TIDE: SELECTIVE STATE EXPOSURE FOR LOW-BUDGET ROBOT TRANSFER FROM PASSIVE VIDEO

Anonymous authors
Paper under double-blind review

ABSTRACT

A low-budget manipulation controller benefits from interaction phase, while direct coordinates for the camera motion and edit boundary that produced a video observation need not be action inputs. Transfer Interface with Differentiated Exposure (TIDE) makes this distinction architectural: action-free pretraining learns persistent slot variables, vector-quantized temporal state, egomotion, and a cut-reset gate, while a frozen export map admits only slots and temporal tokens to the grounded policy. A same-backbone intervention tests this partition by fitting a separate action head for each visible coordinate set. With 100 grounded CALVIN episodes and 100 episodes per RLBench single-task head, heads that jointly receive egomotion and cut coordinates attain success lower by 0.119 and 0.125; withholding the temporal token costs 0.180 and 0.158. A matched-GPU-budget retrain without the two acquisition branches also attains success lower by 0.085 and 0.103. TIDE reaches 0.516 ± 0.029 strict five-instruction-chain success on CALVIN and 0.579 ± 0.028 mean success on RLBench, exceeding a compute-, width-, and head-matched action-free V-JEPA transfer by $+0.096$ $[0.071, 0.124]$ and $+0.087$ $[0.061, 0.115]$. The advantage persists with one shared RLBench policy and across 8 ManiSkill2 tasks; Voltron requires $3.5\times$ $[2.9, 4.2]$ as much grounded interaction to reach the same two-suite target. These results establish policy visibility as a separate design axis from predictive representation in the 25–100-episode regime.

1 INTRODUCTION

A controller closing a gripper can require the hidden interaction phase—approach, contact, or release—even when adjacent RGB frames look similar. Withholding that temporal variable from our frozen representation costs 18.0 CALVIN and 15.8 RLBench percentage points. The opposite interface also performs worse: action heads that additionally receive explicit camera-motion and edit-boundary coordinates lose 11.9 and 12.5 points. Passive-video transfer therefore presents a factor-selection problem at the policy boundary, not merely a choice between frame and sequence encoders.

This boundary matters because passive data are abundant while embodiment-specific actions remain expensive (Walke et al., 2023; Open X-Embodiment Collaboration, 2024; Reed et al., 2022). Egocentric and tutorial video expose contact, object transformation, and task progress, but entangle them with wearer motion, shot changes, and human embodiment. A frame representation discards both acquisition history and useful interaction progress. A monolithic sequence representation preserves both and leaves a small grounded dataset to discover which coordinates should drive action. The missing technical object is an explicit map from the state useful for video prediction to the state readable by control.

Transfer Interface with Differentiated Exposure (TIDE) supplies that map. Action-free pretraining maintains persistent slot variables and vector-quantized temporal state together with six-degree-of-freedom egomotion and an observed cut gate that resets temporal state. Slots and temporal tokens form a 1152-dimensional policy input; egomotion and cut coordinates remain direct-policy-invisible while still shaping slot transitions and resets. Grounding freezes this entire video model and trains only a 0.37M-parameter direct-action head. The transfer boundary is consequently a specified part of the model rather than a feature-selection decision delegated to tens of robot episodes.

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.

The decisive evidence is a factor-by-factor intervention within one pretrained backbone. Relative to TIDE, removing the temporal token from the exported state lowers success by 0.180/0.158 on CALVIN/RLBench. Starting from the slot–token interface, exposing only the cut coordinate lowers success by 0.067/0.067, exposing only egomotion lowers it by 0.054/0.053, and exposing both lowers it by 0.119/0.125. A matched-budget retrain without the egomotion and cut branches loses 0.085/0.103. This three-sided result supports temporal-token export, direct omission of acquisition coordinates, and a richer internal state than the policy-visible state.

A separate matched comparison tests whether the intervention translates into an end-to-end advantage over generic temporal transfer. V-JEPA (Bardes et al., 2024) is pretrained from scratch on the same clips, action-free supervision, optimization steps, and GPU-hours, then connected to the same input width and action head. TIDE improves strict CALVIN chain success by $+0.096$ $[0.071, 0.124]$ and mean RLBench success by $+0.087$ $[0.061, 0.115]$. The ordering remains under one shared RLBench policy and on 8 ManiSkill2 tasks, with the largest gaps between 25 and 100 grounded episodes; Voltron needs $3.5\times$ $[2.9, 4.2]$ as much interaction to reach the same two-suite target.

The primary contribution is the predictive-to-control export map and its same-backbone exposure test. The supporting algorithm factorizes interaction and acquisition variables so that this intervention is possible, while the matched temporal baseline, branch-removal retrain, budget sweep, shared policy, and cross-simulator results establish its consequence: policy visibility changes grounded-data efficiency independently of the amount of state learned for video prediction.

2 Related Work

Passive-video representations for control. R3M (Nair et al., 2023), MVP (Radosavovic et al., 2023), VIP (Ma et al., 2023), Voltron (Karamcheti et al., 2023), and masked motor-control pretraining (Xiao et al., 2022) establish passive video as a source of transferable visual features. More recent systems make motion the transfer object: LAPA learns discrete latent actions for VLA fine-tuning (Ye et al., 2025); Moto autoregresses latent motion tokens and co-fine-tunes them with robot control (Chen et al., 2025); AMPLIFY predicts keypoint-motion tokens for an inverse-dynamics model (Collins et al., 2025); ViPRA jointly predicts future observations and motion-centric latents before decoding continuous action chunks (Routray et al., 2026); and CoMo uses continuous latent motion as pseudo-actions for policy co-training (Yang et al., 2026). These methods ask how to construct and translate motion state. TIDE instead makes policy visibility within a richer predictive state directly intervenable: the same frozen backbone can export interaction factors while withholding explicit acquisition coordinates. Appendix B compares these transfer interfaces dimension by dimension.

Action-free temporal prediction. V-JEPA predicts masked latent video targets without actions, text, pixel reconstruction, or negative pairs (Bardes et al., 2024); V-JEPA 2 scales the objective and adds an action-conditioned world model for planning after video pretraining (Assran et al., 2025). We instantiate the original action-free formulation on the TIDE corpus and compute budget, then match frozen readout width and grounded-head architecture. This isolates partitioned object–skill export from one controlled generic sequence readout. Action-conditioned world models (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2025; Seo et al., 2023; Wu et al., 2023), interactive video models (Bruce et al., 2024; Yang et al., 2024), and sequence control (Chen et al., 2021) use action or interaction signals during representation learning and address a different supervision boundary.

Structured temporal state. Slot Attention and object-centric video models provide permutation-aware scene decomposition (Locatello et al., 2020; Burgess et al., 2019; Greff et al., 2019; Kipf et al., 2022); skill-prior methods provide temporally abstract codes (Pertsch et al., 2021; 2022; Shankar & Gupta, 2020; Lynch et al., 2020); and recurrent modular dynamics motivate factorized transitions (Goyal et al., 2021). TIDE uses these ingredients to define a different technical object: an explicit export map from predictive state to control state. Prior methods do not report a same-backbone intervention that withholds named acquisition coordinates while preserving their internal predictive roles.

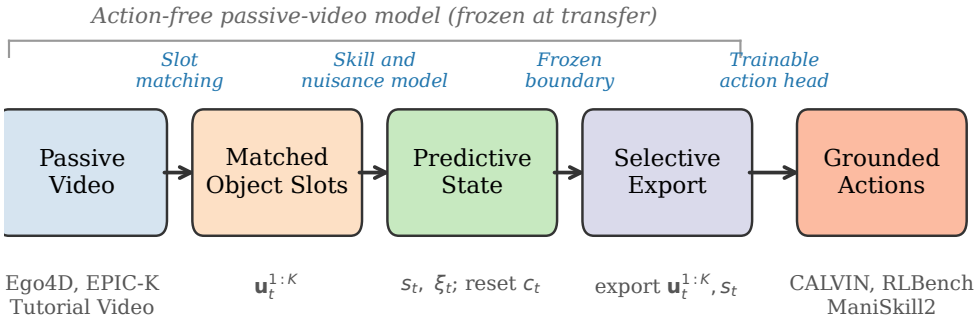

Figure 1: TIDE separates video-predictive state from policy-visible state. Matched object slots and skill tokens cross the frozen boundary; egomotion and the observed cut-reset gate organize pretraining internally. Grounding trains only the direct-action head.

3 TIDE: Predictive State with Selective Exposure

3.1 Problem and Interface

The pretraining corpus is an action-free set of videos $\mathcal{V} = \{V_i\}_{i=1}^N$, with $V_i = (\mathbf{x}_1^i, \dots, \mathbf{x}_{T_i}^i)$ and RGB observation $\mathbf{x}_t \in \mathbb{R}^{H\times W\times 3}$. Grounding uses embodiment-specific trajectories $\mathcal{D}_r = \{\tau_i\}$, where $\tau_i = ((\mathbf{x}_t, \mathbf{g}_t, \mathbf{a}_t, \mathbf{x}_{t+1}))_{t=1}^{T_i}$ preserves task codes and episode boundaries in the simulator-native action space. TIDE separates the latent variables used to model within-shot dynamics from the subset made directly readable by the action head.

Figure 1 summarizes the information path. For each observation, TIDE constructs a predictive latent

$$\mathbf{z}_t = (\mathbf{z}_t^{\text{ctrl}}, \mathbf{z}_t^{\text{acq}}) = \big((\mathbf{u}_t^{1:K}, s_t), (\xi_t, c_t)\big), \tag{1}$$

where $\mathbf{u}_t^k \in \mathbb{R}^{d_{\text{slot}}}$ is object slot k, s_t is a discrete skill token, $\xi_t \in \mathbb{R}^6$ is egomotion, and $c_t \in [0,1]$ is a cut probability. We use $\bar{\mathbf{z}}_t = ((\mathbf{u}_t^{1:K}, s_t), \xi_t)$ for the within-shot transition latent and $g_t = \mathbb{1}[c_t > 0.5]$ for the separately observed reset gate; c_t is interface metadata, not a target predicted from $\bar{\mathbf{z}}_{t-1}$. The export rule is:

Component	Pretraining role	Grounding status
Slots and skill tokens	interaction state	frozen and exported
Egomotion	acquisition state	frozen and internal
Observed cut gate	recurrent segmentation	frozen and internal
Transition modules	temporal prediction	frozen and internal
Direct-action head π_ϕ	embodiment mapping	trainable

Thus the export map is fixed as $\mathcal{I}_{\text{TIDE}}(\mathbf{z}_t) = (\mathbf{u}_t^{1:K}, s_t)$. Grounding cannot directly read ξ_t or c_t because neither variable enters the policy computation graph; changing their visibility requires an explicit intervention on the interface. "Internal" refers only to direct readout: acquisition variables can still affect exported slots and skills through transition conditioning and recurrent resets.

3.2 Slots, Skills, and Acquisition State

Persistent object slots. A Slot Attention encoder (Locatello et al., 2020) maps $\mathbf{x}_t$ to K vectors and spatial soft masks M_t^k. A spatial-broadcast decoder anchors these variables by reconstructing the frame as $\hat{\mathbf{x}}_t = \sum_k M_t^k D(\mathbf{u}_t^k)$; $\mathcal{L}_{\text{rec}}$ combines an ℓ_1 reconstruction term with mask-overlap, entropy, and balanced-coverage penalties. Reconstruction discourages a constant-state solution, while coverage discourages persistent empty slots. At $t > 1$, the previous posterior initializes the slots. We align current and previous slots by Hungarian matching on mask intersection-over-union plus feature

cosine similarity. A newly visible object occupies the best unused learned anchor; a disappearing object is represented by a masked null slot whose transition loss is zero. Let v_t^k be one for an active slot and zero for this null slot. For multi-step prediction, the target assignment is recomputed at every horizon, so

$$\mathcal{L}_{\text{slot}} = \sum_{h=1}^{H} \min_{\sigma \in S_K} \sum_{k=1}^{K} v_{t+h}^{\sigma(k)} \left\| \hat{\mathbf{u}}_{t+h}^{k} - \text{sg}(\mathbf{u}_{t+h}^{\sigma(k)}) \right\|_2^2, \tag{2}$$

This makes the prediction loss invariant to target-slot permutations at each horizon.

Action-free skill discovery. A temporal encoder consumes the $W = 4$ preceding transitions, represented by the five matched slot sets $\mathbf{u}_{t-W:t}^{1:K}$ within the current shot, and emits e_t. Episode starts and observed cuts restart this history from the initialized slot state. The token is the nearest entry in a learned VQ codebook $\{q_j\}_{j=1}^{M}$ (van den Oord et al., 2017):

$$s_t = \arg\min_j \|e_t - q_j\|_2, \qquad \tilde{e}_t = e_t + \text{sg}(q_{s_t} - e_t). \tag{3}$$

The straight-through vector $\tilde{e}_t$ conditions the transition. The exact VQ term is

$$\mathcal{L}_{\text{VQ}} = \|\,\text{sg}(e_t) - q_{s_t}\|_2^2 + \beta \|e_t - \text{sg}(q_{s_t})\|_2^2, \qquad \beta = 0.25. \tag{4}$$

Exponential-moving-average code usage, a batch entropy term, and reinitialization of unused entries prevent token collapse. Consecutive identical assignments define a skill segment, and a token-conditioned log-normal duration model encourages persistence without forcing a fixed duration.

Egomotion and cut semantics. A convolutional pair encoder estimates depth and ξ_t from $(\mathbf{x}_{t-1}, \mathbf{x}_t)$. Its self-supervised geometry loss $\mathcal{L}_{\text{ego}}$ photometrically warps the previous frame into the current frame using the estimated depth and SE(3) transform, with forward–backward flow masking for occlusion and moving foregrounds. This supplies photometric self-supervision for the six-parameter transform; the branch outputs are learned coordinates rather than asserted metric camera pose. A separate classifier produces c_t from the same observed pair. After $\mathbf{x}_t$ arrives, the pair modules compute (ξ_t, c_t); at an episode start or when $g_t = 1$, recurrent, slot-matching, token-window, and duration state reset before slots for $\mathbf{x}_t$ are computed. The post-reset causal prefix then produces s_t and $\mathbf{a}_t$, without access to $\mathbf{x}_{t+1}$.

3.3 Transitions and grounded actions

Within a continuous shot, a graph network predicts interactions among matched slots:

$$\hat{\mathbf{u}}_{t+1}^{k} = \mathbf{u}_t^k + \text{GNN}_\theta\Big(\mathbf{u}_t^k, \{\mathbf{u}_t^j\}_{j \neq k}, \xi_t, \tilde{e}_t\Big). \tag{5}$$

The within-shot transition and observed boundary gate are combined as

$$p_\theta(\bar{\mathbf{z}}_{t+1} \mid \bar{\mathbf{z}}_t, g_{t+1}) = \begin{cases} p_\theta(\mathbf{u}_{t+1}^{1:K}, \xi_{t+1}, s_{t+1} \mid \bar{\mathbf{z}}_t), & g_{t+1} = 0, \\ p_0(\mathbf{u}_{t+1}^{1:K}, \xi_{t+1}, s_{t+1}), & g_{t+1} = 1, \end{cases} \tag{6}$$

where p_0 initializes a new shot. This is a segmented transition objective, not a prediction of g_{t+1} from $\bar{\mathbf{z}}_t$: during pretraining, g_{t+1} is obtained from the observed pair $(\mathbf{x}_t, \mathbf{x}_{t+1})$, and a transition entering a new shot incurs no within-shot slot-prediction term. During online grounding, the analogous gate g_t is computed only after $\mathbf{x}_t$ becomes the current observation and resets state before action $\mathbf{a}_t$ is produced.

Robot observations pass through the same frozen RGB, slot-matching, temporal, egomotion, and cut modules. The internal modules may reset state, but their outputs never enter the policy vector. At an episode start or detected cut, p_0 initializes the recurrent state; thereafter every slot and skill estimate uses only the causal prefix through time t. Let $\mathbf{g}_t$ denote the benchmark-provided task code shared by every method: a 34-way one-hot instruction code for CALVIN, a ten-way code for the shared RLBench policy, and a constant omitted from single-task heads. The grounding head is

$$\mathbf{a}_t = \pi_\phi\big([\mathbf{u}_t^1; \ldots; \mathbf{u}_t^K; q_{s_t}; \mathbf{g}_t]\big), \tag{7}$$

a three-layer MLP with 256 hidden units and ReLU activations. Thus the policy consumes the 128-dimensional codebook embedding q_{s_t}, rather than an integer token index. Behavioral cloning trains only ϕ. The asymmetric information path is therefore structural, rather than a post-hoc probe or feature-selection heuristic.

4 PRETRAINING

The action-free objective is

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{slot}} + 0.5\mathcal{L}_{\text{temp}} + 0.1\mathcal{L}_{\text{persist}} + 0.2\mathcal{L}_{\text{ego}} + \mathcal{L}_{\text{cut}}, \tag{8}$$

where $\mathcal{L}_{\text{temp}} = -\log p(s_{t+1} \mid s_t, \mathbf{u}_{t-W:t}^{1:K}) + \mathcal{L}_{\text{VQ}}$, $\mathcal{L}_{\text{persist}}$ fits token durations, and $\mathcal{L}_{\text{cut}}$ is binary cross-entropy on automatically generated cut labels. For adjacent frames, let d_{app} be one minus the intersection of 32-bin HSV histograms and let r_{flow} be the fraction of pixels whose forward–backward flow error is within 1.5 pixels after best-homography compensation. The cut score is $d_t = 0.6d_{\text{app}} + 0.4(1 - r_{\text{flow}})$; to capture dissolves, labeling uses $\bar{d}_t = \max\{d_t, (d_{t-1} + d_t)/2\}$. Pairs with $\bar{d}_t \geq 0.62$ are positive, those with $\bar{d}_t \leq 0.28$ are negative, and the ambiguous middle band is excluded. Source-file boundaries are never sampled as adjacent frames. Cut minibatches use a 1:3 positive-to-negative ratio with inverse-frequency weighting. Reconstruction anchors object content, reprojection identifies camera motion, and the matched multi-step term learns object dynamics. Prediction uses horizon $H = 8$.

The pre-dedup candidate mixture contains 2,800 hours of interaction-heavy Ego4D (Grauman et al., 2022), 100 hours of EPIC-KITCHENS-100 (Damen et al., 2022), and 770 hours of curated public tutorial video spanning cooking, assembly, and craft. Physical-interaction filtering selects hand–object contact. Deduplication retrieves candidates by pHash and then measures their overlap with evaluation references using cosine similarity between normalized CLIP (Radford et al., 2021) clip embeddings:

$$o(c) = \max_{r \in \mathcal{R}_{\text{eval}}} \text{sim}_{\text{CLIP}}(c, r). \tag{9}$$

We remove a candidate when $o(c) \geq 0.92$. The resulting 3516 hours form the common corpus for TIDE and the matched V-JEPA run.

Training uses 8 A100 80 GB GPUs with CUDA 12.4 for 200K AdamW steps, a batch of 128 clips, 32 frames per clip at 5 fps, and 576 GPU-hours. The learning rate peaks at 3×10^{-4} after 5K warmup steps and follows cosine decay. We use $K = 8$ slots of width 128 and an $M = 512$ skill codebook.

5 EXPERIMENTS

The experiments follow the hidden factors through the transfer boundary. First, within one frozen TIDE backbone, which factors does control lose when they are withheld, and which factors become harmful when directly revealed? Second, does the resulting partition outperform a compute-, width-, and head-matched generic temporal export? Third, at what grounding budgets, task sharing levels, and simulators does that advantage persist?

5.1 PROTOCOL AND COMPARATORS

Suites. CALVIN (Mees et al., 2022) uses a task-code-conditioned ABC-D protocol in environment D. At each reported budget, the grounding set contains that many episodes across the training tasks, and each seed executes 1,000 benchmark-sampled sequences of five ordered instructions over the 34 task classes, with at most 360 control steps per instruction. The active instruction's 34-way benchmark code is supplied to every policy; our primary endpoint is a strict binary chain score that equals one only when all five instructions finish in order. Every method is evaluated with this endpoint; published average-sequence-length results use a different denominator and are not mixed into our comparisons. RLBench (James et al., 2020) uses ten named tabletop tasks. Its single-task protocol trains one head from 100 grounded episodes per task; its shared-policy protocol pools the same per-task budget into one task-conditioned head. Each task contributes 100 evaluation episodes per seed. ManiSkill2 (Gu et al., 2023) adds 8 rigid and articulated-object tasks, each with 100 grounded and 100 evaluation episodes per seed. All benchmark comparisons use 7 independent grounding seeds.

Matched sequence comparator. V-JEPA is trained from scratch on the same passive clips, frame sampler, augmentation stream, optimization steps, exported width, and GPU-hour budget as TIDE. It uses its published masked feature-prediction objective and receives no action labels. We spatially pool patch tokens per frame, fit a 128-dimensional PCA projection on the passive-video training partition, and freeze that projection before grounding. At action time, its buffer contains the available

causal prefix up to the latest 32 frames; adaptive temporal pooling maps that prefix into 9 ordered 128-dimensional tokens without access to future observations. Concatenation yields a frozen 1152-dimensional visual input, exactly matching eight 128-dimensional TIDE slots plus q_{s_t}. The same task code and identical three-layer head are appended in both methods. This comparator matches the absence of action labels, passive data, optimization compute, exported width, and downstream capacity; its feature-prediction objective and factorization differ by design.

Frame and policy comparators. VIP, Voltron, R3M, and MVP use their published pretrained encoders, frozen under the same suite-specific RGB view and single-frame preprocessing. Their native embeddings feed the common three-layer 256-unit head with identical optimization, action normalization, grounded episodes, and training-only augmentation within each suite. These rows test established frame semantics under a common grounding recipe; V-JEPA supplies the tightly compute-, width-, and head-matched temporal comparison. Voltron follows the language-conditioned pretraining method of Karamcheti et al. (2023). A same-head pixel behavioral-cloning control and Conservative Q-Learning (Kumar et al., 2020; Levine et al., 2020) appear in Appendix A.1.

Uncertainty. Tables report mean $\pm$ standard deviation over grounding seeds. Paired differences use a hierarchical percentile bootstrap with the complete grounding replicate as the paired unit: each replicate first resamples paired seed indices, then independently resamples evaluation rollouts inside each selected method–seed cell. No episode-identical trajectory pairing is assumed. We use 10,000 bootstrap replicates. CALVIN contributes 7,000 evaluation rollouts per method; single-task RLBench contributes 7,000 episodes across its ten tasks. Hyperparameters and checkpoint rules are fixed before the paired evaluation.

5.2 Which hidden factors should cross the boundary?

Table 1: Factor-by-factor policy visibility at 100 episodes. Width precedes the task code. Each of the first five rows fits its own head on one frozen backbone with identical splits, seeds, optimizer, and checkpoint rule; only the visible coordinate set changes. The final row retrains the predictor for the same 576 GPU-hours without acquisition branches.

Interface	Exported to policy	Kept internal	CALVIN	RLBench
Slots only	slots (1024)	skill, ego, cut	0.336 ± 0.035	0.421 ± 0.037
TIDE	slots, skill (1152)	ego, cut	0.516 ± 0.029	0.579 ± 0.028
Add cut coordinate	slots, skill, cut (1153)	ego	0.449 ± 0.039	0.512 ± 0.038
Add egomotion coordinates	slots, skill, ego (1158)	cut	0.462 ± 0.036	0.526 ± 0.037
Expose all	slots, skill, ego, cut (1159)	none	0.397 ± 0.037	0.454 ± 0.040
No acquisition branches	slots, skill (1152)	branches absent	0.431 ± 0.038	0.476 ± 0.041

Table 1 follows each factor through the interface. Removing the policy-visible skill embedding loses 0.180 on CALVIN and 0.158 on RLBench. From the TIDE interface, directly appending cut lowers success by 0.067/0.067, appending egomotion lowers it by 0.054/0.053, and appending both lowers it by 0.119/0.125; every paired interval excludes zero. These rows preserve the backbone and hidden-layer architecture. Expose-all changes only seven inputs and 1,792 first-layer weights (less than 0.5% of the 0.37M-parameter head), while the single-factor rows add 256 or 1,536 weights; the lower-dimensional slots-only head performs worst, excluding a monotone input-dimension advantage. Finally, removing both acquisition branches during matched-budget pretraining loses 0.085/0.103. The complete pattern favors a predictor that models acquisition state, exports interaction phase, and blocks direct acquisition coordinates.

5.3 Does the partition improve matched temporal transfer?

Table 2 tests the partition against a controlled generic temporal readout. TIDE improves over V-JEPA by $+0.096$ $[0.071, 0.124]$ on CALVIN and $+0.087$ $[0.061, 0.115]$ on RLBench; both paired 95% intervals exclude zero. It exceeds Voltron by 0.192 and 0.172 under the shared grounding recipe. Interpolating each suite independently in log budget and taking the larger requirement across the two TIDE 100-episode targets, Voltron needs $3.5\times$ $[2.9, 4.2]$ as many demonstrations and VIP needs

Table 2: Frozen passive-video transfer at 100 grounded episodes in total for CALVIN and 100 per RLBench single-task head. Values are mean ± standard deviation over 7 grounding seeds; counts aggregate evaluation episodes. V-JEPA matches TIDE in passive clips, optimization compute, exported width, and action-head architecture.

Method	CALVIN	CALVIN successes	RLBench	RLBench successes
VIP	0.302 ± 0.034	2114/7000	0.381 ± 0.036	2667/7000
Voltron	0.324 ± 0.038	2268/7000	0.407 ± 0.040	2849/7000
V-JEPA, ordered temporal state	0.420 ± 0.035	2940/7000	0.492 ± 0.034	3443/7000
TIDE, selective export	0.516 ± 0.029	3612/7000	0.579 ± 0.028	4053/7000

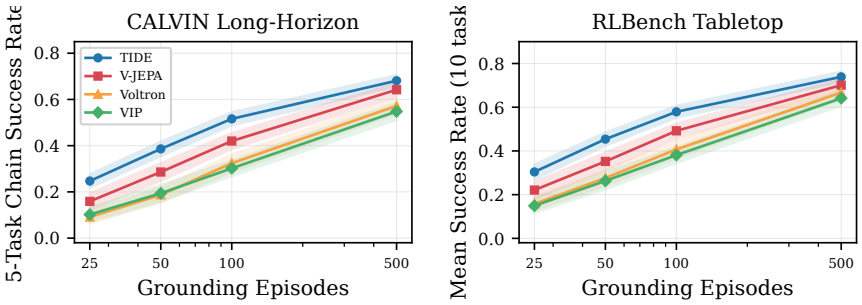

Figure 2: Grounding efficiency on CALVIN and RLBench. Curves show means with one-standard-deviation bands over 7 seeds. TIDE separates most strongly from V-JEPA in the scarce-grounding regime; the margin narrows as additional grounded supervision reduces sensitivity to the exported interface.

$4.1\times$ $[3.4, 4.9]$. The export choice therefore changes grounded-data cost as well as 100-episode success.

Figure 2 shows that the main-table margin emerges before 100 episodes on both suites. At 25 episodes, TIDE improves over V-JEPA by 0.088 on CALVIN and 0.083 on RLBench; at 50 episodes the gains are 0.100 and 0.102. All methods improve with more grounding, and the margins contract to 0.039 and 0.038 at 500 episodes. Selective exposure therefore changes the data-efficiency curve most in the regime for which passive-video reuse is intended. Appendix C reports every mean and dispersion.

5.4 Budget, Task Sharing, and Simulator Breadth

Table 3: Cross-task and cross-simulator transfer. RLBench trains one shared task-conditioned head; ManiSkill2 changes simulator, object families, and controller. Values are mean ± standard deviation over 7 seeds.

Method	RLBench shared	ManiSkill2
VIP	0.347 ± 0.038	0.296 ± 0.039
Voltron	0.361 ± 0.041	0.314 ± 0.041
V-JEPA	0.447 ± 0.037	0.389 ± 0.037
TIDE	0.542 ± 0.032	0.468 ± 0.034

Table 3 shows that the shared RLBench policy preserves a 0.095 TIDE advantage over V-JEPA despite cross-task interference. On ManiSkill2, TIDE leads V-JEPA by $+0.079$ $[0.052, 0.107]$ over 5,600 episodes per method and records 2621/5600 successes, compared with 2178/5600, 1758/5600, and 1658/5600. Appendix E shows that contact-rich and articulated-object tasks both contribute to the aggregate, so the gain is not concentrated in one interaction type.

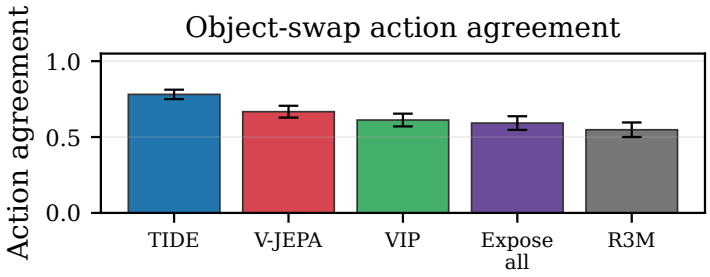

Figure 3: Action-change agreement under 2,400 input-level object swaps per method. Error bars are trajectory-bootstrap 95% intervals. The target is the action difference between the paired source and donor observations; this is an input-sensitivity diagnostic, not a simulator-state intervention.

The diagnostic in Figure 3 replaces one simulator-segmented object region with its geometrically aligned counterpart from a held-out observation, reruns each full encoder and policy, and leaves the remaining pixels fixed. This common RGB perturbation does not require baseline-specific slots. TIDE attains 0.781 ± 0.031 agreement, versus 0.667 ± 0.039 for V-JEPA, 0.612 ± 0.042 for VIP, 0.592 ± 0.045 for expose-all, and 0.548 ± 0.048 for R3M. The paired TIDE gaps are $+0.114\ [0.087, 0.142]$ to V-JEPA and $+0.189\ [0.157, 0.221]$ to expose-all. This evidence concerns action sensitivity to an object-region swap; the policy-success interventions provide the direct evidence about the transfer boundary. Appendix F specifies the agreement rule.

6 Discussion

The experiments distinguish predictive utility from direct policy utility. Skill state is necessary on the visible side of the interface; egomotion and cut-reset coordinates are useful inside the video model but harmful when appended to a scarce-data action head; and the resulting partition outperforms a compute-, width-, and head-matched generic sequence export. This factor-by-factor pattern, rather than the external ranking alone, is the evidence for a richer predictor with a selective control path.

The effect follows the intended data regime: selective export yields its largest savings from 25 to 100 episodes, remains ordered at 500 episodes, survives one shared policy, and transfers across simulator and controller changes. Direct omission does not require statistical invariance—resets and transition conditioning can mediate acquisition information—and each exposure intervention changes at most seven inputs and 0.5% of head parameters. The three simulated suites establish the value of the specified direct interface for frozen-backbone, task-code-conditioned control; evaluations under unseen acquisition processes and physical cameras will determine how far the same boundary transfers beyond this regime.

7 Conclusion

TIDE assigns passive-video prediction and robot control distinct views of one learned process: prediction uses objects, skills, egomotion, and observed edit resets, whereas control directly reads persistent objects and interaction progress. At 100 episodes, this design gains 9.6 CALVIN and 8.7 RLBench percentage points over matched temporal export; revealing both acquisition coordinates within the same backbone costs 11.9/12.5 points. The advantage persists under shared-policy training and across three manipulation suites, while Voltron requires $3.5\times\ [2.9, 4.2]$ as much interaction to reach the same two-suite target. Selective exposure turns the transfer interface itself into a source of measurable data efficiency.

AI Use Statement

Generative AI was used only for manuscript writing, editing, and presentation. The authors reviewed all such edits and remain responsible for the final content.

Reproducibility Statement

Sections 3–5.1 specify the factorization, export map, objectives, data mixture, pretraining optimization, causal action-time histories, heads, benchmarks, and uncertainty procedure. Appendix A gives split, seed, checkpoint, baseline, compute, and evaluation details; Appendices C–F provide the numeric breakdowns behind the main results.

References

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. doi: 10.48550/arXiv.2506.09985. URL https://arxiv.org/abs/2506.09985.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2. Featured Certification.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. URL https://arxiv.org/abs/1901.11390.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15084–15097, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

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 (ICCV)*, pp. 19752–19763, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Chen_Moto_Latent_Motion_Token_as_the_Bridging_Language_for_Learning_ICCV_2025_paper.html.

Jeremy A. Collins, Loránd Cheng, Kunal Aneja, Albert Wilcox, Benjamin Joffe, and Animesh Garg. AMPLIFY: Actionless motion priors for robot learning from videos. *arXiv preprint arXiv:2506.14198*, 2025. doi: 10.48550/arXiv.2506.14198. URL https://arxiv.org/abs/2506.14198.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. doi: 10.1007/s11263-021-01531-2.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mLcmdlEUxy-.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Grauman_Ego4D_Around_the_World_in_3000_Hours_of_Egocentric_Video_CVPR_2022_paper.html.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2424–2433. PMLR, 2019. URL https://proceedings.mlr.press/v97/greff19a.html.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=b_CQDy9vrD1.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. doi: 10.1109/LRA.2020.2974707.

Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems*, 2023. URL https://roboticsconference.org/2023/program/papers/032/.

Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aD7uesX1GF_.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. URL https://arxiv.org/abs/2005.01643.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/8511df98c02ab60aea1b2356c013bc0f-Abstract.html.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Proceedings of the 3rd Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1113–1132. PMLR, 2020. URL https://proceedings.mlr.press/v100/lynch20a.html.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=YJ7o2wetJ2.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. doi: 10.1109/LRA.2022.3180108.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909. PMLR, 2023. URL https://proceedings.mlr.press/v205/nair23a.html.

Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models. In *IEEE International Conference on Robotics and Automation*, 2024. URL https://arxiv.org/abs/2310.08864.

Karl Pertsch, Youngwoon Lee, and Joseph J Lim. Accelerating reinforcement learning with learned skill priors. In *Proceedings of the 4th Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 188–204. PMLR, 2021. URL https://proceedings.mlr.press/v155/pertsch21a.html.

Karl Pertsch, Youngwoon Lee, Yue Wu, and Joseph J Lim. Demonstration-guided reinforcement learning with learned skills. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 729–739. PMLR, 2022. URL https://proceedings.mlr.press/v164/pertsch22a.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 416–426. PMLR, 2023. URL https://proceedings.mlr.press/v205/radosavovic23a.html.

Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=1ikK0kHjvj.

Sandeep Kumar Routray, Hengkai Pan, Unnat Jain, Shikhar Bahl, and Deepak Pathak. ViPRA: Video prediction for robot actions. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/707e34efcabd2b9375f7a64019600aa8-Abstract-Conference.html.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1332–1344. PMLR, 2023. URL https://proceedings.mlr.press/v205/seo23a.html.

Tanmay Shankar and Abhinav Gupta. Learning robot skills with temporal variational inference. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8624–8633. PMLR, 2020. URL https://proceedings.mlr.press/v119/shankar20b.html.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1723–1736. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2226–2240. PMLR, 2023. URL https://proceedings.mlr.press/v205/wu23c.html.

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

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 (CVPR)*, pp. 42352–42363, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Yang_CoMo_Learning_Continuous_Latent_Motion_from_Internet_Videos_for_Scalable_CVPR_2026_paper.html.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5O9JBt35zg.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sV4Y6i1lvE.

A Protocol and implementation details

Splits and seeds. Source videos, rather than overlapping clips, are assigned to pretraining, validation, and internal test partitions in a fixed 94/3/3% split. Grounded episodes use a fixed 90/10 train/validation split within each budget; benchmark evaluation rollouts are generated independently. A seed is a deterministic integer hash of suite, task, and replicate index. The replicate index is paired across methods, while data-loader, initialization, and simulator randomness are independent streams derived from that seed.

Selection and statistics. Architecture and optimizer choices are selected once using pretraining validation loss and the first CALVIN grounding-validation split, then frozen across suites and methods. Grounding checkpoints maximize validation action likelihood, with the earliest checkpoint breaking ties. Mean and standard deviation treat a complete grounding run as the unit. The hierarchical bootstrap pairs methods by replicate, resamples replicate indices, and independently resamples evaluation episodes within each selected method–replicate cell. All reported intervals are two-sided percentile intervals.

A.1 Baseline definitions and additional controls

V-JEPA follows its published target-encoder feature-prediction loss. Its readout spatially pools patch tokens within each frame, applies the frozen training-split PCA projection, adaptively pools time into

Table 4: Manuscript-level specification of the modules and optimization choices used in the reported experiments.

Block	Architecture or state	Training role
Object state	Slot Attention; $K = 8$ slots of width 128; spatial-broadcast decoder; recurrent initialization and Hungarian temporal matching	Frame reconstruction plus visibility-masked, permutation-matched prediction through horizon $H = 8$
Skill state	Four-transition matched-slot-difference encoder; 512-entry VQ codebook; 128-dimensional exported embedding	Straight-through VQ with commitment weight 0.25, batch usage entropy, exponential-moving-average usage tracking, unused-code reinitialization, and duration likelihood
Acquisition state	Convolutional pair encoder for depth and six-DoF egomotion; separate pairwise cut classifier and observed reset gate	Photometric reprojection with flow masking; cut-label cross-entropy; transition conditioning and recurrent segmentation; no direct TIDE policy input
Transition	Residual interaction GNN over matched slots, conditioned on skill and egomotion	Matched multi-step slot prediction inside each observed shot
Pretraining	32-frame clips at 5 fps; batch 128; 200K AdamW steps; 5K-step warmup to 3×10^{-4} and cosine decay	Loss weights: reconstruction 1, slot 1, temporal 0.5, persistence 0.1, egomotion 0.2, cut 1; 576 GPU-hours
Grounding	Frozen video model; concatenated slots, 128-dimensional skill embedding, and benchmark task code; three-layer 256-unit ReLU MLP	Behavioral cloning of simulator-native actions; only the action head is optimized; validation action likelihood selects checkpoints

nine ordered bins, and concatenates the bins without averaging across time. The nine 128-dimensional bins match TIDE's eight slot vectors plus one 128-dimensional skill embedding before the shared task code is appended. Voltron, VIP, R3M, and MVP export their native frozen embedding from the current RGB frame; these external-reference rows share the head architecture and training recipe but retain their published representation widths. Pixel BC receives the same resized RGB observations and trains a small convolutional stem jointly with the common action head; it has no passive-video pretraining. CQL uses the same grounded transition set and action representation, with conservative coefficient selected on the fixed validation split. All policies receive the same benchmark task code. The slots-only control runs every pretrained module but withholds skill and acquisition coordinates; the no-acquisition-branches control uses the same 576 GPU-hour budget to retrain without egomotion and cut branches.

Table 5: Additional 100-episode controls under the expanded 7-seed protocol.

Method	CALVIN	RLBench
Pixel BC	0.218 ± 0.032	0.286 ± 0.036
CQL	0.204 ± 0.031	0.276 ± 0.042
MVP	0.251 ± 0.032	0.334 ± 0.034
R3M	0.282 ± 0.039	0.361 ± 0.041

A.2 Compute and latency

Table 6: Model and compute profile. Grounding cost is per seed; latency is batch-one inference on an A100.

Method	Total	Frozen	Trainable	Ground GPU-h	Latency
TIDE	90.9M	90.5M	0.37M	0.82	13.7ms
V-JEPA	88.7M	88.3M	0.37M	0.79	11.9ms

Both temporal backbones use 576 pretraining GPU-hours; the matched V-JEPA run records 576 GPU-h. The grounding head accounts for the trainable parameters in both rows.

B TRANSFER-INTERFACE POSITIONING

Table 7: Transfer objects and grounded mappings in closely related passive-video approaches. A dash denotes that named-factor visibility is not the method's intervention axis.

Method	Video-trained object	Robot grounding path	Direct exposure intervention
LAPA	Discrete latent actions between frames	Latent VLA fine-tuning to robot actions	—
Moto	Autoregressive latent motion tokens	Motion-token and control co-fine-tuning	—
AMPLIFY	Discrete keypoint-motion tokens and forward dynamics	Separately trained inverse-dynamics mapping	—
ViPRA	Future observations and motion-centric latent actions	Chunked flow-matching action decoder	—
CoMo	Continuous latent motion	Pseudo-action policy co-training	—
V-JEPA	Masked latent video targets	Frozen ordered temporal readout in our protocol	Reference export
TIDE	Slots, skill state, egomotion, and observed cut resets	Frozen object–skill export plus direct-action head	Same-backbone factor visibility

Latent-action methods optimize the content or translation of a motion representation, often with fine-tuning, co-training, or a learned inverse-dynamics decoder. TIDE is complementary: it fixes a frozen grounding path and asks whether every variable useful inside the passive-video model should be a direct policy coordinate. The exposure matrix is the corresponding technical instrument.

C GROUNDING-BUDGET SWEEPS

Table 8: CALVIN success across grounding budgets, mean $\pm$ standard deviation over 7 seeds.

Method	25	50	100	500
TIDE	0.247 ± 0.030	0.386 ± 0.032	0.516 ± 0.029	0.681 ± 0.025
V-JEPA	0.159 ± 0.036	0.286 ± 0.039	0.420 ± 0.035	0.642 ± 0.030
Voltron	0.091 ± 0.028	0.187 ± 0.033	0.324 ± 0.038	0.572 ± 0.034
VIP	0.102 ± 0.031	0.194 ± 0.036	0.302 ± 0.034	0.548 ± 0.037

Across both suites, the 25–100-episode region contains the largest absolute separation between selective and generic temporal export. At 500 episodes, every representation benefits substantially and the remaining TIDE advantage is 0.039 on CALVIN and 0.038 on RLBench, localizing the main benefit to scarce grounding rather than asymptotic policy capacity.

Table 9: RLBench success across grounding budgets, mean $\pm$ standard deviation over 7 seeds.

Method	25	50	100	500
TIDE	0.304 ± 0.035	0.454 ± 0.031	0.579 ± 0.028	0.739 ± 0.024
V-JEPA	0.221 ± 0.038	0.352 ± 0.041	0.492 ± 0.034	0.701 ± 0.027
Voltron	0.158 ± 0.030	0.276 ± 0.036	0.407 ± 0.040	0.667 ± 0.031
VIP	0.149 ± 0.033	0.263 ± 0.039	0.381 ± 0.036	0.641 ± 0.034

The episode ratios in Section 5.3 use monotone linear interpolation in log episode count at the two-dimensional success target defined by TIDE's 100-episode CALVIN and RLBench rates. Each bootstrap replicate interpolates independently, and the larger episode requirement across the two suites defines the joint target.

D BOUNDARY PAIRED DIFFERENCES

Each visibility condition is paired by grounding seed with the selective-export model. Table 10 reports the resulting two-sided intervals; all five contrasts retain the same sign on both suites.

Table 10: Control minus TIDE paired difference for each visibility condition; negative values favor TIDE. Intervals are 95% hierarchical bootstrap.

Interface	CALVIN difference	RLBench difference
Slots only	-0.180 $[-0.211, -0.148]$	-0.158 $[-0.188, -0.127]$
Add cut coordinate	-0.067 $[-0.096, -0.039]$	-0.067 $[-0.098, -0.036]$
Add egomotion coordinates	-0.054 $[-0.080, -0.028]$	-0.053 $[-0.080, -0.026]$
Expose all	-0.119 $[-0.151, -0.088]$	-0.125 $[-0.158, -0.091]$
No acquisition branches	-0.085 $[-0.114, -0.057]$	-0.103 $[-0.136, -0.071]$

E TASK-LEVEL RESULTS

E.1 RLBENCH SINGLE-TASK POLICIES

The aggregate RLBench gain is distributed across the task set. Table 11 reports the common evaluation denominator and Wilson intervals for every task, making both the high-success and contact-intensive regimes visible.

Table 11: Per-task successes out of 700 evaluation episodes, with rate and Wilson 95% interval.

Task	TIDE	V-JEPA
Reach target	642/700 (0.917 [0.894,0.935])	601/700 (0.859 [0.831,0.882])
Push button	529/700 (0.756 [0.723,0.786])	470/700 (0.671 [0.636,0.705])
Slide block	443/700 (0.633 [0.597,0.668])	367/700 (0.524 [0.487,0.561])
Stack blocks	285/700 (0.407 [0.371,0.444])	223/700 (0.319 [0.285,0.354])
Pick and lift	412/700 (0.589 [0.552,0.624])	361/700 (0.516 [0.479,0.553])
Put in drawer	360/700 (0.514 [0.477,0.551])	300/700 (0.429 [0.392,0.466])
Close jar	390/700 (0.557 [0.520,0.594])	316/700 (0.451 [0.415,0.488])
Turn tap	424/700 (0.606 [0.569,0.641])	361/700 (0.516 [0.479,0.553])
Screw nail	276/700 (0.394 [0.359,0.431])	206/700 (0.294 [0.262,0.329])
Sort shape	292/700 (0.417 [0.381,0.454])	238/700 (0.340 [0.306,0.376])

The largest gaps occur in slide, close-jar, turn-tap, and screw interactions, where temporal contact state is especially useful. Reach remains high for both methods, while stack and screw remain the hardest tasks; the macro gain is distributed rather than dominated by one task.

E.2 MANISKILL2

ManiSkill2 changes simulator and controller while retaining the 100-episode grounding budget. The hardest articulated and contact-rich tasks remain difficult for every method, but each row contributes positively to TIDE's aggregate lead.

Table 12: ManiSkill2 successes out of 700 evaluation episodes per task.

Task	TIDE	V-JEPA	Voltron	VIP
PickCube	522	469	406	385
StackCube	399	331	273	259
PegInsertionSide	266	210	161	147
TurnFaucet	392	336	280	266
PushChair	196	154	119	105
MoveBucket	217	168	133	126
OpenCabinetDoor	308	245	189	180
OpenCabinetDrawer	321	265	197	190

F INPUT-SWAP EVALUATOR

Simulator instance masks are used only to construct the evaluation perturbation and are never supplied to a policy. We draw 2,400 probes per method, stratified over task and source-trajectory groups. A probe pairs source and held-out observations from the same task phase and object category, aligns the held-out object crop to the source mask by centroid and scale, and replaces that RGB region while preserving every other input pixel and the preceding video context. Every method then recomputes its representation and action. Let $\Delta\mathbf{a}$ be the policy's action change between source and swapped input, and let $\Delta\mathbf{a}^\star$ be the difference between the paired simulator actions. Continuous dimensions are normalized to $[-1, 1]$. Agreement requires cosine similarity between $\Delta\mathbf{a}$ and $\Delta\mathbf{a}^\star$ of at least 0.90 and normalized magnitude error at most 0.15; policies with a gripper dimension must also match the target gripper transition. The bootstrap unit is the source trajectory, preventing multiple swaps from one trajectory from being treated as independent.