JEPA-DiT: From-Scratch Predictive World-Model Pretraining for Manipulation

Anonymous authors
Paper under double-blind review

Abstract

World-action models—transformers that jointly denoise future video and actions—represent the world almost exclusively in the latent of a reconstruction VAE inherited from a video-generation prior. That is a liability for control: a reconstruction objective must preserve every pixel-level nuisance a decoder has to repaint, whereas a policy needs only the scene's controllable structure. A policy-free diagnostic makes the cost concrete—under domain randomisation the reconstruction latent moves nearly twice as far from its clean distribution as a predictive (JEPA) latent ($\rho = 2.05$ vs. 1.16), while a cross-task control shows both register a change of *task* equally: invariance, not collapse. The field keeps the VAE anyway, because a JEPA latent has no generative prior to build a world model on. We present **JEPA-DiT**, to our knowledge the first world-action model whose diffusion transformer is pretrained *from random initialisation* in a JEPA latent, carried unchanged from egocentric pretraining through teleoperation mid-training to downstream manipulation, and **delta-JEPA**, which makes the two representations complementary rather than competing by having the predictive stream carry the temporal *difference* from the present observation. Training from scratch proves to be not the price of a predictive latent but the reason it works: in a width-matched ablation the from-scratch stream develops a language pathway at $25\times$ the floor uniform attention would give while a warm-started run reaches $2.1\times$, barely above chance, and passes that deficit downstream. Across all 50 RoboTwin 2.0 tasks with randomisation *held out of training*, JEPA-DiT reaches 84.5% clean and 59.5% out of distribution—the best OOD score and best clean/OOD average among world-action models on this protocol. Each representation alone falls short of the pair: with a mature prior but no predictive stream the model scores 1.9% under the shift, with a predictive stream but no prior 11.9%.

1 Introduction

World-action models—transformers that jointly denoise future video latents and action chunks in one backbone, so that predicting the world and choosing an action share parameters—are among the strongest systems on the training distribution in bimanual manipulation, and the weakest off it: on RoboTwin 2.0 they take two of the top three clean scores yet none of the top three under domain randomisation (LingBot Team (Robbyant), 2025; Yuan et al., 2025; Guo et al., 2026). Almost universally they operate in one visual state space—the latent of a *reconstruction* VAE inherited from a generative video prior (Alibaba Wan Team, 2024). That latent is convenient because it is where large pretrained priors already exist. It is not obviously the right state space for *control*.

A reconstruction objective must preserve every pixel-level factor of variation—colour, lighting, background, render style—because a decoder has to reproduce them. An embodied policy needs none of it. A *predictive* (joint-embedding, JEPA) objective (Assran et al., 2025) is under no such obligation, and a policy-free diagnostic makes the difference concrete: under domain randomisation the reconstruction latent shifts nearly twice as far from its clean distribution as the predictive latent does.

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.

Yet the field does not use it, and the obstacle is a missing ingredient rather than a wrong choice. A reconstruction latent arrives with a mature generative prior to initialise a world model; a predictive latent has none, and invariance does not substitute for one. A predictive-latent world model trained without such a prior scores six times higher under randomisation than a reconstruction-latent model, and is still much the weaker system (§4.4). One representation has the quality but lacks the pretraining, the other the reverse. **This paper supplies the missing ingredient.** We pretrain *JEPA-DiT* from random initialisation on egocentric video and carry that same predictive state space through teleoperation mid-training to bimanual manipulation—which lets us ask a question the field has not been able to pose: once a predictive latent is given the pretraining it lacked, does its invariance translate into better generalisation under shift?

A predictive latent need not *replace* the reconstruction latent to earn its place; framing the two as rivals gives up something free, since reconstruction preserves exactly what prediction discards. The obstacle to using both is *redundancy*—two encoders reading the same pixels and both representing absolute content. We therefore propose *delta-JEPA*: the predictive stream represents the temporal *difference* from the present observation. Subtracting the anchor removes the static, appearance-dominated component—the part a reconstruction latent already encodes well and the part randomisation attacks hardest—leaving the scene's change, which is what an action must bring about. The two video streams are kept mutually invisible and the action expert reads both.

Language enters differently too. The world-action models on this benchmark inject text as a frozen T5 embedding through cross-attention—a static key/value the other streams may attend to sparsely, with no gradient pressure forcing otherwise. JEPA-DiT instead makes text a first-class stream inside a single joint attention, so it co-evolves with vision and action at every layer. Joint multimodal attention is not itself new—video generators interleave text and visual tokens in shared attention (§2)—so what we change is narrower: the frozen T5 side input becomes a trained stream.

That pathway is where from-scratch pretraining pays off, and it is not a cost we accept but the reason the approach works. In a matched ablation of two ego pretraining runs, the from-scratch predictive stream builds a depth-organised language pathway at $25\times$ the floor uniform attention would give while the warm-started run sits at $2.1\times$, because the prior supplies text-stream weights but not the predictive stream's projections: a newborn stream must interface with an already-converged one, and never learns to. One rule follows. Language must live in some visual stream, and a stream acquires it only by growing alongside text. Where a pretrained reconstruction stream is present it already carries language and the predictive stream is freed to specialise on change; where none exists, train from scratch so both grow together (§4.3).

The closed-loop result lands where the invariance measurement said it would. Across all 50 RoboTwin 2.0 tasks with the randomised configuration held out of training, JEPA-DiT reaches 84.5% clean and 59.5% out of distribution—the best OOD score and the best clean/OOD average among world-action models on this protocol, while the strongest prior world-action model scores higher on clean scenes yet keeps barely half of it under the shift (Wang et al., 2026). A property measured on the frozen encoders, before a single policy gradient step, predicts how much of a policy survives a change of scene.

Our contributions are threefold. *JEPA-DiT* is, to our knowledge, the first world-action model whose diffusion transformer is pretrained in a predictive (JEPA) latent from random initialisation, carried unchanged through a three-stage pipeline from egocentric video to manipulation; the world-action $\times$ predictive-latent quadrant was otherwise empty, since every neighbour represents the world in a reconstruction latent alone and prior robot uses of JEPA employ it as a frozen encoder rather than a space to pretrain a generative model in. *Delta-JEPA* then makes the two representations complementary rather than competing—the predictive stream carries the temporal difference from the present observation, subtracting the static, appearance-dominated component the other already holds—and text becomes a first-class joint-attention stream in place of the frozen T5 cross-attention the benchmark's world-action models use. Finally, we show that from-scratch pretraining is the enabling condition rather than a cost: a policy-free diagnostic with a cross-task control separates invariance from collapse, a matched initialisation ablation shows the predictive stream's language pathway forms only when trained from scratch and that the deficit is inherited downstream, and the closed-loop result follows on all 50 RoboTwin 2.0 tasks.

2 Related Work

Two axes organise the literature: the *task architecture* (vision–language–action policy vs. world-action model that jointly denoises future video and actions) and the *video representation* (a reconstruction VAE vs. a self-supervised predictive latent). Appendix A surveys each quadrant; here we state where JEPA-DiT sits.

The predictive-representation idea has appeared on the VLA track—VPP (Hu et al., 2024) conditions an action head on predictive features and argues predictive $\succ$ VAE on CALVIN (Mees et al., 2022), and JEPA-style embeddings have been used as VLA encoders (Miao et al., 2026)—but there the predictive network is a *frozen encoder* feeding a policy, not the latent space of a generative world model. The distinction matters for what can be concluded: VPP establishes that predictive *features* are a better input to an action head, a statement about conditioning signals. Our question is different—whether a predictive latent can serve as the *state space a world model is trained in*, which requires pretraining a generative model there and is exactly what the absence of a JEPA-space prior has prevented. On the world-action track, in turn, the representation is almost always a reconstruction VAE (LingBot Team (Robbyant), 2025; Yuan et al., 2025; Zhu et al., 2024; Guo et al., 2026; NVIDIA, 2025a; Zhang et al., 2025; Luo et al., 2025; BeingBeyond, 2026). The one world-action model we are aware of that departs from a VAE is LDA-1B (Lyu et al., 2026), which models unified tasks in a *DINO* image-SSL latent; it is the closest point to our work.

JEPA-DiT differs on three counts that no single prior work combines. **(i)** Representation: our video state space is a *video-native predictive* latent (V-JEPA tubelets), not a per-frame image-SSL latent (DINO) and not a reconstruction latent. **(ii)** Training: we pretrain a generative diffusion transformer *in* that predictive space *from random initialisation*, rather than reusing a frozen predictive encoder or fine-tuning a VAE-space video prior, and the same frozen latent then carries through teleoperation mid-training to post-training without being exchanged for another space. **(iii)** Composition: rather than treating predictive and reconstruction latents as rivals, *delta*-JEPA makes them complementary—the predictive stream carries change, the reconstruction stream absolute content—which to our knowledge no prior world-action model does; the reconstruction-latent models cited above use that representation alone, and LDA-1B substitutes DINO for it.

3 Method

Figure 1 gives the architecture; this section makes it precise.

3.1 Setup and Notation

JEPA-DiT is a world-action model whose video state space is a *predictive* (JEPA) latent: per-stream experts are coupled by a single masked joint attention and denoise future video latents and action chunks together. The formulation follows this family of models (LingBot Team (Robbyant), 2025); ours differs in pretraining the predictive stream from scratch rather than inheriting a VAE-space video prior. The deployed model also keeps a reconstruction stream, because under delta-JEPA (§3.3) the predictive stream carries only differences and something must carry absolute content; the two are *mutually invisible* and the action expert reads both.

A training sample is a temporally aligned *window*: multi-view RGB observations encoded by a *frozen* encoder into F latent frames. The first is the observed present; the remaining $G = F - 1$ each receive N control steps, for an action target of $G \cdot N$. The window fixes that budget, so more latent frames means fewer raw actions per frame—a relation §3.2 turns on. Appendix G lists the symbols.

Training minimises a flow-matching objective with one term per denoised stream,

$$\mathcal{L} \;=\; \mathbb{E}_t\big[\,\lambda_r\,\mathcal{L}_{\text{recon}} \;+\; \lambda_p\,\mathcal{L}_{\text{pred}} \;+\; \lambda_a\,\mathcal{L}_{\text{action}}\,\big], \tag{1}$$

where $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{pred}}$ are the reconstruction and predictive video streams (the backbone of Figure 1 carries only the latter), with text entering as a first-class stream rather than as cross-attention input (Appendix F). Each video term is measured in whatever units its frozen encoder defines, so neither is *comparable* across representations; only $\mathcal{L}_{\text{action}}$, in the shared normalised action space, is. At deployment the model is a chunked open-loop controller: observe, denoise the window, commit

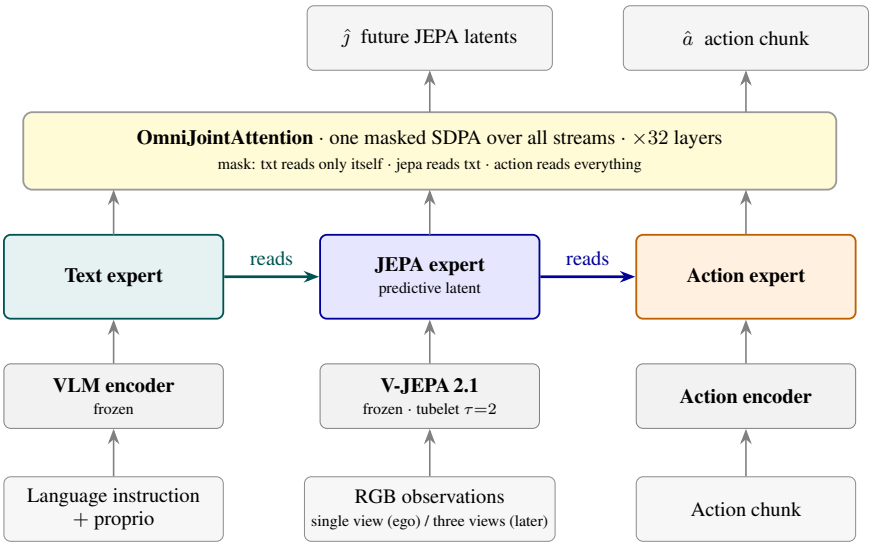


Figure 1: **JEPA-DiT.** The world is represented in a frozen *predictive* (V-JEPA) latent rather than a reconstruction latent. Per-stream experts are coupled by a *single* masked self-attention at every layer—not three transformers, since the coupling lives entirely in the mask—and the model denoises future video latents and action chunks jointly. Visibility is one-way: text reads only itself, the video stream reads text, and the action expert reads everything, so text is a first-class stream rather than a cross-attention side input. §3.3 adds a reconstruction stream to this backbone under delta-JEPA (Appendix O).

the actions of the first K latent frames, execute, re-observe. A rollout therefore never supplies more than the present observation as visual evidence.

3.2 Two Video Representations

The frozen encoder fixes the state space, and the two candidates differ in what their objective *obliges* them to retain.

A causal video VAE of the kind video diffusion models are built on (Alibaba Wan Team, 2024) is trained with a decoder to reconstruct pixels. That objective is sufficient: a latent from which pixels can be reconstructed necessarily retains everything that varies in the image—texture, lighting, distractors—whether or not it is relevant to control. Reconstruction does not merely permit this content, it requires it.

A predictive encoder—we use the V-JEPA 2.1 release of V-JEPA 2 (Assran et al., 2025)—instead predicts the representation of masked spatio-temporal regions from visible context, with no pixel decoder. Because the objective is stated *in representation space*, nothing pushes it to retain detail that is unpredictable or irrelevant to prediction; nuisance appearance may be discarded rather than encoded. That is the mechanism we expect to make it the more invariant of the two, and §3.4 turns the expectation into a measurement. It tokenises with *tubelets*, embedding every τ consecutive frames into one temporal position; the camera views are embedded as a montage, carrying multi-view structure without architectural change.

The two compress time at different rates, so the choice also changes how finely the action stream is supervised: the predictive encoder yields more latent frames, hence more action groups covering fewer raw steps each (Appendix B). This confounds any naive comparison—an experiment that swaps the representation also changes the supervision granularity.

3.3 Delta-JEPA: Making the Two Representations Complementary

Posing the two representations as alternatives discards information for no reason: reconstruction preserves exactly what prediction drops. The obstacle to using both is *redundancy*—two encoders

reading the same pixels and both representing absolute content produce overlapping descriptions the joint attention has no reason to keep distinct. *Delta*-JEPA removes the overlap by changing what the predictive stream represents. Writing $\phi(\cdot)$ for the frozen predictive encoder and g_0 for the anchor group fixed by the present observation, the stream carries

$$\tilde{j}_g \;=\; \phi(g) \;-\; \phi(g_0), \qquad g = 1, \ldots, G, \tag{2}$$

rather than $\phi(g)$.

Redundancy could be attacked in other ways—an orthogonality penalty between the two streams, a contrastive term, a shared bottleneck—but all of them add an objective and hyper-parameters, and none of them says *which* content each stream should give up; they leave the division of labour to optimisation. Differencing fixes it in the input instead: the operation itself adds no parameters and no loss term, and it removes a specific, identifiable quantity—whatever is constant over the window—rather than whatever a penalty happens to find.

Whatever is constant across the window—textures, lighting, clutter, static objects—appears in both terms and cancels: exactly the content a reconstruction latent represents well and, by §3.4, exactly the content randomisation perturbs. The objection is immediate: this hands the reconstruction stream the least stable content in the scene. But absolute content is not optional—a controller must know where things are—so the question is not whether to represent it but what else is available when it degrades. What remains after the subtraction is the scene's change, which is what a manipulation action is defined by.

Complementarity is then enforced rather than hoped for. The difference is taken against the anchor, so the predictive stream cannot recover absolute content alone; nor can it acquire it through attention, since the mask makes the two video streams *mutually invisible*—each attends only itself and text—while the action expert attends everything (Appendix P). On the deployed model the cross-stream shares are exactly zero at every layer, and the action expert draws 0.39 from the reconstruction stream and 0.30 from the predictive one (Appendix T). The two descriptions meet at the policy and nowhere else, which also makes the pair one design rather than two detachable parts: equation 2 presupposes that something carries the state it subtracts.

When to use it. A stream acquires language by growing alongside the text stream; one attached to an already-converged model does not (§4.3). Where a pretrained reconstruction stream exists it already holds both the absolute description and the language pathway, so a predictive stream added afterwards need not reacquire either—delta-JEPA is the right way to add it. Where none exists, whatever visual stream is present must carry both, which it can do only if trained from random initialisation alongside text. One rule, two branches: with a reconstruction pretrain, add the predictive stream as a difference; without one, train from scratch so both grow together.

The anchor must be reproducible at rollout. equation 2 is defined against $\phi(g_0)$, so it is well posed only if a rollout can construct the anchor training used. For a tubelet encoder this is not automatic: with $\tau = 2$ the anchor is the embedding of the first frame *pair*, while a rollout has one observation. Tail padding puts a future frame in that pair, so the deployed model is conditioned on something it never saw in training and the loss cannot see the error; **head** padding makes the pair the present frame duplicated, which a rollout reproduces bit-for-bit. We adopt head padding and verify it rather than assume it (Appendix N).

3.4 Measuring Representation Invariance

Policy success conflates the representation with the policy trained on it, so we also compare the encoders directly, with no policy at all.

RoboTwin 2.0 (Li et al., 2025) renders each task clean and randomised while holding task semantics and demonstrated motion fixed; a representation useful for control should be largely unaffected. Let $\mathcal{C}$ and $\mathcal{A}$ be sets of encoded windows from the two renders of one task and $\bar{d}_{\cos}$ the mean pairwise cosine distance. The *separability ratio* is

$$\rho \;=\; \frac{\bar{d}_{\cos}(\mathcal{C}, \mathcal{A})}{\frac{1}{2}\left[\bar{d}_{\cos}(\mathcal{C}) + \bar{d}_{\cos}(\mathcal{A})\right]}, \tag{3}$$

the clean-to-randomised distance in units of the scatter the representation already shows within a split. Being a ratio of cosine distances it is dimensionless, hence comparable across encoders whose widths differ by nearly two orders of magnitude. $\rho \approx 1$ means a randomised window is no farther from the clean ones than they are from each other; $\rho \gg 1$ means the randomised renders occupy a region the policy never visited.

One control is essential. A low ρ is not by itself evidence of useful invariance, since a representation encoding nothing would score the same. We therefore always report a **cross-task control**: the same quantity between two entirely different manipulation tasks—taken from a separate teleoperation corpus, since RoboTwin's randomisation is what the first measurement varies—a change any useful representation must register. An encoder low on clean-vs-randomised but comparable to the baseline on cross-task is invariant to nuisance while remaining discriminative; one low on both is collapsed. Appendix M records two further controls attributing equation 3 to the encoder alone.

We compute equation 3 on the encoder's *absolute* features, whereas the deployed stream consumes differences. This is conservative rather than mismatched: the difference is taken against a reference from the same clip, so any appearance shift shared across the window cancels exactly, and what the policy reads can only be *more* invariant than the value we report. 32

4 Experiments

Our experiments answer three questions. Is a predictive video latent in fact less sensitive to appearance than a reconstruction latent, and can that be established before any policy is trained (§4.2)? Does pretraining from random initialisation matter, or would a mature prior serve as well (§4.3)? And does the representation-level difference translate into closed-loop generalisation under a distribution shift held out of training (§4.4)? We answer them on all 50 RoboTwin 2.0 tasks, against published world-action models that share our training set and protocol.

4.1 Setup

Model. The world is represented in a frozen V-JEPA 2.1 ViT-G predictive latent (three camera views as one montage), consumed as a temporal difference against the anchor. Four per-stream experts—text, predictive video, action, and a reconstruction video stream holding the absolute content delta-JEPA subtracts—are coupled by a single masked joint attention; the two video streams are mutually invisible and both are read by the action expert. Widths, depths and the mask are in Appendix S.

Stages. (i) *Ego pretraining* on egocentric video from random initialisation—no weights inherited from any pretrained diffusion model. (ii) *Teleop mid-training* from that checkpoint on AgiBot teleoperation, adding the action stream. (iii) *Post-training* on RoboTwin 2.0: a full fine-tune of all four streams with the video encoders frozen.

Single-representation arms. The reconstruction-only arm is a published world-action model sharing our training set and protocol; it differs in backbone, parameter count and pretraining corpus, so it sizes the effect rather than attributing it. The predictive-only arm is ours—a JEPA-DiT from random initialisation with no reconstruction stream and no inherited prior—evaluated on the identical harness.

Protocol. All 50 RoboTwin 2.0 tasks under `demo_clean` and `demo_randomized`, the latter varying background, lighting, clutter, tabletop height, object pose and language paraphrasing. Each cell is 20 rollouts at inference chunk horizon $K = 2$; success is the simulator's completion signal and we report the unweighted mean over tasks. *No randomised data is used in training*—models are adapted on clean demonstrations and the randomised configuration is held out, making it an out-of-distribution (OOD) evaluation. This matches the official protocol (RoboTwin Team, 2026), making published numbers comparable in kind, with two differences: the board uses 100 trials per task against our 20, so our intervals are wider, and we compare against its *co-train* rows—one policy over all 50 tasks, as ours is—not its per-task fine-tuning rows. Training runs on 160 NVIDIA B300 GPUs.

Table 1: **Representation invariance, before any policy.** Clean→randomised separability ρ (equation 3); higher means the representation registers the change more strongly. Both encoders run on identical decoded windows. The cross-task control is what licenses the interpretation: the predictive latent separates *tasks* as well as the reconstruction latent does, so its low clean-vs-randomised value reflects invariance to nuisance rather than a collapsed representation.

Encoder	clean vs. randomised	cross-task control
VAE (reconstruction)	2.05	1.18
JEPA (predictive)	**1.16**	1.17

4.2 Is the predictive latent less appearance-sensitive?

Before any policy exists we compare the two frozen encoders directly, using the separability ratio ρ of §3.4 on the *same* decoded windows (Table 1). Under randomisation the reconstruction latent separates clean from randomised renders at $\rho = 2.05$, the predictive latent at 1.16; all five tasks agree in sign (Appendix M plots them individually).

A low ρ alone would admit a damaging reading, since a latent that distinguishes nothing would also score near one. Repeating the measurement between *two entirely different tasks*—a change any useful representation must register—the encoders are nearly identical (1.18 vs. 1.17): the predictive latent is invariant, not collapsed. The pair then exposes an anomaly on the *other* side. The reconstruction latent responds *more strongly* to a change of background and material (2.05) than to a change of the entire task (1.18); its geometry is organised by appearance rather than task content—precisely the property a policy must overcome, and, as §4.4 shows, fails to.

Because the splits are independently collected, that distance mixes appearance with motion. We remove the confound by construction, re-rendering one trajectory twice under different textures with motion bit-identical ($\max |\Delta q| = 0.0$). Appearance is then the only variable and the ordering survives causally: the reconstruction encoder's similarity between the two encodings of the same motion falls to 0.29–0.36, the predictive encoding stays at 0.47–0.53 (Appendix J).

4.3 Does pretraining from scratch matter?

Warm-starting leaves the predictive stream unable to use language at all, and the deficit is inherited downstream. Raw attention shares depend on segment lengths, so we quote multiples of the uniform-attention floor; all values are means over 8 validation windows and all 32 layers.

A controlled ablation of initialisation. Two ego pretraining runs are identical in per-stream width, data, bucketing, T_g, stage, mask and text handling, differing in initialisation—one random, one warm-started—and in learning rate, the one uncontrolled factor. The from-scratch stream reaches $25.1\times$ the floor and organises it by depth, rising from 0.016 at layer 0 to a peak of 0.503; the warm-started stream reaches $2.1\times$, barely above chance, its deep layers falling to 0.009 (Table 2, Figure 5). The $12\times$ gap reproduces at an earlier checkpoint of the same two runs ($25.5\times$ against the same floor).

The checkpoint shows why. It supplies text-stream weights but not the predictive stream's projections, which are therefore randomly initialised regardless: a newborn stream is asked to interface with an already-converged text representation, and does not learn to. Started together, both co-adapt. The lesson is not that pretrained weights harm, but that a prior covering only some streams can be worse than no prior at all.

The property is inherited, and it changes the policy. Two mid-training runs of the same width differing only in which ego checkpoint they descend from preserve the ordering—$21.2\times$ the floor against $0.90\times$, indistinguishable from uniform—so whether the predictive stream uses language is fixed during pretraining, not recovered later. The policy follows: between those runs action→jepa rises from 0.641 to 0.847 while action→action falls from 0.296 to 0.129.

Table 2: **Attention the predictive stream pays to text**, as a multiple of the uniform-attention floor, which is computed per stage from that stage's sequence layout. The ego rows are a controlled ablation—same width, data, bucketing, T_g, stage and mask, differing in initialisation (and in learning rate, the one uncontrolled factor). The mid-training rows are two runs of the same width differing only in which ego checkpoint they descend from.

Stage	Initialisation	jepa→txt	× floor
Ego pretraining	From scratch	**0.212**	**25.1**
Ego pretraining	Warm-started	0.018	2.1
Mid-training	From-scratch lineage	**0.211**	**21.2**
Mid-training	Warm-started lineage	0.009	0.9

Where language lives with two video streams. The deployed model reads only 0.077 on jepa→txt—itself a warm-start case, its predictive stream introduced onto an already-trained text-and-reconstruction model. The language did not vanish but moved: the *reconstruction* stream reads 0.248, and it is the stream trained from random initialisation alongside text in the first pretraining phase. A wrong-instruction intervention agrees, moving that stream's prediction by $\Delta\cos = 0.17$–0.26 against ≤ 0.022 for the predictive stream (Appendix T).

The account closes on itself. Language must live in some visual stream, and a stream acquires it only by growing alongside text. With one video stream the predictive stream is the only candidate, so warm-starting it is fatal. With two, the reconstruction stream occupies that role and the predictive stream is freed to specialise on change—the role equation 2 assigns it. What from-scratch pretraining buys the deployed model is therefore not a language pathway in the predictive stream but the fact that *some* stream grew into one—impossible under an inherited prior, where text is a frozen side input rather than a stream at all.

4.4 Does it generalise under held-out shift?

On all 50 RoboTwin 2.0 tasks, JEPA-DiT reaches 84.5% clean and 59.5% out of distribution—the best OOD score and best clean/OOD average among world-action models on this protocol (Table 3). No task is at zero in either configuration, and the residual gap concentrates in long-horizon, high-precision sequences: sustained coordination, not perception of the altered scene, is what remains unsolved (Appendix D).

Two arms bracket the trade-off we opened with. Fast-WAM, which carries a mature generative prior in a reconstruction latent, is strong on the training distribution and keeps almost nothing off it ($77.8 \rightarrow 1.9$). We do not read that as a property of reconstruction latents in general: the same representation spans 1.9 to 48.7 across published systems, so how much of the shift survives is evidently not fixed by the representation alone. A JEPA-DiT from random initialisation *without* any prior is the mirror case: it scores six times higher under the shift (11.9 vs. 1.9), so the representation's invariance shows through, but it is 14.5 points down on clean scenes. Lacking both a prior and a reconstruction stream, it bounds the joint effect rather than separating the two.

What the diagnostic predicts here is *retention*, not absolute competence: the predictive-only model keeps 19% of its clean score against the reconstruction-only model's 2% while being the weaker model outright. Pairing the two exceeds both arms on both axes at once—they do not average. That is what delta-JEPA buys: the predictive stream is not a better reconstruction stream but an account of a different aspect of the scene, and equation 2 keeps it from duplicating what the other covers.

Against the field. The strongest world-action model on this protocol, OpenWAM (Wang et al., 2026), is the sharpest comparison: it exceeds us on clean scenes (89.4 vs. 84.5) yet retains less under the shift (48.7 vs. 59.5). Since clean and randomised trade off against one another, the clean column alone is the one a model can buy by fitting the training distribution more closely; the randomised column and the average are not. Repeated evaluation of one checkpoint sharpens this: independent runs span 5.9 points on clean but 0.4 on randomised (Appendix E), so the column we lead on is also the measurably stable one. Across the reconstruction-latent systems retention runs from 2% to 54%, and our 70% is the only entry above that band. Two board entries do exceed our randomised score—both vision–language–action policies that never denoise future video—but each

Table 3: **RoboTwin 2.0, all 50 tasks.** Success rate (%); OOD is the held-out `demo_randomized` configuration and Δ the clean→OOD gap. All rows adapt on the same 2,500 clean demonstrations with randomisation held out, though pretraining corpora differ across rows; ours uses 20 rollouts per task per configuration, leaderboard rows 100. Leaderboard entries are the *co-train* rows—one policy over all 50 tasks—so they differ from per-task fine-tuned numbers a system may report elsewhere. OpenWAM is not a board entry; its numbers are those its authors report under the same held-out protocol. Best per column in bold. Clean and OOD trade off against one another, so we report both and their average. Other architecture classes in Appendix H.

Model	World repr.	Clean ↑	OOD ↑	Avg ↑	Δ
JEPA-DiT (ours)	Pred. + recon.	84.5	**59.5**	**72.0**	**−25.0**
Single-representation arms, same held-out protocol					
Fast-WAM (Yuan et al., 2025)	Recon. only	77.8	1.9	39.9	−75.9
JEPA-DiT, ours, no recon. stream	Pred. only	63.3	11.9	37.6	−51.4
Other world-action models					
AHA-WAM (Cai et al., 2026)	Recon.	64.3	3.2	33.8	−61.1
X-WAM (Guo et al., 2026)	RGB-D	70.0	25.8	47.9	−44.2
4D-WAM	RGB-D	81.5	41.8	61.7	−39.7
OpenWAM (Wang et al., 2026)	Recon.	**89.4**	48.7	69.0	−40.7

gives up clean competence to get there, and neither matches our clean/OOD average (Appendix H). Predicting the world is what buys the average; giving up prediction buys robustness at a price.

Is it just capacity? Carrying two video streams makes our model larger than a single-stream one, so size is the obvious alternative explanation. The matched ablation of Appendix E speaks to it directly: adding the predictive stream at a fixed budget for everything else gains +3.4 points on clean and +6.0 under randomisation, and only the randomised gain exceeds evaluation noise—whereas extra capacity would show up first on the training distribution.

5 Conclusion

A predictive video latent is the more appearance-invariant state space for control; what kept the field from building world models in one was never its quality but the absence of a generative prior, which we supply. *JEPA-DiT* is, to our knowledge, the first world-action model pretrained from random initialisation in a JEPA latent, and *delta-JEPA* makes the predictive and reconstruction representations complementary rather than rival: carrying the temporal difference from the present observation subtracts exactly the static, appearance-dominated content the other already holds.

Across all 50 RoboTwin 2.0 tasks with randomisation held out of training, JEPA-DiT reaches 84.5% clean and 59.5% out of distribution—the best OOD score and the best clean/OOD average among world-action models on this protocol, and far above either representation used alone.

The deeper claim is not about JEPA. A stream learns to read another only while that other is still forming; bolted onto a converged one, the coupling never appears. That is why training from scratch is the enabling condition rather than a cost, and why a prior covering only some streams can be worse than no prior at all. We expect it to bind wherever a modality or expert is added to an already-trained multimodal backbone: the question is not how much of a checkpoint can be reused, but which *couplings* reuse silently forfeits. Limitations and a falsifiable prediction are in Appendices W and X.

Use of AI assistance. An AI assistant polished writing, helped locate related work, and audited the manuscript against its sources. Every number and setting here is a measurement, not a recollection: each was re-derived from the artefact that produced it—evaluation records, the leaderboard's data file, probe outputs, launch scripts, checkpoint tensors. Widths come from weights, not configuration files; baselines from the board's data file, not self-reported figures; and every reference was resolved against its origin. All claims and the experimental design are the authors' own.

REFERENCES

AgiBot World Contributors. AgiBot world: A large-scale embodied foundation dataset for real-world robotic manipulation. https://agibot-world.com/, 2025.

Alibaba Wan Team. Wan2.2: Open and advanced large-scale video generative models. https://github.com/Wan-Video/Wan2.2, 2024.

Mahmoud Assran, Adrien Bardes, David Fan, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning, 2025.

BeingBeyond. Being-H0.7: A latent world-action model from egocentric videos. https://arxiv.org/abs/2605.00078, 2026.

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A vision-language-action flow model for general robot control, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023.

Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, et al. RoboCat: A self-improving foundation agent for robotic manipulation, 2023.

Anthony Brohan et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023a.

Anthony Brohan et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023b.

Jake Bruce, Michael Dennis, Ashley Edwards, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

Jisong Cai, Long Ling, Shiwei Chu, et al. AHA-WAM: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing. https://arxiv.org/abs/2606.09811, 2026.

Chi-Lam Cheang et al. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, et al. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, et al. RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot, 2023.

Galaxea AI. Galaxea open world: An open-world mobile manipulation dataset. https://huggingface.co/galaxea-ai, 2025.

GigaAI Research. GigaWorld-1: A roadmap to build world models for robot policy evaluation. https://arxiv.org/abs/2607.02642, 2026.

GigaBrain Team. GigaBrain-0.7: Scaling embodied foundation models to emergent capabilities with a three-system architecture. https://arxiv.org/abs/2608.15875, 2026.

Jun Guo, Qiwei Li, Peiyan Li, et al. Unified 4D world action modeling from video priors with asynchronous denoising. https://arxiv.org/abs/2604.26694, 2026.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

HPC-AI Tech. Open-Sora: Democratizing efficient video production for all. https://github.com/hpcaitech/Open-Sora, 2024.

Anthony Hu, Lloyd Russell, Hudson Yeo, et al. GAIA-1: A generative world model for autonomous driving, 2023.

Yucheng Hu et al. Video prediction policy: A generalist robot policy with predictive visual representations. https://github.com/roboterax/video-prediction-policy, 2024.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, et al. Prismatic VLMs: Investigating the design space of visually-conditioned language models. In *International Conference on Machine Learning (ICML)*, 2024.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. DART: Noise injection for robust imitation learning. In *Conference on Robot Learning (CoRL)*, 2017.

Tianxing Li et al. RoboTwin 2.0: A bimanual robot manipulation benchmark with strong domain randomization. https://github.com/RoboTwin-Platform/RoboTwin, 2025.

LingBot Team (Robbyant). LingBot-VA: A video–action world model for bimanual manipulation. https://github.com/Robbyant/lingbot-va, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, et al. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Hao Luo, Yicheng Feng, Wanpeng Zhang, et al. Being-H0: Vision-language-action pretraining from large-scale human videos. https://arxiv.org/abs/2507.15597, 2025.

Jiangran Lyu et al. LDA-1B: A latent dynamics-action foundation model with universal embodied data ingestion. https://pku-epic.github.io/LDA/, 2026.

David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

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

Shangchen Miao, Ningya Feng, Jialong Wu, et al. JEPA-VLA: Video predictive embedding is needed for VLA models. https://arxiv.org/abs/2602.11832, 2026.

Yao Mu et al. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. https://robotwin-platform.github.io/.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, et al. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

NVIDIA. Cosmos-Policy: Post-training cosmos for robot policy with joint action, world, and value objectives. https://nvidia-cosmos.github.io/cosmos-cookbook/recipes/post_training/predict2/cosmos_policy/post_training.html, 2025a.

NVIDIA. Cosmos world foundation model platform for physical AI. https://www.nvidia.com/en-us/ai/cosmos/, 2025b.

Octo Model Team, Dibya Ghosh, Homer Walke, et al. Octo: An open-source generalist robot policy, 2024.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models, 2024.

OpenAI. Video generation models as world simulators. https://openai.com/research/video-generation-models-as-world-simulators, 2024.

RoboTwin Team. RoboTwin 2.0 leaderboard. https://robotwin-platform.github.io/leaderboard, 2026. Accessed 2026-09-20; board updated 2026-09-01.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Yu Shang, Zhuohang Li, Yiding Ma, et al. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. https://arxiv.org/abs/2602.08971, 2026.

Stone Tao, Fanbo Xiang, Arth Shukla, et al. ManiSkill3: Gpu parallelized robotics simulation and rendering for generalizable embodied AI. In *Robotics: Science and Systems (RSS)*, 2024.

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

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, et al. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Tianyuan Yuan et al. Fast-WAM: World-action model with train-time video co-training and inference-time action-only decoding. https://github.com/yuantianyuan01/FastWAM, 2025.

Wenyao Zhang et al. DreamVLA: World knowledge forecasting meets diffusion-based vision-language-action model. https://zhangwenyao1.github.io/DreamVLA/, 2025.

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

Chuning Zhu et al. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. https://weirdlabuw.github.io/uwm/, 2024.

APPENDIX

A EXTENDED RELATED WORK

VISION–LANGUAGE–ACTION POLICIES

Behaviour-cloning policies mapping perception and language to actions have scaled rapidly, from RT-1 (Brohan et al., 2023a) and RT-2 (Brohan et al., 2023b), which co-trains an internet-scale vision–language model with robot trajectories, to open-weight VLAs such as OpenVLA (Kim et al., 2024) on a Llama backbone with Prismatic vision (Karamcheti et al., 2024), Octo (Octo Model Team et al., 2024), and π_0 (Black et al., 2024), which pairs a vision–language backbone with a flow-matching action expert. Two ideas from this line carry over to ours: *action chunking* (Zhao et al., 2023), predicting the next H actions in one shot, and diffusion action heads (Chi et al., 2023), which capture multi-modal action distributions. What this line does *not* do is model future world dynamics—assuming implicitly that perception alone suffices to extract the control signal, which a growing body of evidence suggests is data-inefficient at scale (Hu et al., 2024; Yuan et al., 2025).

VIDEO AND WORLD MODELS FOR ROBOT LEARNING

A second line argues that imitation alone is data-hungry and that *predicting the future* is a strong auxiliary signal. Classical world models train recurrent latent dynamics in pixel (Ha & Schmidhuber, 2018) or latent space (Hafner et al., 2020; 2023) for sample-efficient model-based RL. Following diffusion-based image and video generators (Ho et al., 2020; Rombach et al., 2022; Blattmann et al., 2023; HPC-AI Tech, 2024; OpenAI, 2024), robot-relevant video world models followed—GAIA-1 (Hu et al., 2023) for driving, Genie (Bruce et al., 2024) for interactive 2D worlds, Wan-2.2 (Alibaba Wan Team, 2024) as an open-weight video diffusion foundation, and the Cosmos family (NVIDIA, 2025b) trained for physical realism with explicit downstream policy use (NVIDIA, 2025a)—alongside work pushing toward 4D and navigable worlds, and a unified benchmark for scoring them (Shang et al., 2026).

V-JEPA-2 (Assran et al., 2025) and follow-ups such as JEPA-VLA (Miao et al., 2026) instead pretrain in a self-supervised *latent* space, predicting future representations rather than pixels; this discards control-irrelevant appearance at the cost of a directly inspectable pixel rollout—a trade we argue is favourable for control. The generative models above supply the reconstruction latents that current world-action models build on; JEPA-DiT departs from them by placing the world model in a predictive latent (§2).

ACTION-CONDITIONED AND UNIFIED WORLD–ACTION MODELS

Closest to our work is a recent wave of *action-conditioned* world-action models that jointly denoise actions and future video latents in one transformer. A first group unifies several task heads on a shared backbone: UWM (Zhu et al., 2024) supports policy, forward-dynamics, inverse-dynamics and video-prediction modes; DreamVLA (Zhang et al., 2025) routes predicted dynamic, depth and semantic world knowledge into a diffusion action head; and Cosmos-Policy (NVIDIA, 2025a) post-trains a video foundation model into a policy/world/value tri-objective transformer. A second group targets deployment on RoboTwin 2.0: LingBot-VA (LingBot Team (Robbyant), 2025) interleaves video and action tokens autoregressively behind a streaming VAE wrapper, and Fast-WAM (Yuan et al., 2025) argues that *training-time* video co-training is the source of the gain and that explicit future-video generation can be skipped at inference—it is the baseline our main comparison uses. A third group extends the recipe along other axes: X-WAM (Guo et al., 2026) to multi-view RGB-D with asynchronous noise schedules, and concurrent industrial efforts— GigaWorld-1 (GigaAI Research, 2026) and Being-H0/0.7 (Luo et al., 2025; BeingBeyond, 2026)—toward data engines, multi-embodiment alignment and interactive control.

Two members of this space bear directly on our claims. VPP (Hu et al., 2024) fine-tunes a video diffusion model and conditions an action head on its predictive features, validating a "predictive-representation $\succ$ VAE-latent" hypothesis on CALVIN (Mees et al., 2022); and LDA-1B (Lyu et al., 2026) models unified tasks in a *DINO* latent, the one world-action model we know of that departs from a VAE. Otherwise what unites this diverse space is the visual state space: these models rep-

resent the world in a reconstruction latent. Whether a video-native *predictive* latent, given its own from-scratch pretraining, is a better state space for a world-action model is orthogonal to their contributions and, to our knowledge, unaddressed (§2).

Most share a second convention: language enters as a frozen T5 embedding read through cross-attention, a static side input to the video–action backbone. This holds for the systems deployed on RoboTwin 2.0 that our comparison concerns. It is not universal—Cosmos' transfusion (NVIDIA, 2025b;a) interleaves text and visual tokens in a shared attention—and mixing modalities inside one joint attention stack is in any case established in video generation, so we do not claim joint multi-modal attention as novel. What differs here is narrower: the text stream is trained in place rather than supplied frozen, which is the condition under which the language pathway of §4.3 can form at all (Appendix P).

Closed-Loop Chunked Execution and Train–Rollout Consistency

World-action models are deployed as chunked open-loop controllers, so the gap between how they are trained and how they are rolled out matters. This gap has classical roots in receding-horizon and model-predictive control (Mayne et al., 2000); in imitation learning, DAgger (Ross et al., 2011) and DART (Laskey et al., 2017) address compounding error by modifying the training data distribution to include states reached under the learner's own policy, while action chunking with temporal ensembling (Zhao et al., 2023; Chi et al., 2023) tames it in the prediction target by amortising decisions over multiple steps; RoboCat (Bousmalis et al., 2023) and Octo (Octo Model Team et al., 2024) vary chunk length at inference. Our concern is a different, representation-level instance of the same principle: a tubelet video encoder aggregates several frames into one token, so the token a rollout can construct from the single present observation must match the one seen in training. We make this train–rollout consistency exact by construction (Appendix N); to our knowledge it has not been flagged for predictive-latent world-action models, as it does not arise for the per-frame VAE latents used elsewhere.

Bimanual Manipulation Benchmarks and Pretraining Data

RoboTwin (Mu et al., 2024) and RoboTwin 2.0 (Li et al., 2025) are SAPIEN-based simulators (Xiang et al., 2020) built on the Aloha-AgileX bimanual platform, whose 14-DoF protocol follows ALOHA (Zhao et al., 2023). They provide 50 dual-arm tasks, each with a `demo_clean` and a `demo_randomized` configuration varying clutter, lighting, background, tabletop height, language paraphrasing and object pose. LIBERO (Liu et al., 2023), RoboCasa (Nasiriany et al., 2024), ManiSkill3 (Tao et al., 2024), CALVIN (Mees et al., 2022) and the GR-1 / GR-2 benchmarks (Cheang et al., 2024) complement it with single-arm, multi-room and language-chaining scenarios, while large-scale real-robot corpora—Open X-Embodiment (Open X-Embodiment Collaboration, 2024), DROID (Khazatsky et al., 2024), RH20T (Fang et al., 2023), AgiBot World (AgiBot World Contributors, 2025) and Galaxea Open World (Galaxea AI, 2025)— supply the video-action mixture world-action backbones are pretrained on.

We evaluate on RoboTwin 2.0 for two reasons: its bimanual protocol stresses temporal coordination and gripper-contact reasoning that single-arm benchmarks do not, and its public leaderboard (RoboTwin Team, 2026) fixes a clean-only training set with a held-out randomised evaluation—a directly comparable measurement of exactly the distribution shift this paper is about.

B Temporal Compression and Action Granularity

§3.2 states the consequence; here is the arithmetic.

Temporally such an encoder compresses in a $1+ck$ pattern: the first latent frame encodes one raw frame and each later one a block of c, giving

$$F_{\text{VAE}} \;=\; 1 + \left\lceil \frac{P-1}{c} \right\rceil , \tag{4}$$

where c is the encoder's temporal factor. At the teleoperation mid-training setting ($R = 33$, $s = 4$, $P = 9$, $c = 4$) this gives $F_{\text{VAE}} = 3$; post-training uses an encoder with $c = 8$ on a shorter window (Appendix S).

Predictive latents. The alternative is V-JEPA 2.1 (Assran et al., 2025), a joint-embedding predictive encoder trained to predict the representation of masked spatio-temporal regions from visible context, with no pixel decoder. Because the objective is stated *in representation space*, there is no gradient pressure to retain detail that is unpredictable or irrelevant to prediction: nuisance appearance may be discarded rather than encoded. This is the mechanism by which we expect a predictive latent to be the more invariant of the two, and §3.4 turns that expectation into a measurement.

Temporally, a JEPA encoder tokenises with *tubelets*: it consumes a fixed number P' of frames and embeds every non-overlapping group of τ consecutive frames into one temporal position, giving

$$F_{\text{JEPA}} = P'/\tau. \tag{5}$$

With $\tau = 2$ and the $P' = 10$ frame budget of the pretrained checkpoint, $F_{\text{JEPA}} = 5$. Our $P = 9$ sampled frames must therefore be padded by one to reach P'; Appendix N shows the padding *side* is not a free choice. Spatially, the three camera views are embedded as a montage into a single sequence of patch tokens, so the multi-view structure is carried without architectural change.

Consequence: the granularity of action supervision. The two representations do not merely place the world model in different feature spaces; they change how finely the action stream is supervised. We work the argument on the teleoperation mid-training window. The action budget is the same at every stage—a chunk of 32 control steps—so what differs between stages is only the observation window and the resulting group count; post-training values are recorded in Appendix S. Because the action mask of Appendix P assigns actions to $G = F - 1$ groups and the budget $G \cdot N$ is fixed by the window, equation 4 and equation 5 imply different N and different group counts:

Representation	P	F	$G = F - 1$	N	action groups
Reconstruction (c=4)	9	3	2	16	2×16
Predictive (τ=2)	9	5	4	8	4×8

Both cover the same 32 raw actions of the same window, and in both the first group is conditioning rather than prediction. But the VAE path asks the model to emit one action group covering 16 raw steps, whereas the JEPA path asks for four groups of 8. The predictive representation thus yields a *finer* temporal factorisation of the same supervision at no change to the data. We emphasise that this is a consequence of the encoder's temporal compression rate, not an independent design knob, and that it confounds any naive comparison of the two paths: an experiment that changes the representation also changes N and G.

C Ablations: what is and is not isolated

Two design choices are ablated under matched conditions. The first is initialisation: the two ego pretraining runs of §4.3 share width, data, bucketing, T_g, stage and mask and differ in whether the model starts from random weights—and in learning rate, the one uncontrolled factor—with the consequence traced from representation through to how the policy allocates attention.

The second is the predictive stream itself. A matched pair differing *only* in whether that stream is present—its loss weight set to zero, which removes it from the forward pass, so the action expert never sees a predictive token—gives $+3.4$ points on clean and $+6.0$ under randomisation at the matched step. The gain is roughly twice as large off-distribution as on it, the direction the representation diagnostic predicts, and only the randomised gain exceeds the evaluation noise we measure by re-running the same checkpoint. Appendix E gives the protocol and the noise estimate.

Four axes remain unablated, each requiring a matched pair of full pretraining runs. The first is the one a reader will want most: differencing against the anchor versus an *absolute* predictive stream under the same mutually-blind mask, which would separate the contribution of equation 2 from that of simply carrying a second video stream. Our case for differencing is accordingly mechanistic and diagnostic—the mask and the anchor make the two descriptions non-overlapping by construction, and ρ locates the content that cancels—rather than a closed-loop A/B. The remaining three are a controlled swap against a reconstruction-only model pretrained identically, the cross-stream visibility pattern, and the action↔video mask that governs whether predicted video reaches the action stream (Appendix P). We name them because each is a live alternative a reader might expect us to have tested.

D WHERE THE REMAINING GAP LIVES

The two ends of the 50-task distribution are informative, and Table 10 places them side by side. At one end, randomisation is simply free: six tasks score at least as well under randomisation as on clean scenes, among them both `shake_bottle` variants and `grab_roller` at 100% in both configurations. These are tasks whose solution depends on the geometry of a single large object and its motion, precisely the content a predictive latent retains and a reconstruction latent buries under appearance.

At the other end, retention falls below a third on `stamp_seal`, `hanging_mug`, `handover_block` and `place_dual_shoes`. These are not the tasks with the worst clean scores—`stamp_seal` is at 90% clean and `blocks_ranking_rgb` at 95%—so the failure is not simply difficulty carrying over. What they share is length and the need for a precise relative pose sustained across a multi-stage sequence: a rotate-then-hook, a two-arm handover, a stamp aligned to a mark. In such episodes a single appearance-induced error early on cannot be recovered, and the residual sensitivity of the representation is amplified by the horizon rather than averaged out. Improving robustness further is therefore likely to be as much a question of closed-loop recovery as of representation.

E SINGLE-VARIABLE ABLATION OF THE PREDICTIVE STREAM

§4.4 bounds the effect of the predictive stream with two single-representation arms that each differ from our model in more than one respect. This section reports the strict version: a matched pair differing *only* in whether the predictive stream is present.

Protocol. Both runs share initialisation, learning rate, batch size, mask, pixel pipeline, augmentation and every other setting verbatim; the ablation sets λ_p in equation 1 to zero, which removes the stream from the forward pass entirely—its tokens never enter the joint attention, so the action expert's keys and values contain no predictive content. This is the exact counterpart of zeroing the reconstruction weight, which removes that stream. Both are evaluated on all 50 tasks at 20 rollouts per cell under the held-out protocol of §4.1.

Result. At the matched step (Table 4) the predictive stream is worth $+3.4$ points on clean scenes and $+6.0$ under randomisation—the gain is roughly twice as large off-distribution as on it, which is the direction the representation diagnostic predicts.

Scope. The pair is trained at a lower learning rate than the main result, so its absolute numbers are lower; the comparison is between the two arms, not against the system at its best.

Evaluation noise is asymmetric, which bears on how much of the gap to believe. Re-running the identical with-stream checkpoint gives clean 80.4 and 74.5 on two independent evaluations—a spread of 5.9 points—against randomised 41.3 and 40.9, a spread of 0.4. The clean column on RoboTwin is far noisier than the randomised one. Taken with that spread, the $+6.0$ randomised gain is outside the observed noise while the $+3.4$ clean gain is not, and we claim only the former.

Table 4: **Removing the predictive stream.** Success rate (%) on all 50 RoboTwin 2.0 tasks, matched step, both arms identical except for the presence of the predictive stream. The gain is concentrated off-distribution. Repeated evaluation of the same baseline checkpoint spans 5.9 points on clean and 0.4 on randomised, so only the randomised gain exceeds evaluation noise.

Configuration	Clean	Rand.	Avg
With predictive stream (delta-JEPA)	**80.4**	**41.3**	**60.9**
Without predictive stream	77.0	35.3	56.2
Difference	+3.4	**+6.0**	+4.7

F TOKEN LAYOUT AND OBJECTIVE

At each training step we sample a timestep t and form noisy targets for every denoised stream—R_{noisy} and P_{noisy} for the reconstruction and predictive video streams, A_{noisy} for the action stream. Each is concatenated with a *condition* tensor of identical shape, and the text embedding is supplied as a stream alongside them:

$$X_{\text{main}} \;=\; \text{concat}[R_{\text{noisy}},\; R_{\text{cond}},\; P_{\text{noisy}},\; P_{\text{cond}},\; A_{\text{noisy}},\; A_{\text{cond}}]\,.$$

After the block stack, the noisy slots produce the flow predictions that enter the corresponding terms of equation 1; the condition-stream outputs are discarded. Prediction targets differ by stream: the predictive video stream uses x-prediction in its own representation space, the remaining streams use velocity prediction. The three-stream backbone of Figure 1 is the same construction without the R slots.

G NOTATION

Table 5: Notation used in §3 and in Appendix B. F, K and N follow the conventions of prior work in this family (LingBot Team (Robbyant), 2025); R, s and P are introduced here to separate the raw-frame, sampled-frame and latent-frame scopes, which the two representations of §3.2 relate differently.

Symbol	Meaning
R	Raw observation frames per training window
s	Frame-sampling stride within the window
P	Sampled frames handed to the encoder, $P = \lceil R/s \rceil$
F	Latent frames produced by the encoder (written T_g for the predictive stream)
G	Action groups, $G = F - 1$ (see Appendix P)
N	Raw action steps per predicted latent frame, $G \cdot N = R - 1$
K	Chunk horizon: latent frames committed per inference chunk
R, P, A	Reconstruction, predictive and action stream slots (Appendix F)
$\lambda_r, \lambda_p, \lambda_a$	Loss weights on the reconstruction, predictive and action streams
ρ	Clean-to-randomised separability (§3.4)

H FULL LEADERBOARD CONTEXT

Clean and randomised success trade off against one another across checkpoints of a single run: continued training on clean demonstrations keeps improving the clean score while the randomised score falls. Clean alone is therefore a metric that overfitting rewards, which is why we report the randomised score and the average of the two.

Table 3 lists the world-action models our claim concerns. For completeness, the strongest entries from other architecture classes on the same board are below. Four of the five are vision–language–action policies that do not denoise future video at all, so they reach robustness by other means and are not counterexamples to a claim about visual state spaces. Two of them, GigaBrain-0.7 and OLA-Sem, exceed our randomised score; no world-action model on the board does, and neither exceeds our clean/OOD average.

I FULL PER-TASK RESULTS

J CONTROLLED RE-RENDER: RESIDUAL CONFOUND

The confound and how it is removed. The clean and randomised splits are independently collected trajectories, so a distance between them mixes appearance with motion. Matching episodes by action statistics leaves a residual rather than removing it (Appendix J), so we remove it by construction. We render one saved joint trajectory twice, changing only wall and table textures with

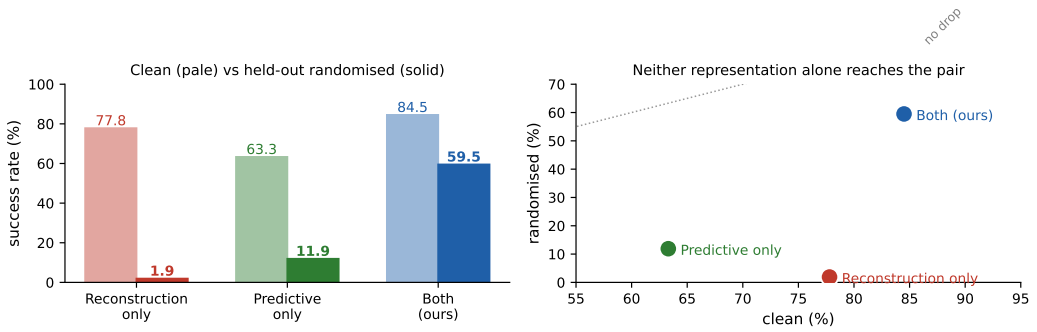


Figure 2: **Our two single-representation arms.** *Left*: clean (pale) and held-out randomised (solid) success for the three arms. *Right*: the same three points in the clean–randomised plane, the dotted line marking no drop under shift. A reconstruction-latent model with a mature prior is strong on the training distribution and keeps little off it; a predictive-only model from random initialisation scores six times higher under randomisation but is the weaker model outright; pairing the two exceeds both of these arms on both axes. Stronger reconstruction-latent systems exist (Table 3); the contrast here is between the two arms, not a claim about reconstruction latents in general.

Table 6: Strongest board entries from other architecture classes, success rate (%). Entries without a citation have no accompanying paper (RoboTwin Team, 2026).

Model	Class	Clean ↑	Rand. ↑	Avg ↑	Δ
4D-WAM	RGB-D WAM	81.5	41.8	61.7	−39.7
OLA-Sem	VLA	75.1	67.6	71.3	−7.5
GigaBrain-0.7 (GigaBrain Team, 2026)	VLA	66.8	67.9	67.3	+1.1
$\pi_{0.5}$	VLA	70.7	46.0	58.4	−24.7
OLA-Geo	VLA	83.0	33.6	58.3	−49.4

Table 7: Full per-task success (%) on all 50 RoboTwin 2.0 tasks for the main-results checkpoint (§4.4), 20 rollouts per task per configuration, sorted by randomised score. No task is at zero in either configuration; six tasks match or exceed their clean score under randomisation.

Task	Clean	Rand.	Task	Clean	Rand.
grab_roller	100	100	place_can_basket	75	65
place_empty_cup	95	100	place_shoe	90	65
shake_bottle	100	100	open_laptop	90	60
shake_bottle_horizontally	100	100	place_a2b_right	85	60
place_burger_fries	100	95	place_object_basket	75	55
place_container_plate	100	95	blocks_ranking_rgb	95	50
place_object_stand	90	95	place_fan	80	50
place_bread_basket	95	85	put_bottles_dustbin	80	50
press_stapler	90	85	beat_block_hammer	95	45
stack_blocks_two	100	85	handover_mic	100	45
place_bread_skillet	95	80	move_can_pot	75	45
place_object_scale	75	80	turn_switch	60	40
stack_bowls_two	90	80	lift_pot	85	35
click_alarmclock	100	75	pick_diverse_bottles	85	35
click_bell	100	75	put_object_cabinet	40	35
place_cans_plasticbox	95	75	rotate_qrcode	95	35
place_phone_stand	85	75	stack_blocks_three	85	35
adjust_bottle	95	70	place_mouse_pad	85	30
dump_bin_bigbin	100	70	blocks_ranking_size	75	25
pick_dual_bottles	95	70	move_stapler_pad	50	25
scan_object	75	70	open_microwave	25	20
stack_bowls_three	75	70	place_dual_shoes	70	20
move_pillbottle_pad	95	65	handover_block	55	10
move_playingcard_away	100	65	stamp_seal	90	10
place_a2b_left	95	65	hanging_mug	45	5

Overall: clean **84.5**, randomised **59.5**, average **72.0**

every randomisation switch disabled; over the evaluation window the two clips are bit-identical in joint angles, $\max|\Delta q| = 0.0$, and in end-effector pose. Appearance is then the only variable, which makes this the one place in the paper where a representation-level claim is established causally rather than correlationally. The ordering survives and sharpens: changing only the textures drops the reconstruction encoder's similarity between the two encodings of the same motion to 0.29–0.36—a third to a half of the representation rewritten by a change no controller should care about—while the predictive stream's predicted groups for the identical pair stay at 0.47–0.53. Both sides are measured on the same re-rendered pair, so the ordering is attributable to the representation and not to the data.

§4.2 gives the protocol and the result. This section records the quantity that motivated building the control rather than relying on episode matching.

The obvious cheaper route is to pair episodes across the clean and randomised splits by action statistics and compare those. It does not work: independently collected episodes never match exactly—the nearest neighbours we found still differ in action distance, and the closest pair differs even in length—so any ρ computed on such pairs remains part motion and part appearance, with no way to say how much of each. Re-rendering a single saved trajectory removes the residual by construction rather than bounding it, which is why we report that instead.

Table 8: Controlled re-render: the same joint trajectory rendered under two appearances ($\max|\Delta q| = 0.0$, verified elementwise), so all difference is attributable to texture. We report *cosine similarity* between the two encodings of the same clip (higher = more invariant to texture change): over the VAE's three latent frames, and over the predictive stream's four predicted non-anchor groups. Note the direction is opposite to ρ in Table 1: ρ is a distance-based ratio (higher = more sensitive) while this is a similarity (higher = more invariant).

Encoder	cos(clean, re-rendered)
VAE (reconstruction)	0.29–0.36
JEPA (predictive)	**0.47–0.53**

K Adaptation protocol and cost

Both video encoders are used *frozen*; only the world model is trained. Within that constraint the protocol has two settings, and which one applies is dictated by whether the training pipeline needs augmented pixels.

Full finetuning with online encoding. The main result (§4.4) updates all four experts, with no low-rank restriction. This is paired with *online* encoding: frames are decoded and passed through the frozen encoders inside the training loop. The pairing is forced rather than incidental. We apply colour jitter to the raw pixels, and a precomputed latent cache is by construction built from unjittered frames, so any pixel-space augmentation is invisible to a cached pipeline. Since the entire claim of this paper concerns robustness to appearance change, giving up pixel-space augmentation to save encoder compute would be self-defeating.

Low-rank adaptation with a cached encoder. Where encoder compute or data volume dominates—compute-constrained and exploratory runs—we instead adapt with low-rank updates and read precomputed per-window features from a memory-mapped cache, which removes the encoder's forward pass from the training loop entirely. Appendix L gives the adapter placement and the two correctness properties such a cache must satisfy, both of which fail silently.

L Feature-cache correctness

Adapter placement. In the low-rank setting of Appendix K we inject rank-r adapters into the attention projections of the video and action experts and merge them into the base weights when checkpointing, so the resulting checkpoint is structurally indistinguishable from a fully finetuned one and the evaluation stack requires no change. Modules whose input or output dimension is dictated by the embodiment—the action encoder and head, the proprioception encoder, and the video patch

embedding and head—are kept fully trainable, since a low-rank update around a randomly initialised projection is not a meaningful restriction.

Cache correctness. Appendix K notes that the precomputed per-window feature cache must satisfy two properties. Both failures are silent, which is why we state them.

The cache must be keyed by the *original* window position, so that a run restricting training to a subset of tasks reads the same rows as a run that does not. Keying by position within the filtered subset instead makes two runs disagree about which window a given index denotes, and nothing in the loss reveals it.

The cache's padding convention must match the one used at rollout (Appendix N). A cache built under the wrong convention encodes a different anchor than the deployed controller constructs, yet produces a training signal that is internally consistent and minimises equation 1 as usual. It fails only at rollout — precisely the class of error equation 6 is stated to exclude, reintroduced one layer down in the data pipeline.

M SEPARABILITY RATIO: PER-TASK VALUES AND ATTRIBUTION CONTROLS

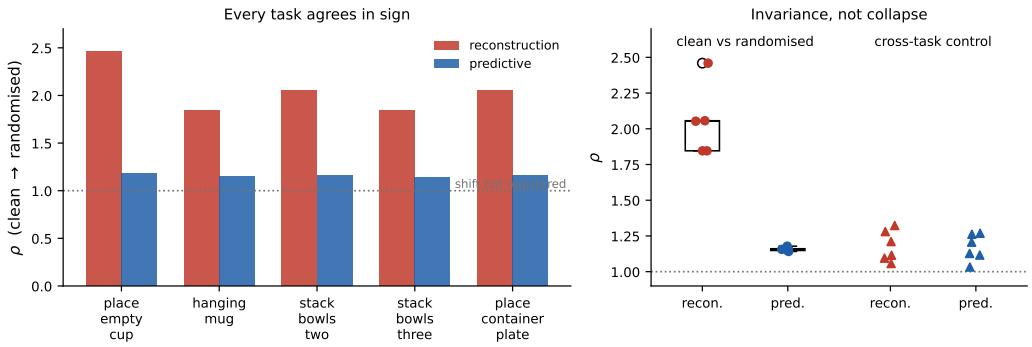


Figure 3: **Separability ratio ρ, per task.** *Left*: all five tasks agree in sign—the reconstruction latent registers the appearance change at 1.8 to 2.5 times its own within-split scatter, the predictive latent barely above it. The dotted line at $\rho = 1$ marks a representation that does not register the shift at all. *Right*: the same values as a distribution (circles) beside the cross-task control (triangles), the latter measured on held-out teleoperation tasks with the same frozen encoders. The control licenses the reading: the two encoders are indistinguishable when the *task* changes, so the predictive latent's low clean-vs-randomised value reflects invariance rather than collapse. The reconstruction latent also responds more strongly to a change of appearance than to a change of task.

§3.4 states the cross-task control, which is what makes a low ρ interpretable. The control is computed on four held-out AgiBot teleoperation tasks, over all six task pairs, with the same frozen encoders and the same estimator: per-pair ratios span 1.06–1.32 for the reconstruction latent and 1.03–1.27 for the predictive one, giving the means of 1.18 and 1.17 quoted in §4.2. It is deliberately a different corpus from the clean-vs-randomised measurement, whose whole variable is RoboTwin's randomisation; what transfers across the two is the encoder, which is the object under test. Note that the control establishes the two encoders register a task change *equally*, not that either registers it strongly—the ratios are well below the 2.05 the reconstruction latent reaches on appearance alone, which is the asymmetry §4.2 draws attention to.

Two further controls are needed for equation 3 to be attributable to the encoder rather than to the data pipeline.

First, both encoders must run on the *same* decoded windows, so that any difference is attributable to the encoder and not to which episodes or frames happened to be drawn. Second, neither may read a precomputed feature cache: caches in this pipeline are built for the clean split only, so a cached-versus-live discrepancy would masquerade as a clean-versus-randomised one — exactly the

quantity being measured. We satisfy both by encoding freshly, in the same process, from one shared set of decoded windows, for both encoders.

N THE ANCHOR CONTRACT

§3.3 states that delta-JEPA is well posed only if a rollout can reconstruct the anchor training used. We give the condition formally, derive the slot arithmetic for both padding sides, and report the verification.

The condition. At training time the encoder sees P sampled frames padded to P'; at rollout the controller has exactly one real observation and must build the same first temporal position. Writing $\mathcal{E}$ for the frozen encoder and $g_0(\cdot)$ for the anchor it produces, the requirement is

$$g_0\big(\mathcal{E}(\text{train window})\big) \;=\; g_0\big(\mathcal{E}(\text{rollout observation})\big) \quad \text{for the same present frame,} \tag{6}$$

i.e. the anchor must be a function of the present observation alone. If equation 6 fails, the deployed model is conditioned on a quantity it never saw in training, and the failure is invisible to the training loss.

Slot arithmetic. The condition above fixes what a rollout must reproduce; here we give the slot arithmetic for both padding sides. The encoder consumes P' frames and groups every τ consecutive frames into one temporal position, so with $\tau = 2$ the anchor is the embedding of slots $(1, 2)$.

Padding the sampled sequence at the **tail** leaves the sampled frames in their original positions, giving slot contents $[0, s, 2s, \dots]$. The anchor pair is then $(0, s)$: it contains a frame s steps in the *future* of the present observation, which no rollout can supply, so equation 6 is violated. Symmetrically, the final pair degenerates to the last sampled frame duplicated, wasting one temporal position on a group that carries no new information.

Padding at the **head** shifts the sampled frames by one, giving $[0, 0, s, 2s, \dots]$. The anchor pair is $(0, 0)$ — the present frame duplicated — which a rollout reproduces exactly by duplicating its single observation across the tubelet. Every subsequent pair then consists of strictly future frames, each carrying a full complement of N raw actions, so no temporal position is wasted and no future information enters the anchor.

Verification. We check equation 6 directly rather than assuming it.

Encoding a training window and encoding the corresponding rollout observation, we compare the resulting anchor tensors elementwise. Under head padding the maximum absolute difference is 0.000 — they are bit-identical. Under tail padding, where the anchor pair spans a frame the rollout cannot supply, it is 0.558. Perturbing the future frames of the training window leaves the head-padded anchor unchanged, confirming that no future information reaches it.

We report this as a correctness property rather than a result: it removes a candidate explanation for closed-loop failure, and is a precondition for any closed-loop number to be meaningful, but on its own it did not improve rollout success.

O DEPLOYED FOUR-STREAM CONFIGURATION

Figure 1 shows JEPA-DiT proper. Under delta-JEPA (§3.3) the predictive stream carries the temporal difference from the present observation and therefore no longer represents absolute scene content, so the deployed model adds a reconstruction video stream to hold it. Figure 4 shows that configuration: the two video streams read the same pixels, are mutually blind, and are both read by the action expert.

P JOINT WORLD-ACTION ATTENTION

All four streams are concatenated and processed by a *single* masked self-attention at every layer; their coupling is entirely a property of the mask, which is two-level rather than a chain. Text attends

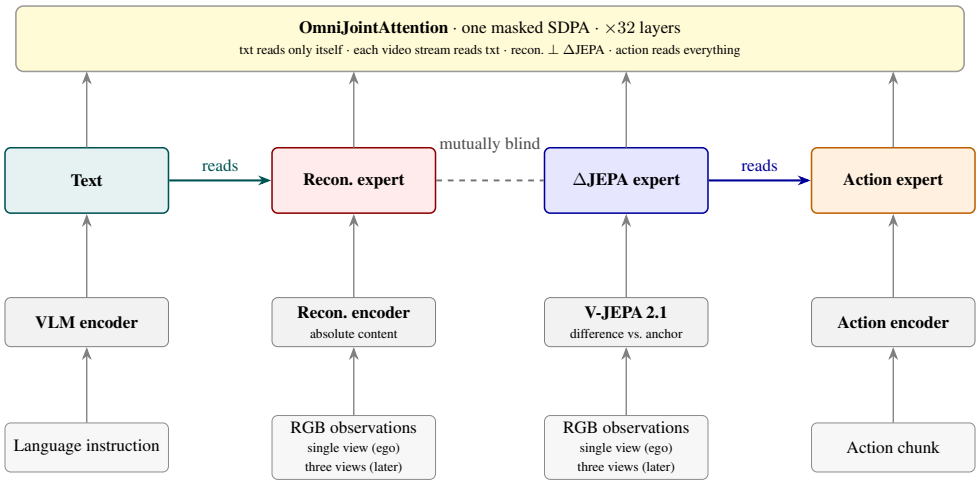


Figure 4: **The deployed four-stream configuration.** Identical to Figure 1 except for the added reconstruction stream. Both video encoders see the same pixels; the predictive stream encodes the difference against the anchor and the reconstruction stream the absolute content. The dashed link marks what the mask forbids: neither video stream attends the other, which is what prevents the differenced stream from recovering absolute content and collapsing the division of labour (§3.3). The action expert is the only place the two descriptions meet.

only to itself, so no stream can write into it. Each video stream attends to itself and to text but *not* to the other video stream—the mutual blindness that protects the division of labour of §3.3, since a differenced stream able to attend the reconstruction stream could re-acquire what equation 2 removes. The action stream attends to everything, and is therefore the single place where an absolute account of the scene and an appearance-invariant account of its change are combined.

This also departs from prior world-action models, which read a frozen T5 embedding through cross-attention: text there is a static key/value that the other streams may attend to sparsely, with no gradient pressure forcing otherwise. Here it co-evolves with them layer by layer.

Actions are assigned to latent frames by a group mask: the first latent frame is *given*, being the observed present, and the remaining $G = F - 1$ groups each receive the raw actions carrying the state from one latent frame to the next. An action is strictly in the future of the frame it starts from, so no group may attend a video position preceding it. We use the JOINT setting, in which each group attends all latent frames subject to that constraint, making the predicted future available to the action branch as evidence. The alternative—restricting each group to the first latent frame—leaves video prediction a pure auxiliary task whose *content* never reaches the actions. The two differ by the mask alone, which would make a matched pair a clean test of whether predicted video carries usable information; we list it among the outstanding ablations (Appendix C). Appendices Q and R state the assignment and both masks precisely.

Q ACTION GROUP ASSIGNMENT

Actions are assigned to latent frames by a group mask. The first latent frame is treated as *given* — it represents the present state, which is observed rather than predicted — so actions are distributed over the remaining

$$G = F - 1 \tag{7}$$

groups, group j receiving the N raw actions that carry the state from latent frame j to latent frame $j{+}1$. This convention is shared by both representations of §3.2; only F, and hence G, differs. It also encodes a temporal fact that is easy to lose: an action is always strictly in the future of the frame it starts from, so no action group may attend to a video position that precedes it.

R ACTION–VIDEO MASK VARIANTS

Appendix P states that our model uses the JOINT action–video mask and names its alternative. Writing g_j for the j-th action group and v_i for the video tokens of latent frame i:

- BASE: g_j attends v_0 only. The action branch is conditioned on the present observation and the instruction; the video streams, though still denoised and still contributing their terms to equation 1, are inaccessible to it.
- JOINT: g_j attends v_i for all $i \leq j$, the inequality being the temporal constraint of Appendix P — an action is strictly in the future of the frame it starts from.

A single checkpoint can be trained on a *mixture* of masks, sampling one per micro-step from a weighted set (BASE, JOINT, and the cross-directional variants V2A/A2V), so that it can be evaluated under several inference-time masks. Two conditions apply. Because the mask is sampled per micro-step, the number of gradient-accumulation micro-steps must be at least the number of active modes for every mode to appear in each optimiser step. And evaluation must use the mask the checkpoint was trained to serve: a checkpoint trained under JOINT and evaluated under BASE is out of distribution, and conversely. Our main-results checkpoint is trained and evaluated under a single fixed mask, so neither condition binds it.

S ADDITIONAL EXPERIMENTAL DETAIL

The task list, rollout budget, chunk horizon, held-out treatment of the randomised configuration and aggregation rule are given in §4.1. This appendix records the remaining reproduction detail for the post-training stage that produces the main result; Table 9 lists it.

T ATTENTION ANALYSIS (FULL MAPS)

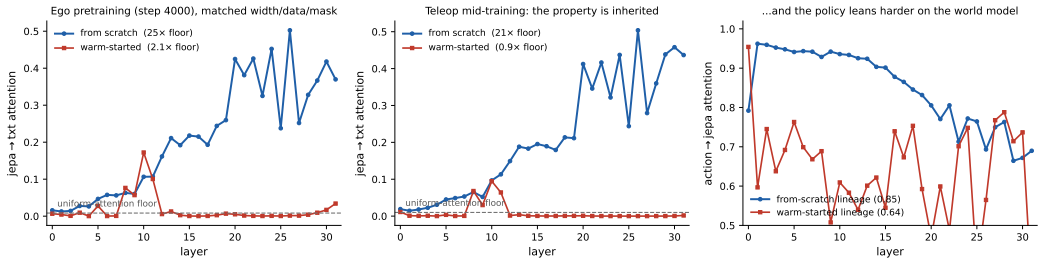


Figure 5: **Training from scratch is what builds the language pathway.** *Left*: in ego pretraining, two runs matched in width, data, bucketing, T_g, stage and mask but differing in initialisation (and in learning rate, the one uncontrolled factor). The from-scratch predictive stream organises its attention to text by depth and reaches $25\times$ the uniform-attention floor (dashed); the warm-started run stays at $2.1\times$, indistinguishable from attending to nothing in particular. *Middle*: the property is inherited—two mid-training runs of the same width, differing only in which ego checkpoint they descend from, reproduce the separation, and the warm-started lineage sits *at* the floor ($0.9\times$). *Right*: the consequence for control—the lineage with a working language pathway yields an action expert that attends markedly more to the predicted world state at every depth.

Full per-layer language profile. In the deployed four-stream model the reconstruction stream's attention to text rises from 0.024 at layer 0 to a peak of 0.687 at layer 14 and stays above 0.33 through layers 9–15; the predictive stream's own peak is 0.253 at layer 20, $2.7\times$ smaller.

Aggregate allocation. Across the stack the action expert draws 68% of its attention from the two world-model streams (0.386 reconstruction, 0.295 predictive) and keeps 26% for itself, the small remainder going to text: visual context is used, not decorative. The two divide by depth—predictive early, reconstruction throughout, with the deep layers refining the chunk autoregressively.

Table 9: Post-training configuration for the main-results checkpoint (§4.4). The run is a full fine-tune—no low-rank adaptation—initialised from the teleoperation mid-training checkpoint, with both video encoders frozen.

Setting	Value
Optimisation	
Optimiser	Muon+
Learning rate	3×10^{-5}, warmup 600 steps then constant
Global batch size	512
Precision	bf16, gradient checkpointing
EMA	decay 0.999, fp32
Diffusion	
Timestep distribution	Beta, shift 5.0
Prediction target	x-prediction (JEPA stream), velocity (others)
Loss weights	video 1.0, JEPA 1.0, action 1.0
Text dropout	0.1
Streams and masking	
Attention mask	txt self-only; each video stream $\leftarrow$ txt; action $\leftarrow$ all; the two video streams mutually blind
Layers per expert	32
Joint attention	width 4096, 32 heads $\times$ 128
Expert widths	recon. 4096, text 4096, JEPA 2048, action 1280
Feed-forward	SwiGLU, sized from each expert width
Encoder feature width	1664 (V-JEPA 2.1 ViT-G), tubelet $\tau = 2$
Reconstruction encoder	causal video VAE, $8\times$ spatial and $c = 8$ temporal; latent $[64, 3, 24, 20]$ (16 channels $\times$ patch 2^2)
Input resolution	384×320, three views as one montage
Anchor convention	head padding, anchor at $t = 0$ (Appendix N)
JEPA input	temporal difference against the anchor
Data and action space	
Training set	50 RoboTwin 2.0 tasks, `demo_clean` only
Usable windows	540,688
Pixel pipeline	online decode, 17 frames, colour jitter $(0.2, 0.2, 0.2, 0)$
Predictive latent frames T_g	3 (1 anchor + 2 predicted, so $G = 2$); latent grid 24×20
Action dimension / chunk	16 (14 actuated, slot-mapped) / 32
Proprioception	16-d, injected on the text stream
Evaluation	
Inference chunk horizon	$K = 2$, joint rollout
Rollouts	20 per task per configuration

Table 10: **Where the gap lives.** Per-task success (%) for the six tasks with the highest and the six with the lowest retention (Rt. = randomised / clean). At 20 rollouts per cell one success is 5 points, so tasks that match or exceed their clean score are reported as ≥ 100 rather than at a precision the protocol does not support. No task is at zero in either configuration; six match or exceed their clean score under randomisation. Full 50-task breakdown in Appendix I.

Highest retention				Lowest retention			
Task	Cl.	Rd.	Rt.	Task	Cl.	Rd.	Rt.
`place_object_scale`	75	80	≥ 100	`stamp_seal`	90	10	11
`place_object_stand`	90	95	≥ 100	`hanging_mug`	45	5	11
`place_empty_cup`	95	100	≥ 100	`handover_block`	55	10	18
`grab_roller`	100	100	≥ 100	`place_dual_shoes`	70	20	29
`shake_bottle`	100	100	≥ 100	`blocks_ranking_size`	75	25	33
`shake_bottle_horizontally`	100	100	≥ 100	`place_mouse_pad`	85	30	35

Depth profile of the action stream. On the deployed four-stream model, reliance on the predictive stream is front-loaded—action→jepa peaks at 0.576 in layer 2 and decays to 0.047 by layer 31—while action→vae oscillates without trend across the whole stack (0.176–0.530, high at both ends) and action→action climbs from 0.027 to 0.620 at layer 25. World information enters the action stream early, carried mainly by the predictive description; the reconstruction description remains available throughout; the deep layers increasingly refine the chunk autoregressively.

Mask self-checks. The visibility structure is verified numerically rather than assumed: txt→{everything downstream} = 0.0000, jepa→act = 0.0000, and in the four-stream model jepa→vid and vid→jepa are both exactly 0.0000 at *every* layer, confirming the two video streams never see each other. Averages alone would not reveal a per-layer violation, which is why the check is made layerwise.

Attention is not by itself evidence of useful prediction. Where a stream directs attention says nothing about whether what it produces is worth attending to. Appendix V addresses that separately, comparing the predictive stream's autoregressive prediction against ground truth in a shared PCA basis on the same deployed model ($\Delta\cos = 0.878$ against a copy-anchor floor of 0, with an anchor self-check that passes on every clip).

All layers are reported, and this matters. Every attention number in this paper is computed over all 32 layers rather than a sampled subset, because equidistant sampling misreads non-monotonic curves. Sampling layers $\{0, 8, 16, 24, 31\}$ on the checkpoint of §4.3 renders action→jepa as a monotonic decay when it in fact rises to 0.576 at layer 2 first; renders action→vae as a gradual decline when it in fact oscillates (trough 0.176 at layer 25, peak 0.530 at layer 30); and puts the vid→txt peak at 0.318 when the true peak is 0.687 at layer 14, an error of 2.2×. The probe also disables fused attention kernels and recomputes the softmax explicitly, since fused implementations do not return the attention matrix.

The core finding is summarised in Figure 5 (§4.3). Here we give complete per-layer, per-stage maps from a separate pair of from-scratch runs: the ego stage's jepa→txt profile (Figure 6a), the same quantity after teleop mid-training (Figure 6b), and the action stream's allocation both as per-layer curves (Figure 6c) and as a stacked share (Figure 6d). Each panel's rows sum to one (a mask sanity check); the unidirectional stream visibility structure is verified by the exact zeros (e.g. txt→jepa = 0).

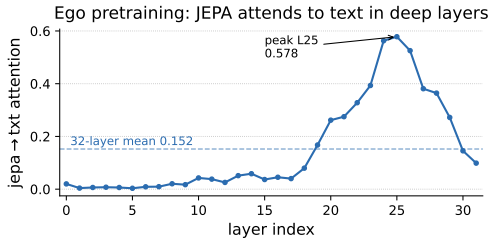
U ABSOLUTE-JEPA PREDICTION (MID-TRAINING)

At the teleop stage we decode the JEPA stream's autoregressive prediction and compare it, in the *same* PCA basis, to the ground truth (both projected onto the GT-fitted components). The prediction tracks the low-dimensional structure of the future rather than washing out toward a static mean: over the clip set the mean $\Delta\cos$ (the change relative to the copy-anchor floor, the primary criterion) is 0.506. Figure 7 shows nine held-out clips; as in Appendix V, the panels are illustrative rather than a benchmark and the aggregate is what we rely on. This is the absolute-feature variant; Appendix V runs the same analysis on the differenced stream the deployed model uses. The two $\Delta\cos$ values are *not* comparable: they come from different checkpoints at different points in training, and a differenced stream is scored against a different floor. Neither section is evidence that one variant predicts better than the other—that comparison would need the matched pair we list as outstanding (Appendix C).

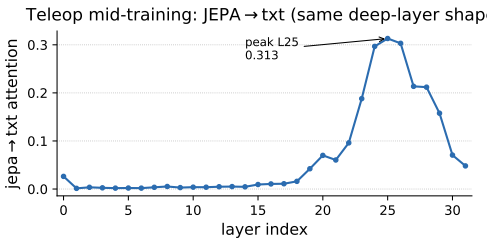
V DELTA-JEPA PREDICTION AND REPRESENTATION GEOMETRY

This section examines what the predictive stream actually predicts, on the four-stream delta-JEPA model that our main result is built on (§4.4): 12 held-out teleoperation clips, four autoregressive rounds spanning 4.3 s. The panels are illustrative rather than a benchmark; the aggregate numbers are what we rely on.

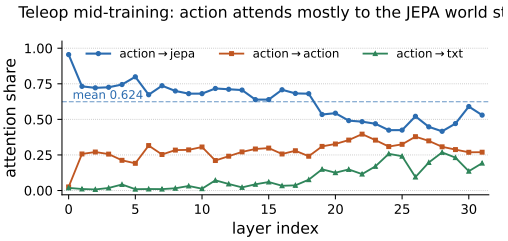
The right criterion for a differenced stream. Table 11 reports the summary. Two quantities must be kept apart. The *absolute* cosine between prediction and ground truth averages 0.295, which



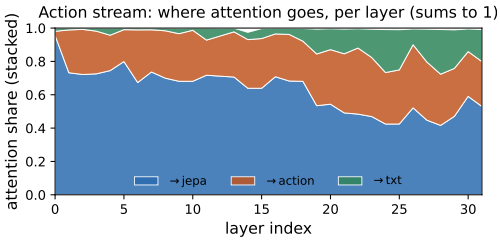
(a) Ego pretraining, jepa→txt.



(b) Mid-training, jepa→txt.



(c) Mid-training, action→jepa.



(d) Action-stream shares, stacked.

Figure 6: **Per-layer attention maps.** These panels come from a separate pair of from-scratch runs (ego and teleop, step 8,000) and show the shape of the profile at full per-layer resolution; the controlled initialisation comparison and its values are Figure 5 and Table 2. (a) and (b): the predictive stream's attention to text is near zero in shallow layers and dominant in deep ones, at both stages. Magnitudes are comparable across stages because the predictive stream attends only to itself and to text in both configurations, so the normalising denominator is the same. (c): the action stream draws heavily on the predictive world representation in shallow layers and shifts toward itself and text with depth. (d): the same data as a share decomposition, each column summing to one.

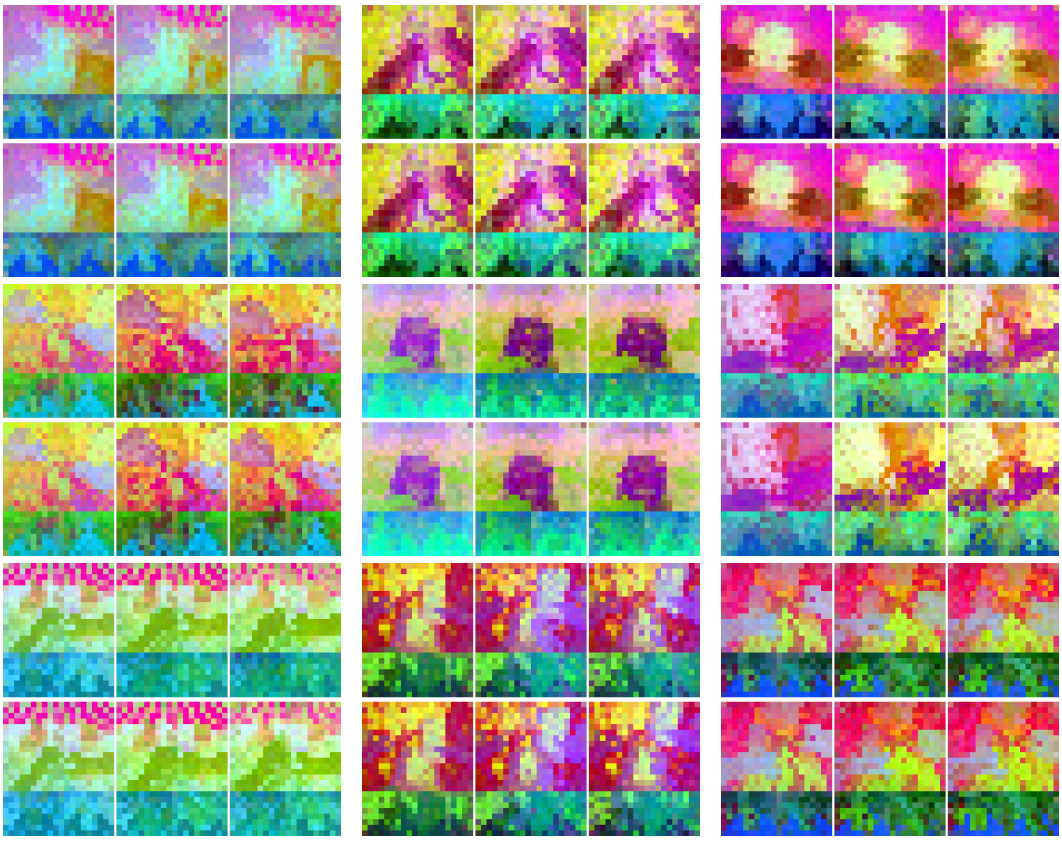


Figure 7: **JEPA prediction vs. ground truth in a shared PCA basis** (teleoperation mid-training; nine held-out clips). In each panel GT and prediction are projected onto the same GT-fitted PCA components, so a faithful prediction reproduces the GT colour field rather than washing out toward a static mean; $\Delta\cos$ is the primary criterion (mean 0.506 over the clip set). The prediction tracks the low-dimensional structure of the future across diverse scenes. The panels illustrate; the clip-set mean is what we rely on.

read naively suggests a poor prediction. The Δ cosine—measured against the copy-anchor floor, i.e. the score obtained by predicting no change at all—averages 0.878. The gap between these two is not a discrepancy but the structural signature of a differenced representation: under equation 2 the stream predicts $[\, f_0, \Delta_1, \Delta_2 \,]$, so most of the absolute feature mass is carried by the anchor and the predicted part is small in norm and large in information. Judging such a stream by absolute cosine measures mainly how much of the window is static. For the same reason the relative L_2 of the two video streams (0.483 reconstruction, 1.116 predictive) lie in different value ranges—one is a VAE latent, the other a feature difference—and must not be compared as magnitudes.

Self-check. The anchor group is re-pinned to the encoded observation at every step, so its cosine must be 1 by construction. Across all 12 clips the *minimum* anchor cosine is 1.0000; we report the minimum rather than the mean because a mean would average an anomalous clip away.

A negative we do not overstate. The effective rank of the prediction is below that of the ground truth (a spectral ratio of 0.819), which is consistent with some smoothing. We do not read it as evidence of representational collapse: effective rank is a tail-accumulating spectral-entropy statistic, and settling the question requires per-component energy ratios, which we did not compute here. An earlier reading of this same statistic as "over-smoothed prediction" was subsequently contradicted by exactly that per-component check, and we record the episode rather than repeat the inference.

Why no decoded video for this stream. The reconstruction stream can be rolled out to pixels through its decoder, and Figure 8 accordingly shows the predictive stream in PCA rather than in pixels. The reason is intrinsic to delta-JEPA: the stream emits temporal feature *differences*, whereas the available JEPA decoder was trained on absolute features. Decoding the differences directly produces noise; one would first have to reaccumulate $g_1 = f_0 + \Delta_1$, $g_2 = g_1 + \Delta_2$. This is a genuine cost of the representation—the predictive stream is harder to inspect than a reconstruction stream—and §5 lists it among the limitations.

Table 11: Delta-JEPA prediction quality on the mid-training checkpoint that seeds the main result (12 held-out clips, four autoregressive rounds). $\Delta\cos$ is measured against the copy-anchor floor and is the criterion appropriate to a differenced stream; absolute cosine is reported for completeness and is low by construction. Dashes mark quantities we did not compute for the reconstruction stream, whose prediction quality is not what this section examines. The two streams' relative L_2 values are in different units and are not comparable.

Stream	rel-$L_2 \downarrow$	$\Delta\cos \uparrow$	abs. cos
Reconstruction (VAE)	0.483	—	—
Predictive (delta-JEPA)	1.116	$\mathbf{0.878}$	0.295
Geometry (predictive stream)			
Effective rank, ground truth		53.8	
Effective rank, prediction		44.0	
Anchor cosine (minimum over clips)		1.0000	

W LIMITATIONS

Delta-JEPA rests on a mechanistic and diagnostic case, not a closed-loop swap against an absolute predictive stream—the matched pair we most want and do not have (Appendix C). The predictive latent has no pixel decoder, so world states are compared in representation space rather than inspected. And our evidence comes from one backbone family and one benchmark family; whether the two-branch rule transfers is untested.

X FUTURE WORK

The mechanism makes a cheap, falsifiable prediction: under randomisation the action expert should shift attention from the reconstruction stream toward the predictive one. Beyond that, scaling the from-scratch corpus and extending the rule to further embodiments and architectures.

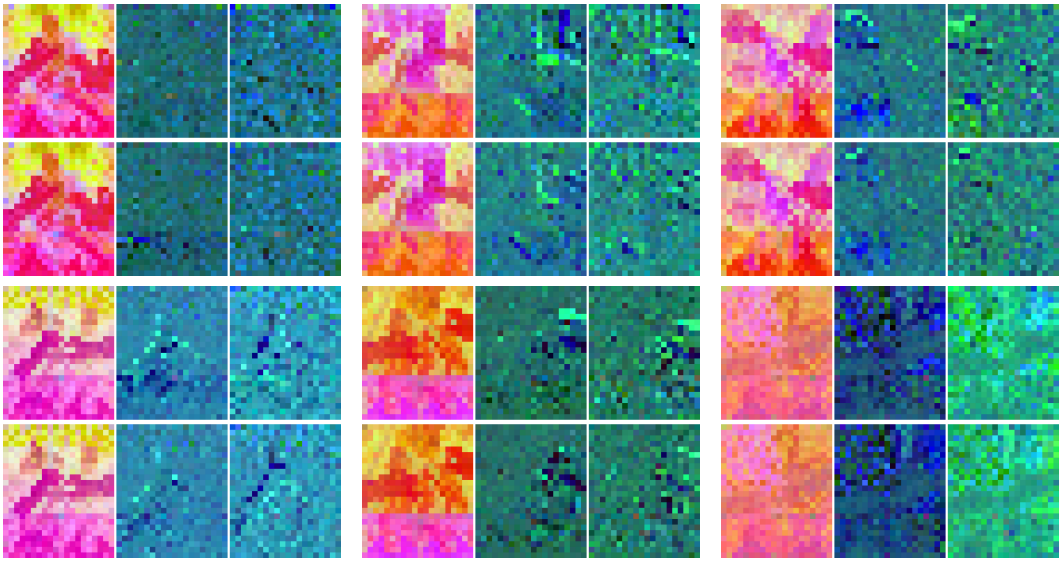


Figure 8: **Delta-JEPA prediction vs. ground truth in a shared PCA basis**, six held-out clips spanning the quality range of the set (ordered by the reconstruction stream's relative L_2, best first: *throw napkin ball into bin*, *hang stick on rack*, *lift cola from counter*, *swing bottle*, *place pencil in holder*, *rotate and move curtain*). Each panel is a false-colour feature map: the first *three* principal components of the ground-truth feature field are mapped to the R, G and B channels, and the prediction is projected onto those same components with the same per-channel scaling—so a faithful prediction reproduces the ground-truth colour field, and washing out toward a static mean shows up as loss of colour structure. Fitting the basis on the ground truth and reusing it for the prediction is essential; fitting each separately would rescale them independently and hide exactly the failure the panels are meant to expose. Per-clip $\Delta\cos$ ranges from 0.840 to 0.900 across the twelve, a narrow spread given the diversity of the scenes.