TERRA: LEARNING TRANSPORTABLE LATENT ACTIONS THROUGH TEMPORAL EFFECT REPRESENTATION AND RELATIONAL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent actions supervise robot policies with action-like codes inferred from visual transitions, and their usefulness hinges on two questions: what a code keeps from a transition, and whether it still means the same thing when reused in a different initial state. The first is a tension in time: an endpoint difference discards how motion unfolds, while the full sequence admits nuisance variation. The second is left open by reconstruction, which only ever observes a latent together with the state it came from. We argue that both questions can be answered in the same place. **TERRA** (*Temporal Effect Representation and Relational Alignment*) describes a transition by a compact temporal effect, its net feature change together with a low-order within-window dynamics component, and learns a continuous latent from this effect. The same effect space then serves as the reference for reuse: **Effect-Anchored Transport (EAT)** decodes a latent in other initial states and anchors the resulting effect to the one observed at its source, so that the latent is shaped by what it does across contexts rather than only by the transition it came from. With frozen linear readers, TERRA predicts actions more accurately than UniVLA and a LAPA-style baseline, degrades more slowly under visual distractors, and keeps transported transitions faithful to the donor action as the recipient context moves farther away; a same-budget control shows that these gains come largely from EAT. At matched pretraining scale, the complete system reaches 93.4% average success on LIBERO, compared with 91.8% for UniVLA.

1 INTRODUCTION

Latent actions provide an intermediate supervision signal between visual transitions and robot control. A latent-action model assigns an action-like representation to an observed transition, which can then supervise a policy conditioned on the current observation and instruction. Recent work has explored discrete task-centric latents (Ye et al., 2025; Bu et al., 2025), continuous latent motion from frame-pair feature differences (Yang et al., 2026), and structured continuous action spaces (Li et al., 2026). Despite this progress, a basic question remains unresolved: *what should a latent action preserve from a visual transition, and how should it behave when reused?*

A first challenge is **selective temporal information preservation**. Strong compression can remove distinctions that matter for control, whereas increasing latent capacity can also admit action-irrelevant appearance and exogenous variation (Yang et al., 2026; Lee et al., 2026). This tension is particularly important in time. Continuous latent-motion models operate on a frame pair, so their input reduces to the endpoint difference: it records how much the scene changed, not how the change unfolded within the window. Two executions that reach the same state at different speeds therefore receive the same latent (Fig. 1a), although they correspond to different actions. Conversely, retaining the entire temporal sequence exposes the representation to substantially more visual variation.

A second challenge is **cross-context effect consistency**. Standard reconstruction objectives observe only matched state–transition pairs and can therefore encode context-specific variation into the latent (Lee et al., 2026). Policy supervision is most useful when similar effects receive compatible latents across episodes, so it matters how a latent behaves when applied in a different initial state; we refer to this reuse as *transport*. A latent that fits its source transition can still decode to a different effect once the scene changes (Fig. 1b).

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

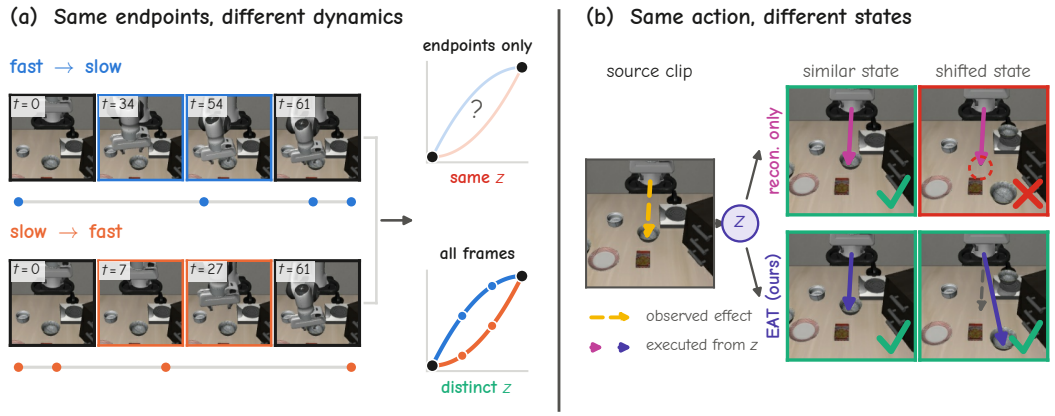


Figure 1: **Two questions for a latent action**. (a) *What to keep.* Two executions share their first and last frames (t=0, t=61) but move at different rhythms (fast-then-slow vs. slow-then-fast, dots mark frame times). A latent built from the endpoint difference alone assigns them the same code; adding a within-window dynamics component keeps them distinct. (b) *How to behave.* The latent z of a source clip (yellow: its observed effect, the gripper lowering onto the bowl) is decoded from two other initial states. Arrows show the effect each model decodes. Trained by reconstruction only, the effect follows the source in a similar state but drifts once the bowl moves; with EAT it stays consistent with the source effect in both, so the same latent transfers across states.

We address both challenges with one principle: *the compact effect space that decides what a latent should keep also provides the reference for how it should behave in a new context*. We instantiate this principle with **TERRA** (*Temporal Effect Representation and Relational Alignment*), which first summarizes a short transition by its *temporal evidence*, a net component that captures accumulated feature change and a dynamics component that captures its low-order within-window trend, and compresses this evidence into a continuous latent. It then applies **Effect-Anchored Transport (EAT)**: the latent is decoded in recipient states from the same data source, and the direction of the resulting effect is anchored to the one observed at the donor. Relative-effect consistency, transition-distribution matching, and nuisance invariance regularize this refinement. Figure 2 summarizes the framework.

Adding the dynamics component to the endpoint difference lowers action NMSE and makes the within-window action trend linearly recoverable. EAT lowers action NMSE further and keeps transported transitions faithful to the donor action as recipient distance grows, which the same amount of additional reconstruction does not achieve. The complete system reaches 93.4% average LIBERO success against 91.8% for UniVLA at matched pretraining scale.

Our contributions are:

- A **temporal-effect representation** that extends frame-pair latent motion with a low-order within-window dynamics component, retaining how change unfolds without encoding the full sequence.
- **Effect-Anchored Transport**, which uses the same effect space to supervise how a latent behaves when decoded in other initial states, constraining what the latent does rather than what it is.
- Controlled evaluations showing that each component contributes to action alignment, nuisance robustness, and transport stability, and a matched-scale system comparison showing higher LIBERO success than UniVLA.

2 Related Work

Latent action representations. Latent action models infer action-like representations from visual transitions and use them as intermediate supervision for robot learning. Early work learned inverse-dynamics labels from unlabeled video (Baker et al., 2022) and discrete latent actions for controllable generation and policies (Bruce et al., 2024; Schmidt & Jiang, 2024); LAPA and UniVLA scale this

recipe to vision-language-action pretraining with quantized codes (Ye et al., 2025; Bu et al., 2025), and Moto, IGOR, AdaWorld, and villa-X refine what such codes capture (Chen et al., 2025c; 2024; Gao et al., 2025; Chen et al., 2025b). More recent work explores continuous representations to preserve richer motion structure: CoMo learns continuous latent motion from frame-pair feature differences (Yang et al., 2026), and RotVLA introduces structured continuous latents with temporal composition for VLA learning (Li et al., 2026). Zhang et al. (Zhang et al., 2025) analyze what these models actually encode. These works establish both discrete and continuous latent actions as viable interfaces between visual transitions and downstream control.

Cross-context grounding and latent consistency. A separate line of work studies how latent actions remain meaningful beyond the transition from which they are inferred. Olaf-World orients latent actions using temporal feature differences from a frozen video encoder for controllable video world models (Jiang et al., 2026), while cycle-based approaches regularize latent identity through generated transitions (Chen et al., 2025a). LAOM shows that action-correlated distractors degrade latent actions learned by reconstruction and measures latent quality by linear probing (Nikulin et al., 2025); recent analyses further examine nuisance sensitivity, action recoverability, and the relationship between latent-quality proxies and downstream control (Lee et al., 2026; Bu et al., 2026).

Continuous VLA prediction. Continuous action prediction is increasingly common in vision-language-action models. OpenVLA-OFT replaces autoregressive action-token decoding with continuous action heads (Kim et al., 2025a), while π_0 uses a flow-matching action expert (Black et al., 2025). RotVLA jointly models continuous latent and physical actions (Li et al., 2026), and LAFP studies latent-policy learning with flow matching (Lyu et al., 2026).

These prior works have studied temporal latent construction, cross-context consistency, and continuous policy prediction largely as separate problems. TERRA uses one effect space for both: temporal evidence defines what the latent keeps, and Effect-Anchored Transport uses the same evidence to supervise how it behaves in other states.

3 TERRA: Temporal Effect Representation and Relational Alignment

TERRA learns transportable continuous latent actions through two complementary components: *Temporal Effect Representation* (TER) and *Relational Alignment* (RA). TER constructs a compact latent from the temporal evidence of an observed transition, while RA refines how this latent behaves when applied in a different initial state, through Effect-Anchored Transport (EAT).

We call a latent *transportable* within a data source if, when applied in a different initial state, the resulting transition preserves the direction of its source temporal effects. We refer to this cross-context reuse as *transport*. Both components use frozen visual features without robot action labels. After refinement, the latent tokenizer is frozen and used to provide latent-action targets for downstream policy training.

3.1 Temporal Effect Representation (TER)

Temporal evidence. For a transition $x_i = (o_{i,0}, \ldots, o_{i,K})$, we extract frozen DINOv2 features (Oquab et al., 2023) and spatially pool them into $F_{i,k} \in \mathbb{R}^{S\times C}$, where S denotes the number of spatial cells and C the feature dimension. Let $\Delta F_{i,k} = F_{i,k+1} - F_{i,k}$. We omit the transition index i when unambiguous.

Rather than retaining either only the endpoint difference or the full increment sequence, TERRA summarizes a transition by two low-order temporal components, which we call its *temporal evidence*. For any increment sequence $Y = (Y_0, \ldots, Y_{K-1})$, we define the *net component* N and the *dynamics component* A_1 as

$$N(Y) = \sum_{k=0}^{K-1} Y_k, \qquad A_1(Y) = \sum_{k=0}^{K-1} b_1(k)\, Y_k, \qquad \langle b_1, \mathbf{1}\rangle = 0. \tag{1}$$

The net component corresponds to the constant temporal mode and captures accumulated change; for observed transitions, $N(\Delta F) = F_K - F_0$ by telescoping. The dynamics component uses the

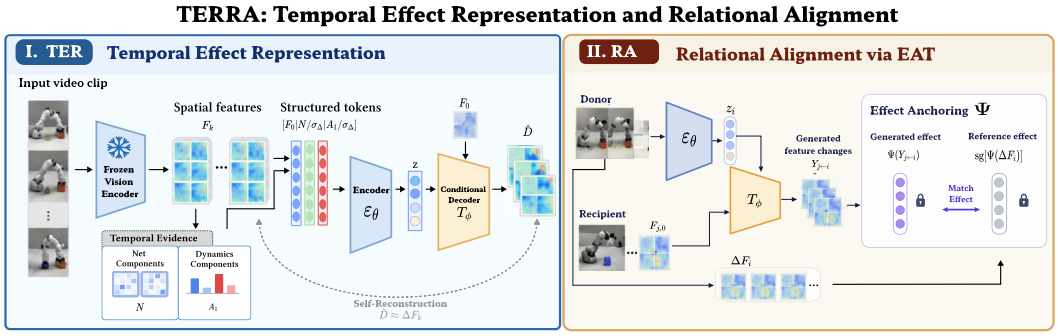


Figure 2: **Overview of TERRA.** Temporal Effect Representation compresses the initial-state feature F_0 and the temporal evidence (net component N and dynamics component A_1) into a continuous latent z. Effect-Anchored Transport decodes a donor latent in recipient states and anchors the resulting effect to the donor's observed effect. See Sec. 3 for details.

lowest-frequency nonconstant DCT-II mode b_1 and captures coarse within-window variation. Together they retain the two lowest-order temporal components of the increment sequence. The temporal evidence is the measured form of the temporal effect that TERRA represents; Section 3.2 reuses the same two components to supervise transport.

We collect the temporal evidence into the *effect vector*

$$\xi(Y) = \mathrm{vec}\left[N(Y), A_1(Y)\right]. \tag{2}$$

Structured tokens and encoding. Feature increments are normalized by a single global scale σ_Δ, defined as the root-mean-square magnitude of raw DINOv2 feature increments estimated once before training and then kept fixed.

We concatenate the initial-state feature and the normalized temporal evidence along the token dimension into the *structured tokens*

$$\left[F_0 \mid N(\Delta F)/\sigma_\Delta \mid A_1(\Delta F)/\sigma_\Delta\right].$$

A transformer with spatial and evidence-type embeddings aggregates the three token groups through four learned queries. Their outputs form

$$z = E_\theta(x) \in \mathbb{R}^{4\times 64},$$

with no quantization or predefined spatial or temporal roles for the latent slots. We refer to E_θ as the *latent tokenizer*.

Self-reconstruction. The conditional decoder T_ϕ maps the initial-state feature and the latent to a normalized prediction of the increment sequence,

$$\hat{D} = T_\phi(F_0, z) \in \mathbb{R}^{K\times S\times C}. \tag{3}$$

The tokenizer and decoder are jointly trained so that $\hat{D} \approx \Delta F/\sigma_\Delta$:

$$\mathcal{L}_{\text{self}} = \mathbb{E}_x\left[\frac{1}{KSC}\left\|\hat{D} - \frac{\Delta F}{\sigma_\Delta}\right\|_F^2\right], \tag{4}$$

where ΔF stacks all K increments. Predicting the increment sequence requires the decoder to model how change is distributed within the window rather than only its endpoint. The tokenizer itself, however, receives only the temporal evidence N and A_1; higher-order temporal detail is not available to it as input.

3.2 Relational Alignment (RA) via Effect-Anchored Transport

The representation above is learned only from matched state–transition pairs. Such reconstruction does not determine how the same latent should behave when applied in a different initial state.

By the definition above, the desired behavior is that the transported transition preserve the donor's temporal-effect direction; EAT enforces this behavior directly.

We sample donor transitions and recipient initial states from the same data source, limiting large embodiment and observation-domain shifts. EAT combines four complementary objectives: *effect anchoring*, *relative-effect consistency*, *transition-distribution matching*, and *nuisance invariance*.

Transport. Let x_i be a donor transition and $F_{j,0}$ a recipient initial state from another episode of the same source. We encode the donor and decode its latent in the recipient state,

$$z_i = E_\theta(x_i), \qquad \hat{D}_{j\leftarrow i} = T_\phi(F_{j,0}, z_i), \qquad Y_{j\leftarrow i} = \sigma_\Delta \, \hat{D}_{j\leftarrow i}. \tag{5}$$

Here $Y_{j\leftarrow i}$ is the generated feature-increment sequence induced by donor latent z_i in recipient state $F_{j,0}$, expressed in feature-increment units so that the temporal evidence of Eq. 1 can be evaluated on it. The tokenizer sees only the donor transition, and the decoder receives no recipient future frames. EAT introduces no additional trainable transport network: T_ϕ is the same conditional decoder used for self-reconstruction.

Effect anchoring. We compare effects through a fixed *effect descriptor*

$$\Psi(Y) = \mathrm{std}\big(R^\top \xi(Y)\big), \tag{6}$$

where R is a fixed Gaussian random projection and std standardizes each coordinate with mean and variance estimated once from real transitions (Appendix A.2). Ψ has no learnable parameters, so the temporal evidence used to construct the representation is the same quantity used to evaluate its behavior after transport. The observed donor transition provides the reference:

$$\mathcal{L}_{\mathrm{effect}} = \mathbb{E}_{i,j}\left[1 - \cos\left(\Psi(Y_{j\leftarrow i}), \mathrm{sg}[\Psi(\Delta F_i)]\right)\right], \tag{7}$$

where sg denotes stop-gradient. This objective aligns the direction of the transported effect with that of the observed donor transition while leaving its magnitude free to depend on context. The generated branch remains differentiable, so $\mathcal{L}_{\mathrm{effect}}$ updates both T_ϕ and the online tokenizer E_θ.

Relative-effect consistency. Effect anchoring constrains individual transported effects. We additionally encourage relationships between different latent effects to remain stable across recipient contexts. For this objective, we maintain an exponential-moving-average target tokenizer $E_{\bar{\theta}}$ that tracks E_θ but receives no gradients. Detached target latents from recent same-source samples provide the donor latents for relative-effect consistency. Let $\bar{z}_i$ and $\bar{z}_{i'}$ denote two such donor latents. For recipient states $F_{j,0}$ and $F_{j',0}$, define

$$e_s^q = \Psi\left(\sigma_\Delta T_\phi(F_{s,0}, \bar{z}_q)\right), \qquad q \in \{i, i'\}, \quad s \in \{j, j'\}, \tag{8}$$

and

$$r_s^{i,i'} = e_s^i - e_s^{i'}.$$

We optimize

$$\mathcal{L}_{\mathrm{rel}} = \mathbb{E}_{i,i',j,j'}\left[1 - \cos\left(r_j^{i,i'}, r_{j'}^{i,i'}\right)\right]. \tag{9}$$

Effect anchoring constrains the direction of each transported effect, whereas relative-effect consistency stabilizes how different latent effects compare across recipient contexts. Together, they reduce context dependence in the effect space measured by Ψ without requiring the full generated transition to be state-independent.

Because the donor latents used by $\mathcal{L}_{\mathrm{rel}}$ are detached EMA-target representations, this term directly regularizes the shared decoder rather than backpropagating through the donor tokenizer. The online tokenizer is refined directly through $\mathcal{L}_{\mathrm{self}}$, $\mathcal{L}_{\mathrm{effect}}$, and $\mathcal{L}_{\mathrm{inv}}$.

Transition-distribution matching. The temporal evidence does not constrain every component of the generated transition. We therefore use maximum mean discrepancy (MMD) (Gretton et al., 2012) to match the empirical distributions of projected real and generated feature increments, denoted $\mathcal{L}_{\mathrm{dist}}$. This term operates in transition space rather than latent space and does not assign a ground-truth counterfactual transition to an individual recipient. Its exact form is given in Appendix A.2.

Algorithm 1 Effect-Anchored Transport (EAT) refinement

Require: dataset $\mathcal{D}$, online tokenizer E_θ, decoder T_ϕ, EMA tokenizer $E_{\bar{\theta}}$, same-source queues $\{Q_d\}$

for each refinement step **do**
  $\bar{\theta} \leftarrow \beta_{\text{EMA}}\bar{\theta} + (1-\beta_{\text{EMA}})\theta$
  sample source d, donor x_i, recipient $F_{j,0} \sim \mathcal{D}_d$
  $z_i \leftarrow E_\theta(x_i)$
  $\hat{D}_{j\leftarrow i} \leftarrow T_\phi(F_{j,0}, z_i)$; $Y_{j\leftarrow i} \leftarrow \sigma_\Delta \hat{D}_{j\leftarrow i}$
  $\mathcal{L}_{\text{effect}} \leftarrow 1 - \cos(\Psi(Y_{j\leftarrow i}), \text{sg}[\Psi(\Delta F_i)])$
  $\bar{z}_i \leftarrow \text{sg}[E_{\bar{\theta}}(x_i)]$; enqueue $\bar{z}_i$ into Q_d
  sample $\bar{z}_a, \bar{z}_b \sim Q_d$ and $F_{u,0}, F_{v,0} \sim \mathcal{D}_d$
  $e^q_s \leftarrow \Psi(\sigma_\Delta T_\phi(F_{s,0}, \bar{z}_q))$, $q \in \{a,b\}$, $s \in \{u,v\}$
  $\mathcal{L}_{\text{rel}} \leftarrow 1 - \cos(e^a_u - e^b_u, e^a_v - e^b_v)$
  compute $\mathcal{L}_{\text{self}}, \mathcal{L}_{\text{dist}}, \mathcal{L}_{\text{inv}}$
  update (θ, ϕ) with $\mathcal{L}_{\text{EAT}}$
end for
return E_θ

Nuisance invariance. To suppress visual variation that should not determine the latent action, we use a BYOL-style objective (Grill et al., 2020): the online tokenizer receives a temporally consistent nuisance-perturbed view of a transition, while the EMA target tokenizer receives the corresponding clean view. The perturbation family consists of mild spatial cropping together with brightness and contrast variation, shared across all frames in the clip. We minimize the cosine discrepancy between the online and EMA-target latent representations, denoted $\mathcal{L}_{\text{inv}}$.

Importantly, these augmentations are used only for $\mathcal{L}_{\text{inv}}$. The effect descriptor Ψ and both effect-based objectives are computed from unaugmented transitions. Thus, the spatial perturbation used for latent invariance does not alter the image-grid coordinates on which effect anchoring is defined. Exact augmentation and target-network details are provided in Appendix A.2.

Joint optimization. We retain self-reconstruction on matched transitions throughout EAT refinement and optimize

$$\mathcal{L}_{\text{EAT}} = \mathcal{L}_{\text{self}} + \lambda_e \mathcal{L}_{\text{effect}} + \lambda_r \mathcal{L}_{\text{rel}} + \lambda_d \mathcal{L}_{\text{dist}} + \lambda_i \mathcal{L}_{\text{inv}}. \tag{10}$$

The loss weights and auxiliary-objective details are given in Appendix A.2.

After refinement, we freeze the online tokenizer E_θ and use it to generate latent-action targets for downstream policy training. Neither T_ϕ, the EMA target tokenizer, nor Ψ is required at policy inference time.

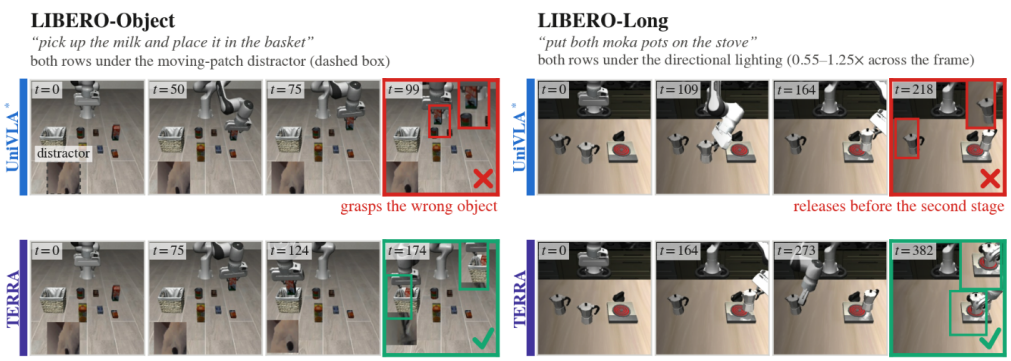
Scope. The descriptor Ψ is built from per-cell effects on the pooled feature grid, and same-source sampling does not guarantee exact geometric correspondence between donor and recipient states. We therefore interpret EAT as a source-restricted visual-effect regularizer rather than supervision for a ground-truth counterfactual physical action.

4 Experiments

4.1 Experimental Setup

Pretraining and baselines. We train latent-action models on an Open X-Embodiment mixture (Open X-Embodiment Collaboration, 2024) containing 29 sources and approximately 8.19M clips. All representation comparisons use the same frozen DINOv2 feature cache. For the system comparison, TERRA and UniVLA are retrained at the same latent-action pretraining data scale and evaluated on LIBERO (Liu et al., 2023). For representation analysis we additionally include a LAPA-style discrete VQ baseline in the same DINO feature space.

Policy protocol. Both systems start from OpenVLA-7B (Kim et al., 2025b), undergo latent-VLM pretraining on the same 10% subset of the pretraining mixture, and are then fine-tuned on LIBERO.



Method	Spatial	Object	Goal	Long	Avg.
LAPA†	73.8	74.6	58.8	55.4	65.7
Diffusion Policy†	78.3	92.5	68.3	50.5	72.4
Octo†	78.9	85.7	84.6	51.1	75.1
OpenVLA†	84.7	88.4	79.2	53.7	76.5
UniVLA*	92.3	92.6	93.4	88.7	91.8
TERRA (ours)	**94.1**	**95.2**	**93.8**	**90.6**	**93.4**

Table 1: **Closed-loop LIBERO success rate (%).**Top: one perturbed rollout each from LIBERO-Object and LIBERO-Long (Appendix B.2). TERRA and UniVLA* are retrained at the same latent-action pretraining data scale from the same OpenVLA-7B backbone and evaluated with the same rollout budget. †Published results under their original protocols.

TERRA uses its continuous query/readout and flow-based policy interface, and UniVLA its native discrete latent-token interface, so the comparison is between complete latent-action systems at matched pretraining scale. Each LIBERO suite contains 10 tasks and is evaluated with 50 rollouts per task.

Representation and robustness evaluation. Every tokenizer is frozen and probed with linear readers (Alain & Bengio, 2017) fit on real held-out transitions only, following the linear action-decoding action spaces are never mixed; source-level NMSEs are macro-averaged. Readers predict five targets from the K-step action sequence a: the 12-step chunk (*action*), the window sum $N(a)$ (*net*), the first step (*first*), the profile $A_1(a)$ of Eq. 1 (*profile*), and the per-step sequence (*seq.*). For nuisance robustness, the trajectory and labels are fixed while a moving visual distractor is scaled by $\alpha \in \{0, 0.25, 0.5, 1.0\}$; *nuisance drop* is the relative loss of net-action explained variance ($1 - \mathrm{NMSE}$) at $\alpha = 1$. Donor–recipient distance is the cosine distance between spatially averaged frozen-DINO initial-state features, ranked within each source into equal-count quartiles.

4.2 Closed-loop Manipulation

We first evaluate the complete system in closed-loop manipulation. Table 1 reports published systems for context and highlights the retrained TERRA–UniVLA comparison.

At matched latent-action pretraining scale, TERRA reaches 93.4% average LIBERO success against 91.8% for retrained UniVLA, with higher success on all four suites; the largest gains are on Object (+2.6) and Long (+1.9), the suites with the most object and horizon variation. Both systems share the VLM backbone, the latent-VLM pretraining mixture, and the LIBERO fine-tuning data; they differ in the latent-action model and the policy interface built on it. Sections 4.3 and 4.4 examine the latent representation directly. The rollouts in Table 1 illustrate two characteristic UniVLA* failures under visual perturbation, distractor confusion on Object and premature release on Long, which mirror the nuisance-robustness and within-window-structure results of Section 4.3.

4.3 Temporal Effect Representation

Action alignment and nuisance robustness. Figure 3 and Table 2 evaluate every tokenizer with the same frozen linear readers. On clean transitions, TERRA obtains the lowest 12-step action

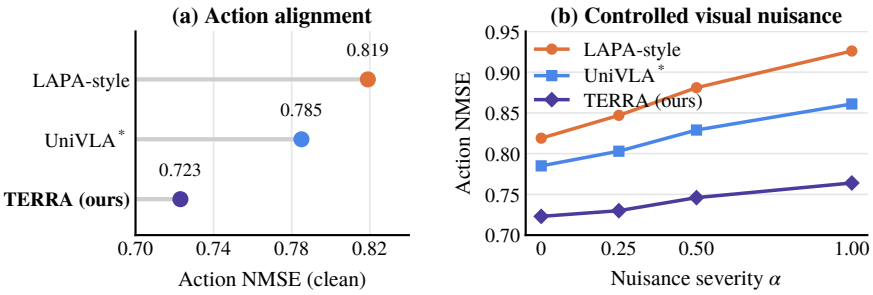


Figure 3: **Action alignment and nuisance robustness. (a)** Action NMSE on clean transitions for TERRA, UniVLA*, and a LAPA-style discrete baseline adapted to the common DINO feature space. **(b)** The same reader under a moving visual distractor of severity α; the robot trajectory and action labels are unchanged.

NMSE (.723), against .785 for UniVLA* and .819 for the LAPA-style baseline, and the ordering persists as the distractor severity grows: from $\alpha = 0$ to $\alpha = 1$, NMSE rises by .041 for TERRA, .076 for UniVLA*, and .107 for the LAPA-style baseline. Against UniVLA*, TERRA reads the net action, the first step, and the per-step sequence more accurately and loses less under the distractor (17.9% vs. 23.0%); UniVLA* reads the action profile slightly better (.875 vs. .909). Profile NMSE stays above .87 for every latent, so the within-window trend is only weakly linearly decodable from any of them.

What the dynamics component adds. The two 65k rows of Table 2 isolate the temporal evidence: two tokenizers with the same architecture and schedule, one receiving the net component only, i.e. the frame-pair feature difference of continuous latent-motion models (Yang et al., 2026), the other additionally receiving A_1. With the net component alone, the action profile is not linearly recoverable (.983): the endpoint difference records how much the scene changed, not how the change was distributed over the window. Adding A_1 lowers profile NMSE to .909, action NMSE from .817 to .807, first-step NMSE from .790 to .767, and the nuisance drop from 33.6% to 22.4%, while net-action NMSE moves from .703 to .734: the latent trades a small amount of endpoint precision for within-window structure, and the added temporal evidence lowers rather than raises sensitivity to the distractor.

4.4 Cross-context Refinement with EAT

Same-budget refinement. The +15k rows of Table 2 hold the optimization budget fixed. From the shared 65k checkpoint, 15k further reconstruction-only updates leave the temporal readouts where they were and lower 12-step action NMSE only from .807 to .784, whereas the same 15k updates with EAT lower net-action NMSE to .671, first-step NMSE to .717, sequence NMSE to .764, and action NMSE to .723, and reduce the nuisance drop to 17.9%; nuisance invariance accounts for roughly half of each gain. Profile NMSE is unchanged (.909), consistent with EAT acting on how

Table 2: **Frozen-reader ablation (NMSE, lower is better).** UniVLA* is the retrained baseline of Table 1. The two 65k tokenizers share the architecture and the self-reconstruction schedule and differ only in whether A_1 is provided; the three +15k rows continue from the $[F_0 \mid N \mid A_1]$ checkpoint under equal budget. *Action* predicts the 12-step chunk; *Net*, *First*, *Profile*, *Seq.* predict the net action, the first step, $A_1(a)$, and the per-step sequence; *Drop* is the relative loss of net-action explained variance at $\alpha = 1$. 95% intervals are bootstrap over the 29 sources (Appendix C.1). Best per column underlined.

Tokenizer	Action	Net	First	Profile	Seq.	Drop
UniVLA*	.785	.763	.778	<u>.875</u>	.785	23.0%
$[F_0 \mid N]$, 65k	.817	.703	.790	.983	.800	33.6%
$[F_0 \mid N \mid A_1]$, 65k	.807	.734	.767	.909	.807	22.4%
+15k reconstruction	.784	.732	.760	.909	.807	22.6%
+15k EAT w/o invariance	.764	.700	.733	.915	.784	19.7%
+15k full EAT (**TERRA**)	<u>.723</u>	<u>.671</u>	<u>.717</u>	.909	<u>.764</u>	<u>17.9%</u>

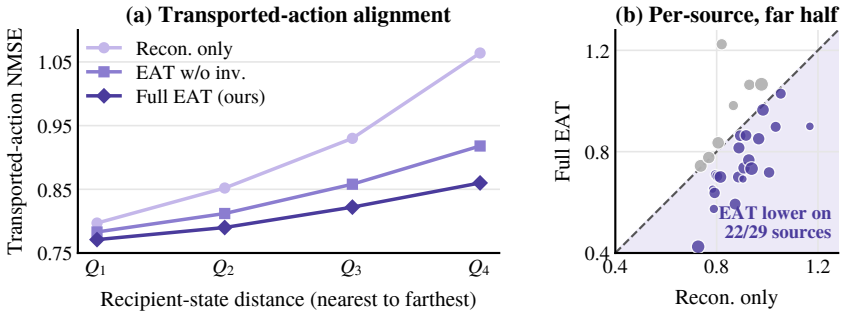


Figure 4: **Cross-context refinement with EAT. (a)** Donor-action NMSE of transported transitions $Y_{j\leftarrow i}$, scored by a reader that is fit only on real transitions and never sees generated ones, so a low value means the transported transition still carries the donor's action. Donor–recipient pairs are grouped by initial-state distance (within-source frozen-DINO cosine quartiles, Q_1 nearest, Q_4 farthest). **(b)** The same NMSE per pretraining source, averaged over the two farthest quartiles, for reconstruction-only vs. full EAT; each point is one of the 29 sources; points below the diagonal favor EAT (22 of 29), and gray points mark the sources where EAT is worse.

the latent behaves across states rather than on the temporal evidence it encodes, and the net-action readout ends below the endpoint-only tokenizer (.671 vs. .703): the endpoint precision traded for A_1 is returned by EAT, not by additional reconstruction.

Transport. To evaluate transport, we fit a linear reader that predicts the recorded action chunk from a real feature-increment sequence, one per data source, and freeze it. We then feed it transported transitions $Y_{j\leftarrow i}$ and score the prediction against the donor's action with the same NMSE as Sec. 4.3. The reader never sees generated transitions, so a low NMSE means the transported transition still carries the donor's action the way a real transition would (details in Appendix B.4).

EAT makes transport substantially more stable as the recipient context moves away from the donor. From the nearest to the farthest quartile, donor-action NMSE rises by .267 under reconstruction-only training, by .136 with EAT without nuisance invariance, and by .087 with full EAT, a three-fold reduction in distance sensitivity (Fig. 4a). The advantage grows with recipient distance, which is exactly the regime EAT targets, and it holds source by source: over the two farthest quartiles, full EAT lowers NMSE on 22 of 29 sources (Fig. 4b). The largest degradation is on roboturk (.822 to 1.224). Recipient-state sensitivity and further transport diagnostics are reported in Appendix C.2.

5 Discussion and Limitations

Our transport objective and metric are defined within a data source, where embodiment and camera geometry are shared; anchoring effects across embodiments would require a correspondence-aware descriptor. The effect space keeps only two low-order temporal components on a coarse spatial grid, which favors robustness over fine temporal detail. The transport metric measures whether donor action semantics survive decoding in another state, not whether the decoded transition matches a unique counterfactual future. Finally, closed-loop results are reported on LIBERO at reduced pretraining scale; larger-scale pretraining and real-robot deployment remain future work.

6 Conclusion

We introduced TERRA, which uses one compact effect space both to define what a latent action keeps and to supervise how it behaves in new contexts. Temporal Effect Representation adds a within-window dynamics component to the endpoint difference, and Effect-Anchored Transport anchors the effect of a latent decoded in other states to the one observed at its source. The resulting latents align more closely with actions, degrade more slowly under visual distractors, and preserve donor actions under transport, and the complete system improves LIBERO success over UniVLA at matched pretraining scale.

AI USE STATEMENT

In this work, we used generative AI tools for drafting, code assistance, experiment orchestration, and language editing. All AI-assisted code and experimental changes were reviewed and tested by the authors, and all quantitative claims were checked against experiment logs and saved artifacts. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations, Workshop Track*, 2017.

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video PreTraining (VPT): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems*, 2022.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2410.24164.

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

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. In *Robotics: Science and Systems*, 2025.

Xizhou Bu, Qingda Hu, Lei Zhou, Lingfeng Zhang, Yingbo Tang, Zihao Liu, Xinyi Tao, Zhiqiang Ma, Qingqiu Huang, Chufeng Tang, Hongbo Wang, Jing Zhang, Jiayi Ma, Hangjun Ye, Wei Li, and Xiaoshuai Hao. What matters for latent actions in robot learning. *arXiv preprint arXiv:2608.19613*, 2026. URL https://arxiv.org/abs/2608.19613.

Guangyan Chen, Meiling Wang, Qi Shao, Zichen Zhou, Weixin Mao, Te Cui, Minzhao Zhu, Yinan Deng, Luojie Yang, Zhanqi Zhang, Yi Yang, Hua Chen, and Yufeng Yue. See once, then act: Vision-language-action model with task learning from one-shot video demonstrations. *arXiv preprint arXiv:2512.07582*, 2025a. URL https://arxiv.org/abs/2512.07582.

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

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, Jianyu Chen, and Jiang Bian. villa-X: Enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025b.

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

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. AdaWorld: Learning adaptable world models with latent actions. In *International Conference on Machine Learning*, 2025.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL https://jmlr.org/papers/v13/gretton12a.html.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284, 2020. URL https://papers.nips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

Yuxin Jiang, Yuchao Gu, Ivor W. Tsang, and Mike Zheng Shou. Olaf-world: Orienting latent actions for video world modeling. In *International Conference on Machine Learning*, 2026.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in Modern Analysis and Probability*, volume 26 of *Contemporary Mathematics*, pp. 189–206. American Mathematical Society, 1984.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems*, 2025a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Jung Min Lee, Taehyun Cho, Li Zhao, and Jungwoo Lee. Why latent actions fail, and how to prevent it. *arXiv preprint arXiv:2605.20223*, 2026.

Qiwei Li, Xicheng Gong, Xinghang Li, Peiyan Li, Quanyun Zhou, Hangjun Ye, Jiahuan Zhou, and Yadong Mu. RotVLA: Rotational latent action for vision-language-action model. *arXiv preprint arXiv:2605.13403*, 2026. URL https://arxiv.org/abs/2605.13403.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2023.

Jiexi Lyu, Xizhou Bu, Qingqiu Huang, Chufeng Tang, Xiaoshuai Hao, Hongbo Wang, and Wei Li. LAFP: Preserving latent action structure in latent policy learning via flow matching. *arXiv preprint arXiv:2606.10517*, 2026. URL https://arxiv.org/abs/2606.10517.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.00379.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *European Signal Processing Conference (EUSIPCO)*, pp. 606–610, 2007.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2312.10812.

Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. CoMo: Learning continuous latent motion from internet videos for scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2505.17006.

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

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn? In *Advances in Neural Information Processing Systems*, 2025.

A Method Details

A.1 Temporal Effect Representation

Structured tokens. The main tokenizer uses five observations, giving $K = 4$ adjacent feature increments. Frozen DINOv2 features have $C = 768$ channels and are pooled to a 4×4 grid ($S = 16$). The structured tokens therefore contain 48 tokens: 16 each from F_0, the net component $N(\Delta F)$, and the dynamics component $A_1(\Delta F)$, distinguished by spatial-position and evidence-type embeddings. A six-layer Transformer aggregates them through four learned latent queries whose outputs are projected to 64 dimensions, giving $z \in \mathbb{R}^{4\times 64}$. The tokenizer queries are distinct from the readout queries used during downstream VLA training and share no parameters with them.

Temporal-evidence basis. For $K = 4$, the first nonconstant DCT-II direction used by the dynamics component has relative coefficients

$$b_1 \propto (0.924,\ 0.383,\ -0.383,\ -0.924).$$

The same coefficient convention is used when constructing the structured tokens and when evaluating the effect descriptor.

Feature-increment normalization. All feature increments are normalized by one fixed global scalar σ_Δ, the root-mean-square magnitude of raw DINOv2 feature increments, estimated once from real training transitions and reused throughout training and evaluation. The decoder predicts normalized increments,

$$\hat{D} = T_\phi(F_0, z) \approx \Delta F / \sigma_\Delta,$$

and transported transitions are converted back to feature units as $Y = \sigma_\Delta \hat{D}$ before the temporal evidence is evaluated on them.

A.2 Effect-Anchored Transport

Effect descriptor. The effect vector $\xi(Y) = \mathrm{vec}[N(Y), A_1(Y)]$ has $2SC = 2 \times 16 \times 768 = 24{,}576$ dimensions. The descriptor first applies a fixed Gaussian random projection $R \in \mathbb{R}^{24576\times 512}$ with entries drawn i.i.d. from $\mathcal{N}(0, 1/512)$ (one draw at initialization with a fixed seed; no gradient), and then standardizes each of the $d = 512$ output coordinates:

$$\Psi(Y) = \frac{R^\top \xi(Y) - \mu}{\sqrt{\sigma^2 + \epsilon}}, \qquad \epsilon = 10^{-5},$$

where $\mu, \sigma^2 \in \mathbb{R}^{512}$ are the per-coordinate mean and variance of $R^\top \xi(\Delta F)$ over real training transitions, estimated once before EAT refinement and then frozen. The standardization is per coordinate (diagonal), not a full whitening: it equalizes the scale of the projected coordinates and does not

decorrelate them. The same transform is applied to real and generated transitions, and Ψ has no learnable parameters.

Because ξ is defined cell by cell on the pooled feature grid and R is a distance-preserving Johnson–Lindenstrauss projection (Johnson & Lindenstrauss, 1984), matching Ψ constrains the per-cell image-space effect of the transported transition up to the projection distortion and the diagonal rescaling; it does not imply equality of a state-independent physical action.

EMA target tokenizer and donor queue. EAT maintains an exponential-moving-average target tokenizer $E_{\bar{\theta}}$, initialized from the online tokenizer and updated at the beginning of each refinement iteration by

$$\bar{\theta} \leftarrow \beta_{\mathrm{EMA}}\bar{\theta} + (1-\beta_{\mathrm{EMA}})\theta, \qquad \beta_{\mathrm{EMA}} = 0.99.$$

Relative-effect consistency uses detached target latents $\bar{z} = \mathrm{sg}[E_{\bar{\theta}}(x)]$ stored in a per-source FIFO queue Q_d holding the most recent 16 minibatches (approximately 1024 latents at batch size 64). Because the sampled donor latents are detached, $\mathcal{L}_{\mathrm{rel}}$ regularizes the shared decoder directly and reaches the online tokenizer only indirectly through the jointly optimized decoder.

Transition-distribution matching. Let $y_p = \chi(Y_{j\leftarrow i}), p = 1, \ldots, m$, denote projected generated increments in a minibatch and $r_q = \chi(\Delta F_i), q = 1, \ldots, n$, projected real increments, where χ is a fixed preprocessing and projection applied identically to both. We use the unbiased squared MMD

$$\mathcal{L}_{\mathrm{dist}} = \frac{1}{m(m-1)}\sum_{p\neq q}\kappa(y_p, y_q) + \frac{1}{n(n-1)}\sum_{p\neq q}\kappa(r_p, r_q) - \frac{2}{mn}\sum_{p,q}\kappa(y_p, r_q),$$

where κ is a mixture of RBF kernels. This term matches transition marginals and does not provide a target counterfactual trajectory for an individual recipient.

Nuisance invariance. The online tokenizer receives a temporally consistent perturbed view while the EMA tokenizer receives the corresponding clean clip. The perturbation consists of a square random crop retaining 90% of the image area (resized back to the input resolution), brightness scaling sampled from $\mathcal{U}(0.8, 1.2)$, and contrast scaling sampled from $\mathcal{U}(0.8, 1.2)$. One augmentation sample is shared by all frames in a clip; no flips, rotations, hue/saturation changes, or per-frame noise are applied. For a 224×224 input, the crop permits at most about 5% translation along either axis, roughly 0.2 of one cell of the 4×4 grid. The augmented view is used only by $\mathcal{L}_{\mathrm{inv}}$; Ψ and both effect-based objectives are evaluated on unaugmented transitions.

Loss weights and schedule. The reported refinement objective uses

$$\lambda_e = 1, \qquad \lambda_r = 0.5, \qquad \lambda_d = 1, \qquad \lambda_i = 1.$$

The representation is first trained for 65k updates with self-reconstruction only and then refined for 15k updates with the full EAT objective while self-reconstruction is retained. After refinement, only the online tokenizer is retained for generating downstream latent-action targets.

Compute. All training uses NVIDIA H100 GPUs. Tokenizer training runs on one GPU: the 65k self-reconstruction stage takes about 7.8 h and the 15k EAT stage about 1.5 h. Latent-VLM pre-training takes about 18 h, and LIBERO fine-tuning takes 5.4–5.6 h per suite on four GPUs; both are identical for TERRA and UniVLA*.

B Experimental Protocol

B.1 Baselines

LAPA-style discrete baseline. The representation experiments use a *LAPA-style* discrete VQ baseline, not the released LAPA tokenizer. We adapt the LAPA-style bottleneck/reconstruction formulation to the same frozen DINO feature space used by TERRA and UniVLA and evaluate its discrete codes with the same frozen linear-reader protocol. The baseline uses NSVQ quantization with a codebook of 8 entries and 4 codes per transition (12 bits), batch size 64, and is trained for 65k updates on the same data, matching the TERRA self-reconstruction stage; its codes are one-hot concatenated before the linear reader.

Retrained UniVLA. UniVLA is retrained at the same latent-action pretraining data scale as TERRA. Its tokenizer follows the official recipe, which inherits the released "Stage-1" task-irrelevant prior; see the provenance note below.

B.2 Downstream VLA Protocol and Provenance

The closed-loop comparison includes latent-VLM pretraining before LIBERO fine-tuning. TERRA and retrained UniVLA both start from OpenVLA-7B and use the same 10% subset of the latent-action pretraining mixture for this stage. All closed-loop numbers use a single fine-tuning seed, 10 tasks per suite, and 50 rollouts per task.

TERRA uses a continuous policy interface: four learned readout embeddings are placed after the image and instruction, and their hidden states condition a flow-matching expert that jointly predicts the continuous latent target and a 12-step, 7-DoF action chunk. UniVLA uses its native discrete latent-token interface. Table 1 therefore compares complete latent-action systems at matched pretraining scale rather than fixing the policy interface.

At inference, TERRA uses only the current observation and language instruction. Future frames, the TERRA tokenizer, the EMA tokenizer, the conditional decoder, and the effect descriptor are not executed online.

Provenance of the retrained UniVLA row. The retrained UniVLA row in Table 1 is not the published UniVLA model. Published UniVLA uses its original larger-scale pretraining protocol, whereas the 91.8% row uses the reduced 10% latent-VLM pretraining mixture described above. The retrained recipe also inherits the small released "Stage-1" task-irrelevant prior required by the official recipe. We therefore report it as a matched-scale system baseline rather than a strict same-provenance-from-random comparison.

Closed-loop perturbation. The rollouts shown above Table 1 are evaluated under the LIBERO-Plus perturbations (Fei et al., 2025) at level L3 (lighting, background texture, and object layout with distractor objects), plus the moving distractor of Appendix B.3, on the 10 tasks of LIBERO-Object and LIBERO-Long with 50 rollouts per task (500 episodes per suite and condition). Perturbed rollouts are used for the qualitative comparison only; the success rates in Table 1 are unperturbed.

B.3 Action Readers and Controlled Nuisance

All representation comparisons freeze the tokenizer. Because the 29 OXE sources use heterogeneous action spaces, we fit a separate linear action reader within each source d and macro-average the source-level normalized mean-squared error

$$\mathrm{NMSE}_d = \frac{\sum_i \|a_i - \hat{a}_i\|_2^2}{\sum_i \|a_i - \bar{a}_d\|_2^2},$$

where $\bar{a}_d$ is the mean action of source d. Readers are ridge regressors. Held-out transitions are split by episode, two thirds for fitting and one third for scoring; the ridge coefficient is selected from $\{10^{-1}, 1, 10, 10^2, 10^3, 10^4\}$ on a 20% episode-level validation split of the fitting portion. The same split and protocol are used for all tokenizers and all five targets of Sec. 4.1.

For nuisance robustness, the underlying robot trajectory and action labels are held fixed while a moving visual distractor is scaled by $\alpha \in \{0, 0.25, 0.5, 1.0\}$. The distractor is a 64×64 natural-video patch pasted opaquely onto the 224×224 frame; it moves along the frame border (inset 32 px) at 4–8 px per frame, with its trajectory and speed determined by the episode id. The severity α scales the distractor magnitude: $\alpha = 0$ is the clean frame and $\alpha = 1$ the fully pasted patch. The same distractor realization is reused across severity levels; only its magnitude changes.

B.4 Transported-Action Evaluation and Context Sensitivity

Transported-action reader. To evaluate transport without training on generated samples, we use a *transported-action reader* $g_{\mathrm{act}} : \mathbb{R}^{K \times S \times C} \to \mathbb{R}^{12 \times 7}$, a linear map from a feature-increment sequence to the 12-step action chunk used in Sec. 4.3. One reader is fit per data source by ridge

Table 3: **95% bootstrap intervals** for the readouts of Table 2, obtained by resampling the 29 sources 2,000 times. Intervals reflect variation across sources, not across training seeds (tokenizer training uses one seed).

Tokenizer	Net	First	Profile	Seq.	Drop (%)
$[F_0 \mid N]$, 65k	.703 [.644, .762]	.790 [.712, .864]	.983 [.964, 1.000]	.800 [.730, .866]	33.6 [24.2, 46.9]
$[F_0 \mid N \mid A_1]$, 65k	.734 [.653, .813]	.767 [.700, .834]	.909 [.875, .942]	.807 [.741, .866]	22.4 [13.7, 33.2]
+15k reconstruction	.732 [.650, .815]	.760 [.692, .828]	.909 [.874, .943]	.807 [.740, .869]	22.6 [14.5, 33.4]
+15k EAT w/o invariance	.700 [.615, .781]	.733 [.663, .806]	.915 [.879, .951]	.784 [.715, .848]	19.7 [12.6, 27.4]
+15k full EAT	.671 [.584, .759]	.717 [.647, .790]	.909 [.878, .940]	.764 [.693, .831]	17.9 [11.1, 24.9]
UniVLA*	.763 [.703, .819]	.778 [.729, .824]	.875 [.836, .911]	.784 [.729, .833]	23.0 [12.5, 34.1]

Table 4: **Cross-context transport diagnostics.** Context sensitivity is $\mathrm{CtxSens}$ from Appendix B.4; latent effective rank is $\exp(-\sum_i p_i \log p_i)$ with $p_i = \sigma_i / \sum_j \sigma_j$ over the singular values σ_i of the centered $4 \times 64 = 256$-dimensional latent matrix on held-out transitions (Roy & Vetterli, 2007).

Diagnostic	65k start	EAT w/o invariance	Full EAT
Context sensitivity	–	.465	.487
Latent effective rank	210.9	–	165.7

regression on *real* held-out pairs $(\Delta F_i/\sigma_\Delta,\, a_i)$ and then frozen. For a transported transition we score

$$\hat{a}_i = g_{\mathrm{act}}(Y_{j \leftarrow i}/\sigma_\Delta)$$

against the recorded donor action a_i with the source-wise NMSE of Sec. 4.1. Because g_{act} never sees generated transitions and reads the increment sequence directly rather than $\Psi(Y)$, a low NMSE means the transported transition carries the donor's action in the same form a real transition would.

Recipient distance. Recipient distance is computed independently of any learned tokenizer. Let

$$c_i = \mathrm{norm}\left(\frac{1}{S}\sum_{s=1}^{S} F_{i,0}^{(s)}\right), \qquad d_{\mathrm{ctx}}(i,j) = 1 - c_i^\top c_j,$$

i.e. the cosine distance between spatially averaged frozen DINO initial-state features. Distances are ranked within each data source and divided into equal-count quartiles; Q_1 contains the nearest 25% of donor–recipient pairs and Q_4 the farthest 25%.

Context sensitivity. To monitor whether the decoder ignores the recipient state, we report

$$\mathrm{CtxSens} = \frac{\|\Psi(T_\phi(F_{j,0}, z)) - \Psi(T_\phi(F_{\ell,0}, z))\|_2}{\|\Psi(T_\phi(F_{j,0}, z))\|_2 + \epsilon},$$

averaged over same-source recipient pairs (j, ℓ). A value near zero would indicate little recipient dependence in the descriptor space.

C Additional Results

C.1 Bootstrap Intervals for Table 2

The same-budget comparison isolates EAT refinement from additional reconstruction training and isolates the nuisance-invariance term; it does not individually isolate $\mathcal{L}_{\mathrm{effect}}$, $\mathcal{L}_{\mathrm{rel}}$, and $\mathcal{L}_{\mathrm{dist}}$.

C.2 Transport Diagnostics

The nonzero context-sensitivity values show that changing the recipient state materially changes the transported descriptor. The full model's value of .487 argues against a state-independent donor replay as the sole explanation for its lower transported-action NMSE.

Table 5: **Per-source transported-action NMSE over the two farthest recipient quartiles** (data of Fig. 4b). n is the number of held-out test clips per source. Full EAT is lower on 22 of 29 sources; the mean over sources is .893 (reconstruction only) vs. .806 (full EAT).

Source	n	Recon.	EAT	Source	n	Recon.	EAT
austin_buds	84	.905	.693	fractal20220817_data	120	.875	.593
austin_sailor	114	1.009	.718	furniture_bench	123	.808	.835
austin_sirius	144	.790	.644	iamlab_cmu_pickup_insert	109	.931	1.065
bc_z	128	.929	.767	jaco_play	124	.801	.708
berkeley_autolab_ur5	140	.729	.425	kuka	126	.917	.865
berkeley_cable_routing	126	.968	.851	language_table	116	.895	.863
berkeley_fanuc_manipulation	120	.888	.699	nyu_franka_play	130	.817	.701
berkeley_gnm_cory_hall	80	1.170	.900	roboturk	105	.822	1.224
berkeley_gnm_recon	128	.985	.966	stanford_hydra	113	.794	.637
berkeley_gnm_sac_son	109	1.055	1.030	taco_play	123	.890	.815
bridge_big	128	.739	.744	toto	108	1.035	.898
cmu_stretch	126	.910	.736	ucsd_kitchen	88	.791	.574
dlr_edan_shared_control	95	.868	.982	utaustin_mutex	122	.770	.777
dobbe	141	.979	1.067	viola	142	.940	.733
fmb	115	.897	.869				

Table 6: **Spatial evidence resolution under matched 256-D compression** (net-action NMSE, lower is better). Nuisance drop is defined in Sec. 4.1.

Grid	Cells	Clean	Distractor	Nuisance drop
16×16	256	.675	.977	93.0%
8×8	64	.658	.860	59.0%
4×4	16	.642	.799	44.0%
2×2	4	.650	.810	45.7%

C.3 Spatial Evidence Resolution

We evaluate spatial evidence resolution under a fixed representation budget. Frozen DINO patch features are pooled to different grids, the same (N, A_1) temporal evidence is constructed, and every representation is PCA-compressed to 256 dimensions before applying the same linear reader. No tokenizer is retrained for this sweep.

Under this matched compression/readout protocol, robustness improves strongly as resolution is reduced to 4×4 and then largely plateaus. Because finer grids undergo stronger PCA compression, this result does not imply that a learned full-patch tokenizer is intrinsically worse; it shows that additional spatial capacity admits nuisance faster than it improves linear action readout under a fixed readout budget.