results should not be collapsed into a universal ranking of human, robot, and object-centric data: the evaluated objectives and deployment conditions differ. CortexBench examines broad transfer and adaptation (Majumdar et al.,

2023); the Data-Centric Revisit further investigates object-centric data and masked pretraining (Wen et al., 2025). Together, they motivate recording provenance and evaluating adaptation rather than treating data volume as a sufficient explanation of how well an encoder transfers.

M.7 What is measured: statistics, localization, and physical structure

Feature statistics and visual capabilities. Early work by Hsu et al. explicitly correlates representation measurements with closed-loop behavior-cloning performance (Hsu et al., 2022); we treat it as a workshop precursor. Burns et al. investigate visual capabilities as proxies for robust manipulation, with segmentation emerging as a more informative predictor than several conventional image-recognition measures in their study (Burns et al., 2023). Theia instead examines the distribution of spatial-token norms, reporting positive associations between histogram entropy and control performance, with distilled and non-distilled cohorts considered separately (Shang et al., 2024). Its spatial norm maps also expose task-irrelevant outlier tokens. Finding 7 finds the same positive direction for our norm-histogram entropy on two benchmarks and both training regimes, with nominal significance in one of three frozen cohorts, and a weaker, non-significant one once fine-tuned checkpoints are added. These measurements address different properties: semantic delineation, feature magnitude diversity, and spatial allocation should not be treated as interchangeable scores of a single underlying representation quality.

Task relevance and the readout interface. MCR measures manipulation centricity by comparing binarized Grad-CAM regions with end-effector and task-object masks, using average Jaccard overlap; its observed relationship with success motivates dynamics-aligned robot pretraining (Jiang et al., 2025a). VER analyzes patch norms and expert routing, showing how task-conditioned feature selection can suppress background outliers and change which regions are emphasized (Wang et al., 2026c). Such visualizations diagnose allocation, but a high patch norm is not itself an attention weight or proof of causal importance.

AFA makes aggregation the intervention. It compares readouts across 14 pretrained representations and relates out-of-domain success to attention mass on task-relevant regions and attention-map entropy (Tsagkas et al., 2025). Its learned pooling improves robustness without requiring encoder fine-tuning. Notably, lower attention entropy can indicate useful concentration, whereas Theia's higher norm-histogram entropy describes magnitude diversity (Shang et al., 2024). There is no contradiction: the random variable, spatial normalization, and evaluation target differ. Patch Policy supplies complementary evidence for retaining dense pretrained tokens in control (Zhou et al., 2026a). These works make aggregation an independent experimental axis; they do not imply that every dense interface outperforms every learned pooling mechanism.

Object structure and recoverability. Object-Centric Representations Improve Policy Generalization compares object-centric, dense, and global interfaces under lighting, texture, and distractor changes in simulated and real manipulation (Chapin et al., 2025). HODOR constructs task-oriented object/part hierarchies and studies the invariances inherited by downstream policies (Qian et al., 2024). We discuss HODOR as a structural representation method, rather than treating it as a general encoder-proxy benchmark. Stable Control Representations similarly broadens the representation families of interest through diffusion-model features that preserve fine-grained visuospatial information (Gupta et al., 2024).

SPA motivates spatially grounded pretraining (Zhu et al., 2025). Capturing Visual Environment Structure Correlates with Control Performance (CVESC) directly probes how well encoders support decoding geometry, object structure, and physical attributes, and relates decoding quality to control (Dong et al., 2026). This is a direct precedent for our state and object probes. However, a robot-state probe and an object-position probe each test a specified subset of physical information; neither certifies complete scene understanding.

Control-related latent geometry. Control-oriented Clustering studies the geometry of features learned through behavior cloning (Qi et al., 2025). For continuous control, its classes depend on relative-pose structure, and its diagnostics include within-class variability relative to class separation and the arrangement of class means. The work also tests a regularizer based on this structure. This is distinct from fitting an unlabelled feature distribution to a Gaussian or sphere. A compact cluster

is useful only relative to what should be distinguished: eliminating action-relevant variation would be undesirable even if the resulting geometry looked regular.

M.8 Temporal information, world models, and VLA internals

The Temporal Trap studies whether representations distinguish task progression, and reports that standard temporal augmentations do not fully resolve entanglement in its evaluated settings (Tsagkas et al., 2026). Its explicit time-based baseline is evidence about those settings, not a recommendation that elapsed time will generalize under arbitrary delays, resets, or speed changes. The paper's current version focuses on temporal entanglement; AFA is the companion study split from its earlier version (Tsagkas et al., 2025). They should not be counted as wholly independent replications of the same empirical claim.

Reconstruction or Semantics compares six encoder-defined latent spaces for robotic world models under a fixed training protocol (Nilaksh et al., 2026). It separates visual fidelity from planning/policy utility and latent quality, including action recoverability and task-success separability. Its patch-token inverse-dynamics decoder predicts an action chunk from a sequence of latent frames. Our two-frame target, one terminal command displacement, differs, so the losses cannot be numerically compared without matching the target and normalization. It is an adjacent world-model diagnostic study, not a direct policy-encoder leaderboard.

Decoding Task Progress from VLA Representations probes the residual stream of $\pi_{0.5}$, finding linearly readable progress information even in its pretrained PaliGemma backbone (Bhardwaj et al., 2026). The study also reports that readable progress does not enable meaningful steering of the policy. This distinction matters: decodability, policy use, and causal control are separate claims. It also explains why results on contextual VLA activations should not be conflated with probes on isolated frozen visual towers.

M.9 A question-driven comparison and the remaining gap

The literature does not lack explanations for representation transfer. It provides complementary explanations whose variables, interventions, and success criteria differ. Table 43 organizes those explanations by what is actually measured. Attention localization, object decomposition, state recoverability, action recoverability, and task progression can each be informative without inducing the same ranking.

Table 43: Representation-analysis precedents grouped by their central question. F: controlled factor study; D: explicit diagnosis/probing; M: a new method; combined labels indicate both. The descriptions state each study's emphasis, not an exhaustive list of its experiments or a shared numerical ranking.

Study	Question and measurement	Evidence beyond a success ranking	Implication for HIVE-Bench
Parisi et al. (Parisi et al., 2022)	Which pretraining recipe and feature hierarchy transfer?	F: objectives, augmentation, and hierarchy.	Control the extraction interface.
Hu et al. (Hu et al., 2023)	Does encoder quality depend on the learner?	F: 14 models, RL/BC/visual-reward learning, 21 tasks.	A shared head defines one comparison regime.
Dasari et al. (Dasari et al., 2023)	Which pretraining distribution transfers?	F: dataset comparisons and simulation/real ranking consistency.	Record data composition; qualify simulation evidence.
Jing et al. (Jing et al., 2023)	Which data, architecture, and objective matter?	F: semantic and temporal supervision comparisons.	Separate sources of inductive bias.
CortexBench / VC-1 (Majumdar et al., 2023)	How broad is transfer across tasks and adaptation?	F: heterogeneous embodied evaluation and adaptation.	Multi-regime evaluation has clear precedents.
Burns et al. (Burns et al., 2023)	Which visual abilities predict robust control?	D: visual proxies including segmentation.	Test proxy validity on the intended outcome.
Theia (Shang et al., 2024)	Do spatial feature statistics explain transfer?	M+D: token-norm entropy and outlier maps.	Re-test statistics across cohorts.
MCR (Jiang et al., 2025a)	Does the representation localize manipulation regions?	M+D: Grad-CAM/mask overlap and dynamics-aligned learning.	Distinguish task relevance from global diversity.

Study	Question and measurement	Evidence beyond a success ranking	Implication for HIVE-Bench
VER (Wang et al., 2026c)	Which experts and patches matter during a task?	M+D: routing, norm maps, and feature analysis.	Readout and task stage affect allocation.
AFA (Tsagkas et al., 2025)	Does aggregation remove irrelevant information?	M+D: pooling interventions, attention mass and entropy, OOD tests.	Pooling is an independent control variable.
Data-Centric Revisit (Wen et al., 2025)	How do data and object-centric learning affect transfer?	M+D: object-centric data and masked pretraining.	Data curation and feature structure interact.
Object-centric generalization (Chapin et al., 2025)	Which representation structure survives visual shifts?	F+D: global/dense/object-centric comparisons with perturbations.	Separate ID utility from OOD robustness.
SPA (Zhu et al., 2025)	Does spatially grounded pretraining improve embodied transfer?	M+F: spatial learning and broad embodied comparisons.	Geometry is a tested bias, not a universal ordering.
CVESC (Dong et al., 2026)	Can latent physical structure be decoded?	D: environment-state probes related to control.	Direct precedent for world recoverability.
Control-oriented Clustering (Qi et al., 2025)	Does latent geometry organize around control?	M+D: action/relative-pose classes and clustering regularization.	Label-conditioned structure differs from marginal shape.
Temporal Trap (Tsagkas et al., 2026)	Is task progression disentangled?	D: temporal representation measures and temporal baselines.	Static quality need not imply phase information.
Patch Policy (Zhou et al., 2026a)	Can dense tokens support an efficient policy?	M+F: dense-token policy interface.	Prior patch-level control; motivates matched readout ablations.
Reconstruction or Semantics (Nilaksh et al., 2026)	Which latent spaces support robotic world models?	D: fidelity, planning, action recovery, success separation.	Similar probes require matched targets to compare.
VLA progress probing (Bhardwaj et al., 2026)	Where is progress readable in a policy?	D: residual-stream probes and runtime monitoring.	Information availability is not causal use.
HIVE-Bench (this paper)	Which token properties track egocentric manipulation success?	D+F: four diagnostic families; per-task direction, combination under held-out selection, and paired adaptation.	Readout probes track success on every cohort; token statistics are weaker and fade under fine-tuning.

HIVE-Bench connects encoder family, scale, extraction layer, visual interface, diagnostic measurements, and closed-loop outcomes under one documented learning setup: native patch tokens, two robot-centric benchmarks, frozen and fine-tuned encoders, and four diagnostic families. For vision encoders whose policies read images without proprioception, readout probes of robot state and commanded motion track success on every frozen cohort, more closely than token statistics, while the state probe misranks the three Qwen3-VL-derived VLMs (Section H). Sweeps over viewpoint, fixed model capacity, and systematic visual shifts extend this design naturally. Xie et al.'s decomposition of generalization into 11 variation factors across 19 tasks is a precedent for such sweeps (Xie et al., 2024). Thoughts and Lessons on Using Visual Foundation Models for Manipulation, which relates edge and keypoint preservation to success, offers further context for interface and training choices in manipulation policies (Chen et al., 2025a).

N Evidence and Definitions for the Coverage Matrix

Table 42 describes reported experimental coverage, rather than ranking scientific quality. A cross means that the defined criterion is not established by the reviewed experiments, not that a method cannot support it. A check can arise from a subset of experiments; it does not imply matched tasks,

equal policy capacity, or a complete encoder-by-protocol grid. We use the same rule for HIVE-Bench, whose measured coverage is reported per encoder in Figure 9 and per analysis in Appendix E.

Table 44: Complete coverage of representation evaluation. Works are ordered by first arXiv release (Year), not conference year. Ego: onboard manipulation, including mixed views; Bi: dual-arm; Patch: spatial readout, including learned pooling; F/FT: frozen and policy-fine-tuned encoders; S/M: single-/multi-task policies; Stat: feature/attention statistics; State: physical-state/pose decoding; Act: action recovery from visual transitions; Corr: metric–success analysis. ✓: reported; ×: not established; †: qualified coverage (Appendix N). Checks denote study-level coverage, not every encoder/protocol combination. Coverage includes simulation and real-robot experiments where reported.

Study		Setting / input			Training		Diagnostics			
Work	**Year**	**Ego**	**Bi**	**Patch**	**F/FT**	**S/M**	**Stat**	**State**	**Act**	**Corr**
Parisi	2022	×	×	×	×	×	×	×	×	×
R3M	2022	×	×	×	×	×	×	×	×	×
VC-1	2023	✓	×	✓	✓	×	×	×	×	×
Hu	2023	×	×	×	×	×	×	✓	×	✓
Factor World	2023	✓	×	×	×	✓†	×	×	×	×
Vi-PRoM	2023	✓	×	×	×	×	×	×	×	×
D4R	2023	×	×	×	×	×	×	×	×	×
Burns	2023	✓	✓	✓†	×	×	×	×	×	✓
SCR	2024	✓	×	✓	×	×	×	×	×	×
Theia	2024	✓	×	✓	✓	×	✓	×	×	✓
COC	2024	×	×	×	×	×	✓	×	×	×
SPA	2024	✓	✓	✓	×	✓	×	✓	×	✓
MCR	2024	×	×	×	×	×	×	×	×	✓
HODOR	2024	✓	×	✓†	×	×	×	×	×	×
Temporal Trap	2025	×	×	×	×	✓	✓	×	×	✓
DCR	2025	×	×	✓	×	×	×	×	×	✓
OCR	2025	✓	×	✓	×	×	×	×	×	×
VER	2025	✓	×	✓	×	✓†	✓	×	×	✓
AFA	2025	×	×	✓	×	×	✓	×	×	✓
CVESC	2026	×	×	×	✓	×	×	✓	×	✓
RoS	2026	×	×	✓†	×	×	×	×	✓	×
Patch Policy	2026	✓	×	✓	×	✓	×	×	×	×
VLA Progress	2026	✓	×	×	×	×	×	×	×	×
HIVE-Bench	2026	✓	✓	✓	✓†	✓	✓	✓	✓†	✓

Observation and representation criteria. Ego includes head, body, or wrist cameras used for robot manipulation, even alongside external cameras. Egocentric human pretraining and navigation-only evaluation do not count. Bi requires a reported dual-arm manipulation setup; a dexterous single hand is not a second arm. Patch covers spatial encoder features evaluated through a local or token-based readout, including convolutional compression and learned attention pooling. It does not require raw patches to remain separate until the final action prediction. Spatial attention maps used only for analysis do not suffice. A dagger identifies an auxiliary experiment, a conversion into object tokens, or a world-model rather than direct-policy pathway through which the visual features reach the actions.

Analysis and training criteria. The added diagnostic columns distinguish what is measured from whether it is related to success. Stat includes norm distributions (Theia), attention statistics (AFA), latent class geometry (COC), temporal feature similarity (Temporal Trap), and feature/routing statistics (VER). State requires physical-state or pose decoding, as in Hu, SPA, and CVESC; object segmentation or task-progress decoding alone does not qualify. Act requires decoding actions from visual transitions, as in Reconstruction or Semantics and HIVE-Bench; action-conditioned prediction, a single-frame action-loss baseline, or action supervision during pretraining alone does not qualify. HIVE-Bench covers all three through its token statistics, its state and object probes, and its IDM probe. Its Act entry carries a dagger because the IDM probe decodes one terminal command displacement rather than an action sequence, and its F/FT entry because seven of the twenty encoders are fine-tuned.

Corr requires an explicit relationship between a representation measurement and control success across models or variants. A reported trend qualifies; a Pearson coefficient is not mandatory. Success-rate ablations, simulation-to-real success comparisons, and a progress probe without a representation-metric/control relationship are different evidence. F/FT requires both frozen transfer and policy-related fine-tuning of pretrained visual features to be investigated; training an encoder from scratch does not qualify. Updating only a pooling head or router, or adapting on general pretraining data before freezing, does not qualify either. S/M records coverage of both task-specific policies and a policy shared across tasks, not a matched causal test of task sharing. Multi-task data alone and evaluating one shared policy separately on each task do not count as both regimes, because only one of the two policy types is then trained.

Short names in Tables 42 and 44. DCR: Data-Centric Revisit; CVESC: Capturing Visual Environment Structure Correlates with Control Performance; D4R: Data4Robotics; COC: Control-oriented Clustering; OCR: Object-Centric Representations; RoS: Reconstruction or Semantics; SCR: Stable Control Representations. Citations give the full titles.

Table 45: Evidence and qualifications for the compact main-text comparison; section and appendix names refer to the cited works, not to this paper.

Study	Basis and qualification
VC-1 (Majumdar et al., 2023)	CortexBench includes onboard MobilePick and spatial features for mobile tasks, alongside global-feature manipulation. Frozen and adapted representations are studied. Different task suites are not one shared multi-task policy.
Burns (Burns et al., 2023)	Section 5.4 uses ALOHA with two wrist and two external cameras. Patch coverage is qualified: the real ACT pathway differs from the global-feature simulation study. Segmentation is related to robustness. The ACT setup alone does not explicitly establish a frozen/FT encoder comparison.
SPA (Zhu et al., 2025)	Single-task and language-conditioned multi-task evaluation are explicit. LIBERO includes onboard observations; RLBench/RVT-2 uses spatial features. Real experiments include dual-arm cloth folding. Camera-pose decoding is related to transfer; evaluated encoders are frozen.
DCR (Wen et al., 2025)	Manipulation uses frozen patch features with learned attention pooling; objectness is compared with success. Egocentric navigation and egocentric pretraining do not establish onboard manipulation evaluation.
CVESC (Dong et al., 2026)	State-proxy correlations are reported for frozen and fine-tuned encoders. Section 4 describes a shared text-conditioned diffusion policy; switching to an MLP in Appendix A.2 does not by itself establish single-task training. Spatial probes do not turn the global policy readout into a patch-policy study.
Patch Policy (Zhou et al., 2026a)	Explicit patch/global readout comparisons; wrist-camera real tasks and mixed-view LIBERO. Task-specific settings and shared LIBERO policies are included. A fine-tuned OpenVLA baseline does not establish frozen/FT comparisons for the studied visual encoders.
Parisi (Parisi et al., 2022)	Frozen global features, hierarchy, and pretraining ablations. The egocentric setting is ImageNav; it does not establish egocentric manipulation. A from-scratch baseline is not fine-tuning.
Hu (Hu et al., 2023)	Frozen global features evaluated with BC, RL, and visual-reward learning. State-probe errors are related to BC performance. Different learners are not different task-sharing regimes.
D4R (Dasari et al., 2023)	Downstream encoders are fine-tuned with the policy. Comparisons isolate pretraining data and examine simulation/real rankings. The latter compares policy outcomes, not an independent representation metric with success.
Vi-PRoM (Jing et al., 2023)	External-view simulation and wrist RGB-D real manipulation; frozen global features. Dataset, architecture, and objective studies are distinct from proxy correlations and downstream multi-task policies.
Theia (Shang et al., 2024)	Spatial-token compression and token-norm entropy analysis; Spot drawer opening uses a body-mounted camera. Real experiments include frozen and fine-tuned variants. CortexBench tasks use separate policies.
MCR (Jiang et al., 2025a)	Manipulation-region overlap is explicitly related to success. Downstream control uses frozen global features. Multi-task robot pretraining does not establish shared multi-task downstream evaluation.
COC (Qi et al., 2025)	Global latent geometry is studied during end-to-end policy learning, including class-conditioned collapse and regularization. This is not a cross-encoder scalar-proxy correlation study, nor a frozen-versus-fine-tuned sweep.

Study	Basis and qualification
OCR (Chapin et al., 2025)	Global, dense, and object-slot representations are compared under visual shifts, with LIBERO/real mixed cameras and shared task-conditioned policies. These interventions do not establish a separate scalar-proxy correlation or an explicit single-task/shared-policy comparison.
Temporal Trap (Tsagkas et al., 2026)	Task-progression loss is related to success. Frozen global PVRs support single-task policies and a four-task causal-transformer experiment (Section V-C). No onboard or dual-arm setting is established.
VER (Wang et al., 2026c)	Spatial experts, feature statistics, and routing are analyzed. LIBERO supplies mixed-view evidence. S/M is qualified: task-specific suites and a four-task LIBERO protocol use different heads. Router adaptation is not full visual-backbone fine-tuning.
AFA (Tsagkas et al., 2025)	Learned pooling reads spatial features; task-region attention mass and entropy are related to OOD success. The PVR stays frozen; training the pooling module is not encoder fine-tuning.
RoS (Nilaksh et al., 2026)	Patch coverage is qualified: dense latent spaces condition a robotic world model rather than a newly trained direct action policy. Action recovery and success separability are diagnostics, but are not a reported cross-encoder metric–policy-success correlation.
HODOR (Qian et al., 2024)	Table and wrist cameras support real manipulation. Patch coverage is qualified: dense DINOv2 features are pooled into task-selected object/part tokens. The study tests structure and invariance rather than scalar-proxy correlations.
SCR (Gupta et al., 2024)	Dense diffusion features support control; OVMM includes mobile manipulation. SCR-ft adapts on general robot/human datasets with the generative objective (Section 4.5), then supplies frozen features. This differs from policy fine-tuning.
Factor World (Xie et al., 2024)	Onboard real-robot evaluation complements external-view simulation. S/M is qualified: separate simulation policies and a shared real RT-1 policy are different systems. Its PVR ablation is global; generalization-gap decomposition is not a representation-proxy correlation.
VLA Progress (Bhardwaj et al., 2026)	Front/side/wrist observations and shared VLA policies. Probing residual-stream activations across pretrained and robot-trained checkpoints is not an isolated patch-encoder policy comparison or a frozen/FT policy-training ablation. Progress readability alone does not establish metric–success correlation.
R3M (Nair et al., 2022)	Frozen global scene embeddings, task-specific policies, and pretraining ablations. Human egocentric pretraining does not establish an onboard downstream benchmark; from-scratch controls do not establish encoder fine-tuning.