LEARNING WORLD REPRESENTATIONS FROM FUTURE PHYSICAL OUTCOMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-stage manipulation requires early actions to leave suitable conditions for later contact. We study how to learn world-model states that retain physical information needed for subsequent actions under limited interaction. Small pose differences can change later outcomes, motivating supervision that describes action consequences. We train a shared state to predict outcome distributions under different candidate action sequences. A generative model jointly predicts physical outcomes and next states, with training through generated states for sequential prediction and feedback planning. Across three manipulation tasks, matched downstream models show that physical supervision improves held-out prediction and mean task success from frozen representations. The full system improves success over direct task-value prediction by 7.3 to 11.1 percentage points on jointly unseen objects and action combinations. Further tests demonstrate useful prediction from generated states, frozen dynamics reuse for new programs, and source-dependent representation transfer.

1 INTRODUCTION

Multi-stage manipulation requires early actions to leave suitable physical conditions for later stages. In assembly, two grasps may both lift a part while leaving different poses for transport and insertion. Task-oriented grasping and skill planning account for these downstream requirements (Zhao et al., 2021; Agia et al., 2023). For a learned world model, anticipating these consequences depends on which information its state preserves from the observed history.

The representation-learning question is what this state should retain under limited interaction data and model capacity. Small pose differences can change later contact, and training targets guide which distinctions the state preserves. A task-success target evaluates outcomes under a specified criterion. Physical measurements, such as object positions and rotations after an action, describe the conditions that later actions encounter. We hypothesize that predicting these measurements encourages states to retain information useful for subsequent manipulation. Visually different objects may also exhibit related physical responses, motivating the question of whether such supervision supports experience reuse.

Our principle is to train one shared state to predict physical outcome distributions under different candidate action sequences (Fig. 1). The state encodes observed history and available task context, while candidate sequences enter a shared response predictor. Errors on recorded sequences and outcomes update the encoder, encouraging the state to retain information useful across possible continuations. This principle builds on predictive-state representations and future-measurement learning (Littman & Sutton, 2001; Dosovitskiy & Koltun, 2017).

Sequential prediction requires generated states to support subsequent motion primitives, such as transport or insertion, before new observations arrive. After representation learning, we freeze the encoder and fit a transition model that jointly generates physical outcomes and the next state. Each sampled outcome stays paired with the state used by the next primitive. A generated intermediate state can differ from the recorded one, making the recorded next outcome a potentially inconsistent training target. We therefore train on the joint physical outcomes of complete two-primitive chains. A continuation value estimates task success from generated terminal states. The robot executes the first selected primitive and replans from real observations.

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.

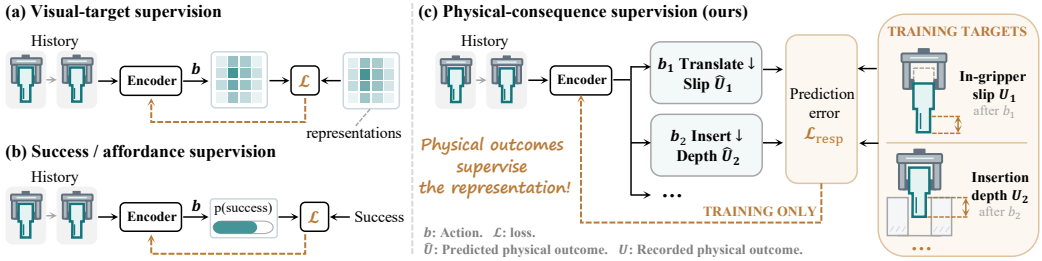

Figure 1: Learning from future physical outcomes. Candidate action sequences query a shared state, and their recorded outcomes supervise its encoder. Visual and success-based targets provide alternative supervision.

Across peg insertion, kit assembly, and tool-based cube pushing, we evaluate unseen objects and new combinations of familiar actions under matched interaction budgets. With encoders frozen and downstream models matched on common targets, physical supervision improves physical prediction and mean complete-task success. Further tests show useful prediction from generated states, frozen dynamics reuse within the learned primitive domain, and source-dependent transfer under limited target data.

Our contributions are threefold. First, we establish the value of physical-outcome supervision under matched downstream learning and characterize how its benefits vary with supervision horizon and interaction budget. Second, we connect these states to sequential prediction through joint outcome-and-state generation and training on complete two-primitive chains. Third, we demonstrate frozen dynamics reuse for new programs and source-dependent gains in cross-object representation transfer.

2 Related Work

Predictive-state representations describe histories through future tests (Littman & Sutton, 2001; Sun et al., 2016), while direct future prediction (DFP) supports action selection across goals through future measurements (Dosovitskiy & Koltun, 2017). Horde learns predictions under multiple policies and termination conditions (Sutton et al., 2011). Value-aware models and value equivalence prioritize value-relevant distinctions (Farahmand et al., 2017; Grimm et al., 2020). Successor features and Outcome-Predictive State Representations connect predictive abstractions to transfer across tasks (Barreto et al., 2017; Vereecken et al., 2026). We study how the prediction target shapes a state for multi-stage manipulation under limited interaction. Frozen-encoder comparisons test the resulting representations with matched physical predictors and a shared feedback planner.

Joint forward and inverse models learn visual features for object manipulation (Agrawal et al., 2016), while inverse-dynamics features support curiosity-driven exploration (Pathak et al., 2017). Sensorimotor World Models jointly train an encoder with forward and inverse dynamics, encouraging states to retain action-related information (Ivashkov et al., 2026). WorldSync broadens action coverage, supervises video features with future end-effector trajectories, and aligns action-intervention effects (Chen et al., 2026). PhyLatent supervises encoded and predicted representations with physical states and separates branches conditioned on different actions (Zeng et al., 2026). We connect physical supervision to joint outcome-and-state generation and test subsequent prediction from generated states.

Task-oriented grasping anticipates assembly requirements (Zhao et al., 2021). Deep Affordance Foresight predicts future affordances, and STAP coordinates skill sequences through their feasibility (Xu et al., 2021; Agia et al., 2023). Generative Skill Chaining composes learned skill distributions for long-horizon planning and replanning (Mishra et al., 2023).

Latent world models support imagined control and planning (Hafner et al., 2019; 2025; Zhou et al., 2025). TD-MPC2 combines short-horizon prediction with terminal values (Hansen et al., 2024). Diffuser plans by jointly denoising state-action trajectories (Janner et al., 2022). Self Forcing trains autoregressive video diffusion on self-generated context using sequence-level distribution-matching

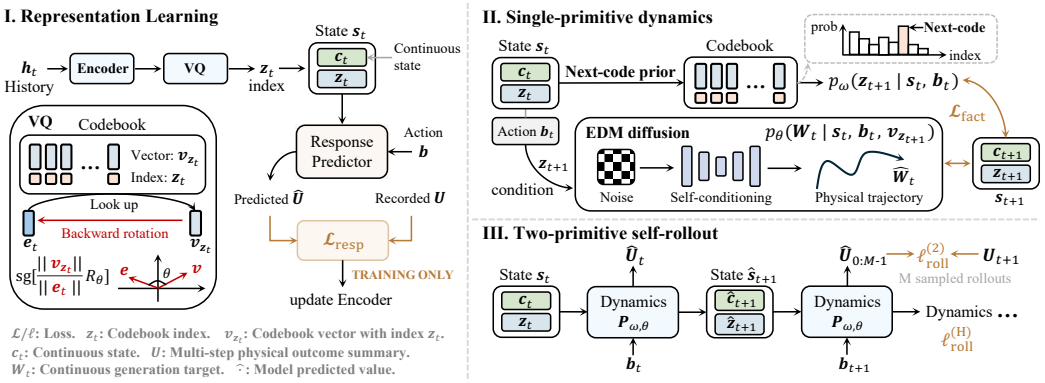

Figure 2: Physics-VQ training. I: physical-outcome supervision and quantization. II: next-code prediction and conditional diffusion. III: generated-state rollout with joint outcome scoring. Inset: sg stops gradients. R_θ rotates encoder vectors by angle θ. Rollouts use H primitives and M sampled chains ($H = 2$, $M = 4$ in training).

losses (Huang et al., 2025). We fit joint physical outcome distributions across two primitives, conditioning later predictions on generated states. App. H expands these connections.

3 Method

We learn shared states from action-conditioned physical outcomes, then freeze the encoder to fit joint outcome-and-state transitions. Training through generated states supports sequential prediction and feedback planning (Fig. 2).

3.1 Learning a state from future physical outcomes

One model step advances a motion primitive, such as carrying or insertion. Its command b_t specifies motion parameters, duration, and a controller that uses sensor feedback during execution. The encoder $\mathcal{E}_\Phi$ maps the available history h_t of images, depth, robot readings, and executed commands to a state s_t. Geometry and task context, supplied to all methods, are implicit in this notation. During representation learning, candidate commands condition the response head, while the encoder computes one current state shared across those commands.

A training record contains a command pair $\mathbf{b} = (b_t, b_{t+1})$ fixed before execution and its physical outcomes Y_t, Y_{t+1} from one branch. These outcomes record object positions and rotations at fixed phases of each primitive. The fixed map $\mathcal{S}$ concatenates and normalizes them using training statistics to give $U_t^{(2)} = \mathcal{S}(Y_t, Y_{t+1})$. A Gaussian-mixture response head r_ψ predicts this joint two-primitive summary:

$$s_t = \mathcal{E}_\Phi(h_t), \qquad \mathcal{L}_{\text{resp}} = \mathbb{E}_{(h_t, \mathbf{b}, U_t^{(2)}) \sim \mu} \left[-\log r_\psi\big(U_t^{(2)} \mid s_t, \mathbf{b}\big) \right]. \tag{1}$$

Here μ balances histories and command families. Prediction errors across commands update the shared encoder, encouraging it to retain distinctions that affect future physical outcomes. Simulator physical measurements supply targets, while the encoder reads available observations.

Both state variants retain continuous fields $c_t = C(h_t)$ and a learned history feature $e_t = E_\phi(h_t)$, where C assembles available fields and ϕ is part of Φ. The fields contain geometry, goals, robot measurements, estimated object poses, perception confidence, and remaining budgets. Physics-C uses $s_t^{\text{C}} = (c_t, e_t)$. Physics-VQ applies vector quantization (VQ) to form $s_t = (c_t, z_t)$, where $z_t = \arg\min_{1 \le k \le K} \|e_t - v_k\|_2^2$ selects the nearest of K code vectors v_k (van den Oord et al., 2017). Downstream networks read (c_t, v_{z_t}). We use 128-dimensional history features and $K = 256$ codes, with SimVQ and the Rotation Trick for quantization (Zhu et al., 2025; Fifty et al., 2025). Physics-VQ adds codebook fitting and commitment losses. Neither physical model uses reconstruction loss. Predictive information is retained when the state and the full history with task context give the same

physical-outcome distribution under the sampled command pairs. Histories with different outcome distributions under the same sampled commands should therefore remain distinguishable in the state. App. A formalizes this criterion and gives the full objective.

3.2 Predicting outcomes and the next state

The response head predicts a two-primitive summary directly from the observed starting state. For sequential prediction, a separate transition model generates each primitive's physical outcome and passes its generated state to the next command. Write $c_t = (c_t^{\text{static}}, c_t^{\text{dyn}})$ for static context and dynamic observed fields. The map $\mathcal{T}$ packs the physical trajectory and next dynamic fields at fixed scales, giving $W_t = \mathcal{T}(Y_t, c_{t+1}^{\text{dyn}})$. Physics-VQ factors this joint transition into a next-code prior p_ω and a continuous generator p_θ:

$$p_{\omega,\theta}(z_{t+1}, W_t \mid s_t, b_t) = p_\omega(z_{t+1} \mid s_t, b_t)\, p_\theta(W_t \mid s_t, b_t, v_{z_{t+1}}). \tag{2}$$

Each sampled code remains paired with its generated outcome and dynamic fields. The next state copies static context and reads predicted observation fields, which have separate targets from physical ground truth. Physics-C instead jointly generates (W_t, e_{t+1}). The routine $\text{Sample}(s_t, b_t)$ returns the paired outcome and next state, with hats denoting generated quantities. App. A specifies its sampling and rotation projection.

We freeze the encoder, codebook, continuous interface, and normalization before fitting dynamics. The continuous generator uses EDM diffusion (Karras et al., 2022), and recorded histories provide fixed state labels for the factual loss:

$$\mathcal{L}_{\text{fact}} = \mathbb{E}[-\log p_\omega(z_{t+1} \mid s_t, b_t)] + \mathcal{L}_{\text{EDM}}. \tag{3}$$

The prior predicts the recorded next code, and the denoiser recovers W_t from its noisy version, conditioned on that same code. EDM with self-conditioning (EDM-SC) supplies a preliminary denoiser prediction as an additional input (Chen et al., 2023). Factual training averages both recorded transitions per example. Physics-C omits code cross entropy and denoises (W_t, e_{t+1}). We then fit the continuation value used below and retain it during dynamics refinement.

3.3 Training through generated states

During sequential prediction, the second primitive reads a generated state without an intermediate observation. A plausible generated intermediate state may differ from the recorded state. Pairing it directly with the recorded second-primitive outcome can therefore impose an inconsistent conditional target. We instead generate complete two-primitive chains from the observed starting state and score their joint physical outcomes.

Let U and $\widehat{U}_m$ denote recorded and generated two-primitive summaries. For $M = M_{\text{roll}} \geq 2$ independent chains, the energy-score estimate is

$$\widehat{\ell}_{\text{roll}} = \frac{1}{M}\sum_{m=1}^{M} \|\widehat{U}_m - U\|_2 - \frac{1}{2M(M-1)} \sum_{m \neq n} \|\widehat{U}_m - \widehat{U}_n\|_2. \tag{4}$$

Together, these terms score agreement with observations and predictive dispersion (Gneiting & Raftery, 2007). Because each generated summary contains both primitives from one chain, the score evaluates their joint physical distribution, including dependence across primitives. Under finite first moments, the expected score is minimized by the true joint summary distribution conditional on the starting state and commands. Terminated executions retain their final physical state through the prescribed two-primitive window. Fig. 2 denotes the joint target by U_{t+1} and indexes generated summaries from zero.

In two-primitive rollout training (T2), scheduled updates add this score with $M_{\text{roll}} = 4$ and unit weight, while every update retains factual loss. Continuous gradients connect both generated primitives. For Physics-VQ, ∇^{path} holds sampled code indices fixed, while factual cross entropy trains the code prior. Physics-C also propagates gradients through its generated history feature. Algorithm 1 gives one scheduled rollout update, where $\ell_{\text{fact}}(j)$ is the factual loss on recorded transition j and $\text{mean}_{\mathcal{B}}$ averages over minibatch $\mathcal{B}$. App. A gives the schedule and full gradient analysis.

Algorithm 1 Scheduled rollout update

Input: Records, ω, θ, $M = M_{\text{roll}}$.
Frozen: Representation and value.
1: Sample $\mathcal{B}$; encode $s_{t:t+2}$ and compute U
2: $\mathcal{L}_{\text{fact}} \leftarrow \text{mean}_{\mathcal{B}}[\frac{1}{2}\sum_{j=t}^{t+1} \ell_{\text{fact}}(j)]$
3: **for** $m = 1, \ldots, M$ **do** (per record)
4: $\quad(\widehat{Y}_t, \widehat{s}_{t+1}) \leftarrow \text{Sample}(s_t, b_t)$
5: $\quad(\widehat{Y}_{t+1}, \widehat{s}_{t+2}) \leftarrow \text{Sample}(\widehat{s}_{t+1}, b_{t+1})$
6: $\quad\widehat{U}_m \leftarrow \mathcal{S}(\widehat{Y}_t, \widehat{Y}_{t+1})$
7: **end for**
8: $\mathcal{L}_{\text{roll}} \leftarrow \text{mean}_{\mathcal{B}}[\widehat{\ell}_{\text{roll}}(U; \widehat{U}_{1:M})]$
9: Update ω, θ with AdamW $\nabla^{\text{path}}(\mathcal{L}_{\text{fact}} + \mathcal{L}_{\text{roll}})$

Algorithm 2 Feedback planning

Input: Real history h_t, goal g, horizon H.
Frozen: Encoder, transition, and V_g.
1: $s_t \leftarrow \mathcal{E}_\Phi(h_t)$; propose candidates $\mathcal{C}_t$
2: **for each** $\mathbf{b} \in \mathcal{C}_t$ **do**
3: $\quad$Draw M_{plan} complete chains from s_t using Sample for each primitive
4: $\quad$Compute $\widehat{J}_g(\mathbf{b} \mid s_t)$ by Eq. 5
5: **end for**
6: $\mathbf{b}^* \leftarrow \arg\max_{\mathbf{b} \in \mathcal{C}_t} \widehat{J}_g(\mathbf{b} \mid s_t)$
7: Execute b_t^*; observe the new history h_{t+1}
8: Set $t \leftarrow t + 1$; repeat until execution ends

3.4 Planning with real feedback

A continuation value V_g predicts task success for goal g, given the remaining budget and a fixed controller for the program's remaining primitives. With the representation frozen, binary cross entropy fits V_g to real continuation outcomes, including completed success and terminal failure. For each candidate $\mathbf{b} = (b_t, \ldots, b_{t+H-1})$, the planner generates M_{plan} independent chains over H primitives and averages their terminal values:

$$\widehat{J}_g(\mathbf{b} \mid s_t) = \frac{1}{M_{\text{plan}}} \sum_{m=1}^{M_{\text{plan}}} V_g(\widehat{s}_{t+H}^{(m)}). \tag{5}$$

All sampled branches contribute. The robot executes the selected sequence's first primitive, then re-encodes real observations before replanning (Alg. 2). Model parameters remain fixed during execution.

Complete-task success evaluates the resulting feedback policy. New goals, programs, and geometry enter through the shared task interface. App. E specifies the adaptation protocols.

4 Experiments

4.1 Setup and Complete-Task Generalization

Peg insertion, kit assembly, and tool-based cube pushing (Peg, Kits, and Poke) extend ManiSkill with varied object or tool geometry (Gu et al., 2023; Tao et al., 2025). Each couples grasping or tool pickup to later contact. Poke scores target placement, with strict tool use evaluated separately. We cross seen/unseen objects with seen/unseen action combinations built from covered local transitions. Each split has 1,024 starts: eight families, four objects per family, and 32 starts per object.

Main comparisons match 160,000 primitives per task, observation histories, geometry, candidate actions, and low-level feedback. A two-primitive execution counts as two interactions. Visual and Affordance change representation supervision within the common physical-model pipeline. Other baselines use geometric feasibility (Geometry), complete-task success (Value), future measurements (DFP), or skill dynamics with a Q-function (Skill+Q). DFP predicts the same physical measurements as our model, also testing how predictions are modeled and used. DINO-WM* and TD-MPC2* are development-selected local adaptations (Zhou et al., 2025; Hansen et al., 2024). Physics-VQ, Physics-C, Value, Affordance, and Kits TD-MPC2* use ten training seeds; other learned systems use five. Each seed independently trains the representation and downstream modules. Where shown, SD denotes training-seed standard deviation. Differences are computed before rounding and reported in percentage points (pp) for success. Bold and underlining mark the best and second-best means, including ties. We use paired hierarchical bootstrap intervals with crossed training seeds, Holm correction over 30 specified contrasts, and finite-cluster sensitivity analysis (App. C).

On unseen objects and unseen combinations, Physics-VQ improves over Value by 11.0, 11.1, and 7.3 percentage points on Peg, Kits, and Poke (Table 1). The paired 95% intervals are [7.8, 14.3],

Table 1: Complete-task success (%) at 160k primitives. S/U means seen/unseen, for objects then action combinations. Δ is U/U success minus Value (pp). Across tables, teal/ochre marks positive/negative differences. Asterisks mark local adaptations.

	Peg					Kits					Poke				
Method	S/S	S/U	U/S	U/U	Δ	S/S	S/U	U/S	U/U	Δ	S/S	S/U	U/S	U/U	Δ
Geometry	69.6	66.5	65.6	60.4	-4.1	60.0	56.1	53.0	45.9	-7.4	73.2	75.7	73.6	70.5	+2.3
Value	75.3	70.5	70.1	64.5	0.0	66.7	63.3	59.1	53.3	0.0	74.8	74.7	73.3	68.2	0.0
DFP	73.4	67.4	66.8	60.8	-3.7	65.7	62.9	58.8	52.7	-0.6	76.2	76.3	74.7	69.9	+1.6
Skill+Q	74.1	68.2	67.6	62.0	-2.5	67.8	65.5	59.9	55.7	+2.3	74.8	75.0	73.8	69.4	+1.1
Visual	76.1	71.5	70.7	64.7	+0.3	66.9	63.5	58.8	54.2	+0.9	75.8	75.4	72.8	69.9	+1.6
Affordance	79.7	74.8	74.6	68.5	+4.1	71.2	68.8	64.3	59.2	+5.9	80.4	79.6	78.9	73.5	+5.3
DINO-WM*	78.6	73.1	73.0	67.4	+2.9	71.5	68.3	63.2	58.8	+5.4	77.2	78.5	75.9	71.3	+3.1
TD-MPC2*	77.8	73.5	72.8	66.5	+2.1	72.5	69.4	65.5	60.1	+6.8	78.2	77.8	75.4	71.6	+3.3
Physics-C	83.2	78.3	77.9	71.7	+7.2	77.0	73.5	69.5	64.2	+10.9	**82.0**	81.4	**79.7**	75.4	+7.1
Physics-VQ	**86.5**	**81.7**	**81.2**	**75.4**	+11.0	**77.5**	**74.0**	**69.7**	**64.4**	+11.1	**82.0**	**81.6**	79.6	**75.5**	+7.3

Table 2: Frozen-encoder prediction and control at 160k primitives (mean ± SD; five seeds). The last row gives PHY2-minus-AFF2 differences with paired-seed SD, in ES units or pp.

	Peg		Kits		Poke	
Target	ES ↓	Success (%) ↑	ES ↓	Success (%) ↑	ES ↓	Success (%) ↑
REC	0.746 ± 0.019	61.2 ± 1.7	0.674 ± 0.020	47.9 ± 1.4	0.672 ± 0.021	65.2 ± 2.3
VIS2	0.656 ± 0.024	64.7 ± 2.3	0.628 ± 0.014	54.2 ± 3.0	0.572 ± 0.019	69.9 ± 2.1
VAL	0.642 ± 0.019	68.1 ± 2.0	0.606 ± 0.023	58.1 ± 1.5	0.550 ± 0.016	72.3 ± 2.1
AFF2	0.625 ± 0.018	68.3 ± 1.5	0.585 ± 0.004	59.5 ± 2.4	0.541 ± 0.017	72.5 ± 1.5
PHY1	0.609 ± 0.006	70.0 ± 2.4	0.575 ± 0.006	62.0 ± 0.6	0.533 ± 0.010	75.5 ± 1.4
PHY2	**0.581** ± 0.004	**70.7** ± 1.2	**0.545** ± 0.004	**66.4** ± 1.9	**0.498** ± 0.002	**76.0** ± 1.5
PHY2 − AFF2	-0.044 ± 0.016	+2.4 ± 2.4	-0.040 ± 0.005	+6.9 ± 2.7	-0.043 ± 0.017	+3.4 ± 2.3

[7.6, 14.5], and [4.5, 10.3] points, respectively. These gains remain significant after Holm correction and finite-cluster sensitivity analysis. Physics-C also improves mean success over Value on all three tasks, with results close to Physics-VQ on Kits and Poke.

4.2 Effects of Physical Supervision

We next isolate how representation supervision contributes to prediction and control. We compare reconstruction (REC), two-primitive visual prediction (VIS2), complete-task value (VAL), two-primitive binary affordances (AFF2), and one- or two-primitive physical outcomes (PHY1/PHY2). All six use continuous history features, matched inputs and capacity, and 160k interactions. We freeze each encoder, fit a newly initialized width-128 two-primitive physical head, and evaluate its predictions on held-out actions. The energy score (ES) evaluates predicted outcome distributions against observations, with lower scores indicating better predictions. Common transition models use the same physical targets, and value heads use the same task-success labels. A shared planner tests U/U complete-task execution from these frozen encoders. The five paired seeds share system-comparison runs wherever configurations match; main Physics-C and Affordance means include five additional seeds. VAL uses the common physical model, whereas Value in Table 1 directly scores candidate sequences.

PHY2 yields the lowest frozen-encoder ES and highest mean success on all three tasks (Table 2). All nine ES reductions relative to AFF2, VAL, and VIS2 remain significant after Holm correction. PHY1 also yields lower ES and higher mean success than AFF2 on all three tasks, so physical supervision remains useful with a shorter supervision horizon.

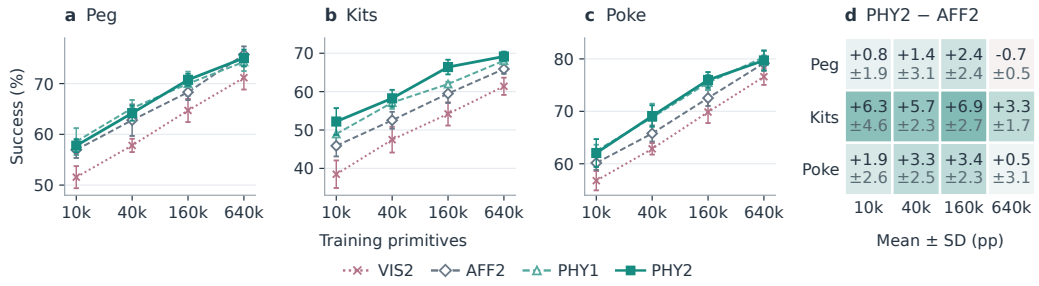

Figure 3: Frozen continuous encoders across budgets (EDM-SC/T2). (a–c) Success (mean ± training-seed SD), with task-specific vertical ranges. (d) Paired PHY2-minus-AFF2 gains (mean ± SD, pp), using the same five seeds.

Table 3: Generated-state prediction and training at 160k primitives. (a) Physics-VQ/T2 mean joint ES; the reference ES is shared across the five runs. (b) Success, mean ± SD over five seeds. Δ: T2 − T2-stop, with paired-seed SD. C/VQ: continuous/quantized history features.

(a) Joint query ES ↓

Predictor	Peg	Kits
Real-state reference	0.5620	0.5387
Generated state	0.5727	0.5500
Observed fields	0.8328	0.8275
Actions only	1.5145	1.5438
ES gap	+0.0107	+0.0112
Relative (%)	+1.9	+2.1

(b) Evaluation success (%) ↑

Training	Peg (C)	Peg (VQ)	Kits (C)	Kits (VQ)
T0	67.4 ± 1.7	73.1 ± 1.9	61.3 ± 3.1	61.6 ± 1.0
T1a	68.3 ± 1.2	73.3 ± 1.0	62.9 ± 3.3	62.2 ± 0.8
T1b	69.6 ± 1.5	74.4 ± 1.3	64.8 ± 3.2	62.9 ± 1.2
T2-stop	70.6 ± 2.5	74.9 ± 1.7	64.9 ± 1.7	64.0 ± 2.2
T2	70.7 ± 1.2	75.7 ± 0.9	66.4 ± 1.9	64.2 ± 0.7
Δ (pp)	+0.2 ± 3.2	+0.7 ± 1.6	+1.5 ± 2.1	+0.2 ± 2.4

Across four budgets, physical supervision gives higher mean success than visual supervision at every measured budget (Fig. 3). Kits favors PHY2 throughout, while PHY1 is competitive and sometimes better on Peg and Poke. At 640k, AFF2 slightly exceeds PHY2 on Peg. The useful supervision horizon therefore depends on the task and available data.

Additional controls retain the PHY2 advantage over AFF2 across continuous and quantized states and mixture and diffusion generators (App. D). Removing learned history lowers success despite retaining observed continuous fields. Broader command coverage helps at fixed starting histories, but competes with scene coverage under a fixed acquisition budget.

4.3 Subsequent Prediction from Generated States

Before new observations arrive, generated states must support held-out actions. We query Physics-VQ trained with T2, including insertion after predicted transport.

The generated path predicts the transport outcome and next state, while the reference uses repeated real transports from the same restored scene. Both paths use the same separately fitted, frozen one-primitive prediction head and are scored against separate recorded joint outcomes. On Peg and Kits, five seeds evaluate 1,024 starts with four queries each, scoring three physical coordinates per primitive jointly.

Generated-state joint ES is 1.9% and 2.1% above the real-state reference on Peg and Kits, respectively (Table 3a). Both scores are below the observed-fields-only and action-only controls, supporting useful subsequent prediction of the physical summary from generated states.

T0 uses factual transitions, T1a adds cost-matched factual training, and T1b trains a direct two-primitive physical head without intermediate-state generation at deployment (Table 3b). T2 trains through generated states; T2-stop cuts its cross-primitive continuous gradient. At 160k, T2 improves mean success over T1b in both state forms. For Physics-VQ, these gains are 1.3/1.2 points on

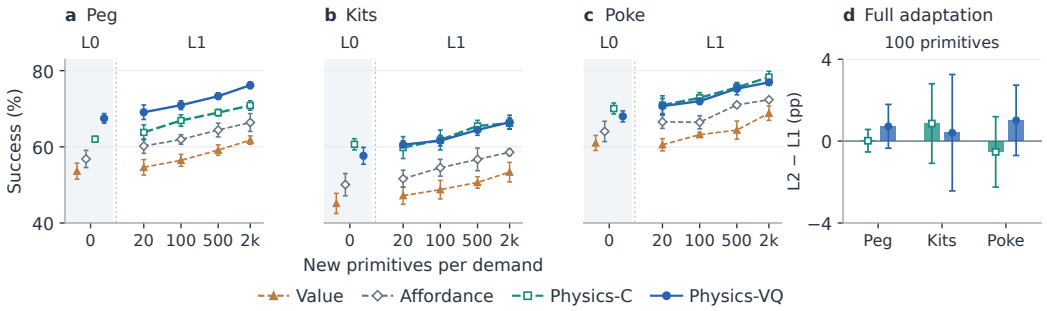

Figure 4: New programs (five seeds). (a–c) Success, mean ± SD, with frozen (L0) or high-level-adapted (L1) models. (d) Full-adaptation gain (L2 − L1) at 100 primitives per demand, mean ± paired-seed SD (pp).

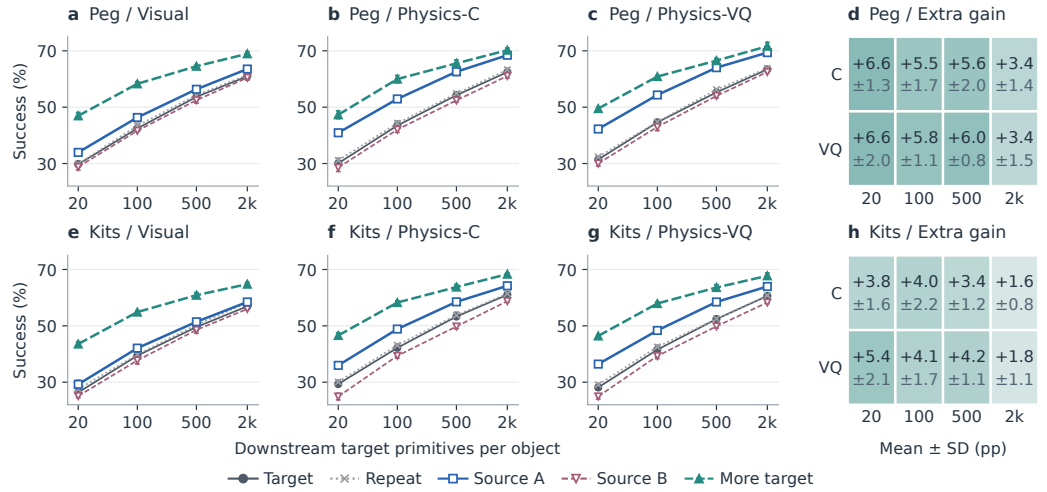

Figure 5: Source transfer (five seeds, each averaging 12 targets). (a–c,e–g) Success, mean ± SD. (d,h) Extra gain is each physical state's Source A-minus-Target gain minus Visual's, mean ± paired-seed SD (pp).

Peg/Kits, whereas gains over T2-stop are 0.7/0.2 points. The contribution of continuous cross-primitive gradients varies across tasks and budgets.

A separate depth-holding test requires insertion depth and alignment to persist over consecutive control steps. Trajectory prediction improves success over endpoint-only prediction on this test, while adding event targets recovers part of the gain. Endpoint models remain competitive on ordinary tasks, so the benefit of trajectory prediction depends on the requirement. App. F gives query protocols, these comparisons, the 40k training results, and a separate third-primitive diagnostic.

4.4 Reusing Dynamics and Transferring Representations

We now change goals, action programs, or target geometry within the existing primitive domain, retaining the low-level controller. Program demands change the combinations, order, or parameters of existing primitives. All methods receive the new requirement for evaluation and planning. After 160k-primitive pretraining, L0 freezes all modules without new interaction. L1 adapts only the method's value, Q-function, or evaluator, keeping representation and dynamics fixed. L2 also adapts the representation and prediction modules using the same new-data budget. For each task, we average four demands per category within five seeds to report means and SDs (Fig. 4).

Without adaptation, both physical models exceed Value on new programs across all three tasks. Physics-C has higher mean success than Physics-VQ on Kits and Poke. High-level adaptation

improves Physics-VQ success, while updating representation and dynamics makes small, task-dependent changes at 100 primitives per demand. These results support adapting how predicted consequences are evaluated while reusing learned local transitions. App. E reports changed goals and geometry alongside the full adaptation comparisons.

Cross-object transfer compares target-only representations with encoders trained exclusively on 160k primitives from Source A or B. They are two of eight fixed source pools grouped by geometry, contact structure, and program similarity. After freezing each encoder, we fit fresh dynamics and value models on the same target records. Peg and Kits each use 12 target objects from six held-out families, five paired training seeds, and 256 starts per target.

At the primary 100- and 500-primitive budgets, Source A improves continuous PHY2 mean success by 5.2 to 9.4 points (Fig. 5). All four individual intervals exclude zero, and three pass Holm correction. Source B reduces mean success. The mean Source A gain exceeds Visual's by 3.4 to 5.6 points. Repeated target records help less than Source A; 160k extra target primitives for representation training help more, with downstream target data matched throughout.

4.5 Real Execution, Computation, and Limits

Real peg assembly starts from each method's development-selected model trained on 160k primitives in the PegInsertionSide-Geometry simulation task. Deployment uses common geometry and sensor calibration, without additional representation or dynamics training on the robot.

The frozen protocol keeps all learned modules fixed (Fig. 6). High-level adaptation updates only the evaluator, using 100 primitives per object from a shared 1,200-primitive pool. Each method/protocol has 480 trials across six families and four sessions. Physics-VQ has the highest observed success rates in both protocols. Frozen Physics-VQ improves over Value by 11.7 points, with a paired 95% confidence interval of [2.3, 21.0] points.

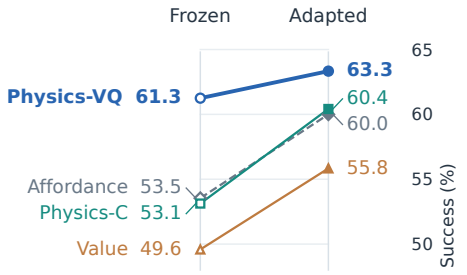

Figure 6: Robot success (480 trials per rate).

Retraining on MimicGen Three Piece Assembly and Threading at 160k primitives gives 59.6/87.9% success for Physics-VQ versus 49.8/76.8% for Value (Mandlekar et al., 2023).

Under a 100 ms deadline for every decision, Physics-VQ reaches 73.2/62.1/72.9% deadline-qualified success on Peg/Kits/Poke, versus Value's 64.5/54.0/69.8%. Mean per-run 95th-percentile (p95) latency is about 80 ms for Physics-VQ and 24 ms for Value.

Geometry scores higher on the simpler PickCube control (96.7% versus Physics-VQ's 91.9%). A separate physics shift lowers Physics-VQ from 74.7/65.4/76.0% to 61.8/55.6/63.1% on Peg/Kits/Poke (App. G).

5 Conclusion and Limitations

Future physical outcomes provide a useful supervision signal for world-model states in multi-stage manipulation. Matched downstream models show that this signal improves physical prediction and mean complete-task success from frozen representations. The useful supervision horizon depends on the task and interaction budget. Training joint outcome-and-state transitions through generated states supports subsequent prediction and feedback planning. Frozen dynamics serve new programs, with further gains from high-level adaptation. Source-object experience can improve representation learning under limited target data, with gains depending on the source pool.

These results concern short prediction horizons with real feedback and fixed low-level controllers. Longer horizons, changed physical conditions, source selection, and prediction cost remain central challenges for extending this approach.

AI USE STATEMENT

We used generative AI tools to polish the writing and assist with literature retrieval and discovery. The authors reviewed the AI-assisted text and checked the cited sources. We take responsibility for the final content of this work, including its text, claims, and results.

ETHICS STATEMENT

This work studies representation learning for robotic manipulation through simulated tasks and robot experiments. We do not identify ethical concerns specific to the reported study.

REPRODUCIBILITY STATEMENT

Section 3 presents the model, training objectives, and planning algorithms. Appendix A details the state and target interfaces, normalization, sampling procedure, and derivations with their assumptions. Appendices B and C describe the data and comparison protocols, evaluation hierarchy, and statistical analysis. Appendices D–G provide the settings and results for the supplementary experiments. Source code is provided in the supplementary materials.

REFERENCES

Christopher Agia, Toki Migimatsu, Jiajun Wu, and Jeannette Bohg. STAP: Sequencing task-agnostic policies. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7951–7958, 2023. doi: 10.1109/ICRA48891.2023.10160220.

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Andre Barreto, Will Dabney, Remi Munos, Jonathan J Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Byron Boots, Sajid M Siddiqi, and Geoffrey J Gordon. Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research*, 30(7):954–966, 2011.

Sixiang Chen, Jiaming Liu, Jixian Wu, Yichen Guo, Tinghao Wang, Siyuan Qian, Hao Chen, Jiajun Cao, Jian Tang, and Shanghang Zhang. Do robotic world models really follow actions? diagnosing and aligning action-conditioned generation for policy learning, 2026. URL https://arxiv.org/abs/2608.24885.

Ting Chen, Ruixiang ZHANG, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

Alexey Dosovitskiy and Vladlen Koltun. Learning to act by predicting the future. In *International Conference on Learning Representations*, 2017.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control, 2018. URL https://arxiv.org/abs/1812.00568.

Amir-Massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-Aware Loss Function for Model-based Reinforcement Learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1486–1494. PMLR, 2017.

Christopher Fifty, Ronald Guenther Junkins, Dennis Duan, Aniketh Iyengar, Jerry Weihong Liu, Ehsan Amid, Sebastian Thrun, and Christopher Re. Restructuring vector quantization with the rotation trick. In *International Conference on Learning Representations*, 2025.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: a new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Christopher Grimm, Andre Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5541–5552. Curran Associates, Inc., 2020.

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.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, volume 38, pp. 167283–167308. Curran Associates, Inc., 2025. doi: 10.52202/085713-5576.

Petr Ivashkov, Randall Balestriero, and Bernhard Schölkopf. Sensorimotor world models: Perception for action via inverse dynamics, 2026. URL https://arxiv.org/abs/2606.20104.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915. PMLR, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577. Curran Associates, Inc., 2022. doi: 10.52202/068431-1926.

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Michael Littman and Richard S Sutton. Predictive representations of state. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1820–1864. PMLR, 2023.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024.

Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2905–2925. PMLR, 2023.

Lorenzo Pacchiardi, Rilwan A. Adewoyin, Peter Dueben, and Ritabrata Dutta. Probabilistic forecasting with generative networks via scoring rule minimization. *Journal of Machine Learning Research*, 25(45):1–64, 2024.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 2017.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Satinder Singh, Michael R. James, and Matthew R. Rudary. Predictive state representations: a new theory for modeling dynamical systems. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pp. 512–519. AUAI Press, 2004.

Wen Sun, Arun Venkatraman, Byron Boots, and J.Andrew Bagnell. Learning to filter with predictive state inference machines. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1197–1205. PMLR, 2016.

Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th international conference on autonomous agents and multiagent systems-volume 2*, pp. 761–768, 2011.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Demonstrating GPU Parallelized Robot Simulation and Rendering for Generalizable Embodied AI with ManiSkill3. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.021.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Ruben Vereecken, Luke Dickens, and Alessandra Russo. Predictive representations for skill transfer in reinforcement learning, 2026. URL https://arxiv.org/abs/2604.07016.

Danfei Xu, Ajay Mandlekar, Roberto Martín-Martín, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Deep affordance foresight: Planning through what can be done in the future. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6206–6213. IEEE Press, 2021. doi: 10.1109/ICRA48506.2021.9560841.

Xi Zeng, Haojie Ren, and Ziying Song. Phylatent: Learning dynamics-relevant representations for jepa world models, 2026. URL https://arxiv.org/abs/2608.05720.

Jialiang Zhao, Daniel Troniak, and Oliver Kroemer. Towards robotic assembly by predicting robust, precise and task-oriented grasps. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1184–1194. PMLR, 2021.

Gaoyue Zhou, Hengkai Pan, Yann Lecun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025.

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22968–22977, 2025.

A METHOD DETAILS AND DERIVATIONS

State and physical targets. Table S4 specifies the observed state and its supervision. The notation $C(h_t)$ includes the given geometry and goal parameters. The transition jointly predicts the dynamic fields, including remaining primitive and control-step budgets when execution ends before a command's maximum duration.

Table S4: State and target interfaces. Physical normalization uses fixed training statistics.

Quantity	Recorded source	Generated transition	Role or scale
c_t^{static}	Geometry, calibration, and fixed goal	Copied within a rollout	Available task context
c_t^{dyn}	Robot readings, pose estimates, confidence, budgets	Jointly predicted	Observed state; interface scales
e_t or z_t	Encoded observation history	Generated feature or sampled code	Learned state; frozen interface
Y_t	Object positions and rotation matrices at fixed phases	Physical trajectory	Outcome target; physical units
W_t	Y_t and next dynamic fields	Joint continuous output	Normalized physical fields; unchanged dynamic scales
$U_t^{(2)}$	Physical fields from both primitives	Complete generated chain	Representation and rollout target; normalized

Each tracked object contributes its position in metres and nine rotation-matrix entries in a fixed world frame calibrated to the robot base. The collection interface samples these fields over the command's prescribed control-step window. After early completion or termination, the final physical state is held through the remaining window. The fixed map Ξ flattens the phase samples and recorded physical fields, giving

$$U_t^{(2)} = \mathcal{S}(Y_t, Y_{t+1}) = \mathcal{N}_U\big([\Xi(Y_t), \Xi(Y_{t+1})]\big). \tag{S6}$$

The map $\mathcal{N}_U$ uses per-coordinate training means and standard deviations, with a scale floor of 10^{-3}. It applies the same one-primitive statistics to each half of the summary and remains fixed throughout training. These scales determine the relative contribution of physical coordinates to the Euclidean rollout score. Diagnostic queries reverse this normalization and apply a common scale fixed on development data. Diagnostic coordinates summarize task-relative positional and orientation alignment plus a signed margin of insertion depth (Peg), seating (Kits), or directed advancement (Poke). A nonnegative margin indicates attainment of the corresponding physical threshold. The two-primitive test in App. F concatenates these summaries into six coordinates, distinct from the complete representation and rollout target.

Each training pair follows one precommitted command sequence from one observed boundary. Ordinary collection continues the same execution; branch collection restores a saved boundary before executing each alternative sequence. Snapshots include simulator and controller state, recorded history, and stage bookkeeping. The outer controller resets its hold counter at each primitive boundary. ManiSkill restores native controller state with the snapshot; MimicGen resets native controller goals at boundaries and after recovery. The controller continues to use sensor feedback within each primitive.

Representation objective and information retained. The history encoder uses a gated recurrent unit with hidden width 256, reading 16 frames and producing a 128-dimensional feature. Physics-VQ selects one of $K = 256$ effective code vectors. SimVQ constructs them from fixed base vectors $a_k \in \mathbb{R}^d$, where $d = 128$:

$$v_k = Aa_k + d_0, \qquad A \in \mathbb{R}^{d\times d}, \quad d_0 \in \mathbb{R}^d. \tag{S7}$$

Only A and d_0 are learned, and both nearest-neighbor assignment and downstream lookup use these effective vectors (Zhu et al., 2025).

The response head is a five-component diagonal Gaussian mixture:

$$r_\psi(U_t^{(2)} \mid s_t, \mathbf{b}) = \sum_{j=1}^{5} \pi_j \mathcal{N}\Big(U_t^{(2)} \mid \mu_j, \operatorname{diag}(\tau_j^2)\Big), \qquad \tau_j = \operatorname{softplus}(\alpha_j) + 10^{-3}. \tag{S8}$$

The weights π_j, mean vectors μ_j, and unconstrained scale vectors α_j depend on $(s_t, \mathbf{b})$. The weights are nonnegative and sum to one. PHY2 predicts the joint two-primitive target; PHY1 restricts the head and target to one primitive.

Physics-VQ minimizes

$$\mathcal{L}_{\text{repr}} = \mathcal{L}_{\text{resp}} + \mathbb{E}_{\mu}\left[\frac{1}{d}\|\operatorname{sg}(e_t) - v_{z_t}\|_2^2 + \frac{\beta}{d}\|e_t - \operatorname{sg}(v_{z_t})\|_2^2\right], \qquad \beta = 0.25, \tag{S9}$$

where sg stops gradients, the first penalty fits the codebook, and the second commits the encoder to its selected vector. Physics-C minimizes $\mathcal{L}_{\text{resp}}$ alone. The supervision comparisons replace $\mathcal{L}_{\text{resp}}$ with the corresponding REC, VIS2, VAL, or AFF2 target, retaining quantization penalties only for quantized states. Sampling weights histories equally, then command families within a history, then records within each family.

To formalize target preservation, let X contain the observed history and task context, B the commands, and U their physical target. For the deterministic state $S = f(X)$, let p denote the distribution under these sampling weights. Conditional mutual information gives the criterion

$$\mathsf{I}(U; X \mid S, B) = \mathbb{E}_{X,B}\,\mathrm{KL}\left(p(U \mid X, B) \,\|\, p(U \mid S, B)\right), \tag{S10}$$

where KL is Kullback–Leibler divergence and I is conditional mutual information. The identity follows from $p(U \mid X, S, B) = p(U \mid X, B)$ because $S = f(X)$. It vanishes exactly when $p(U \mid X, B) = p(U \mid S, B)$ almost surely under the sampled commands. This condition also preserves expectations of integrable evaluations of (U, S, B). The criterion concerns the measured physical targets under the sampled commands. The response loss trains a finite-capacity predictor of their conditional distribution from the complete state S.

Quantization gradients. The Rotation Trick provides a surrogate encoder gradient through hard assignment (Fifty et al., 2025). For a nondegenerate pair, let R_t rotate $e_t/\|e_t\|_2$ to $v_{z_t}/\|v_{z_t}\|_2$ in their spanned plane. It acts as the identity on the orthogonal complement. Define

$$G_t = \operatorname{sg}\left[\frac{\|v_{z_t}\|_2}{\|e_t\|_2} R_t\right], \qquad \widetilde{v}_t = \operatorname{sg}(v_{z_t}) + G_t e_t - \operatorname{sg}(G_t e_t). \tag{S11}$$

The forward value is v_{z_t} and the surrogate Jacobian is G_t. A low-rank implementation uses an identity straight-through fallback for near-zero or nearly antipodal pairs, preserving the selected-code forward value. Fig. 2 denotes this rotation by R_θ; its angle is distinct from the generator parameter θ used here.

The response loss updates the encoder through $\widetilde{v}_t$, while the detached selected vector and rotation block its gradients to the codebook. The fitting penalty updates A, d_0, and commitment updates the encoder. Subsequent dynamics training freezes the encoder, effective codebook, sensor encoding, continuous interface, and normalization. Recorded next-state labels use this frozen interface and are detached.

Joint transition and rotation projection. The routine $\mathrm{Sample}(s_t, b_t)$ first draws $\widehat{z}_{t+1} \sim p_\omega(\cdot \mid s_t, b_t)$ and independent $\epsilon_t \sim \mathcal{N}(0, I)$. The finite sampler in Alg. S1 produces $\widehat{W}_t = G_\theta(s_t, b_t, v_{\widehat{z}_{t+1}}; \epsilon_t)$. After unpacking and rotation projection, it returns the paired $\widehat{Y}_t$ and $\widehat{s}_{t+1} = ((c_t^{\text{static}}, \widehat{c}_{t+1}^{\text{dyn}}), \widehat{z}_{t+1})$. Only fields with identical recorded sources share generated values; physical poses and perception estimates remain separate.

Rotation projection first reverses each field's normalization. For the first two columns a, b of a generated 3×3 matrix, it forms

$$q_1 = \frac{a}{\|a\|_2}, \qquad \bar{b} = b - (q_1^\top b) q_1, \qquad q_2 = \frac{\bar{b}}{\|\bar{b}\|_2}, \qquad R = [q_1, q_2, q_1 \times q_2]. \tag{S12}$$

A norm at most 10^{-6} or a nonfinite input triggers an identity-matrix fallback with zero input derivative. The projected matrix is flattened and returned to the field's original scale before scoring or next-state assembly.

Physics-C uses the joint transition

$$p_\theta^{\mathrm{C}}(W_t, e_{t+1} \mid s_t^{\mathrm{C}}, b_t). \tag{S13}$$

Its factual target appends the detached next history feature to W_t, and its condition contains only (s_t^{C}, b_t). The same sampler jointly generates the dynamic fields and history feature, which form the next state without reconstructing future observations. Continuous gradients propagate through both parts across primitives.

EDM training and sampling. For Physics-VQ, let $x_\sigma = W_t + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ and $\Gamma_t = (s_t, b_t, v_{z_{t+1}})$. The denoiser uses a neural network f_θ, a self-conditioning buffer $\bar{W}$, and EDM preconditioning (Karras et al., 2022):

$$D_\theta(x_\sigma, \sigma, \Gamma_t, \bar{W}) = c_{\mathrm{skip}} x_\sigma + c_{\mathrm{out}} f_\theta(c_{\mathrm{in}} x_\sigma, c_{\mathrm{noise}}, \Gamma_t, \bar{W}), \tag{S14}$$

with coefficients

$$c_{\mathrm{skip}} = \frac{\sigma_{\mathrm{data}}^2}{\sigma^2 + \sigma_{\mathrm{data}}^2}, \qquad c_{\mathrm{out}} = \frac{\sigma\sigma_{\mathrm{data}}}{\sqrt{\sigma^2 + \sigma_{\mathrm{data}}^2}}, \tag{S15}$$
$$c_{\mathrm{in}} = (\sigma^2 + \sigma_{\mathrm{data}}^2)^{-1/2}, \quad c_{\mathrm{noise}} = \tfrac{1}{4}\log\sigma.$$

The generator term in Eq. 3 is

$$\mathcal{L}_{\mathrm{EDM}} = \mathbb{E}\left[\frac{\sigma^2 + \sigma_{\mathrm{data}}^2}{\sigma^2\sigma_{\mathrm{data}}^2}\,\mathrm{MSE}_{\mathrm{valid}}\left(D_\theta(x_\sigma, \sigma, \Gamma_t, \bar{W}), W_t\right)\right], \tag{S16}$$

where $\mathrm{MSE}_{\mathrm{valid}}$ averages valid physical coordinates and all recorded dynamic fields. The next-code label and denoiser condition come from the same recorded next state. Physics-C omits code cross entropy and denoises its augmented continuous target.

The implementation uses the fixed data-scale parameter $\sigma_{\mathrm{data}} = 0.5$ and samples $\log\sigma \sim \mathcal{N}(-1.2, 1.2^2)$. A scalar-token transformer embeds each output coordinate together with its self-conditioning value. For half the factual training samples, a preliminary forward pass sets $\bar{W} = \mathrm{sg}[D_\theta(x_\sigma, \sigma, \Gamma_t, 0)]$; the rest use zero (Chen et al., 2023). The final pass uses the same noisy input and noise level.

Sampling uses $N = 4$ positive levels with $\sigma_{\min} = 0.002$, $\sigma_{\max} = 5$, and $\rho = 7$:

$$\sigma_i = \left[\sigma_{\max}^{1/\rho} + \frac{i}{N-1}(\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho})\right]^\rho, \quad i = 0, \ldots, N-1, \qquad \sigma_N = 0. \tag{S17}$$

Within a primitive, $\Gamma = (s_t, b_t, v_{\hat{z}_{t+1}})$ stays fixed. Predictor and corrector share the incoming buffer, which updates after each interval and resets for each primitive. The final interval uses Euler to avoid denoising at zero noise. Without self-conditioning, every buffer remains zero.

This finite sampler uses $2N - 1 = 7$ denoiser evaluations per sample. For M_{plan} particles and H primitives, the work per candidate is $M_{\mathrm{plan}}H(2N-1)$ scalar-sample evaluations, excluding prior and value networks. Batching affects network invocations and measured latency. T2 differentiates through all sampling updates, self-conditioning buffers, and continuous state assembly, while the factual self-conditioning prepass stays detached.

Algorithm S1 EDM sampling with self-conditioning

Input: Γ, D_θ, $\epsilon \sim \mathcal{N}(0, I)$, noise grid $\sigma_0 > \cdots > \sigma_{N-1} > \sigma_N = 0$.

$x_0 \leftarrow \sigma_0\epsilon$, $\bar{W}_0 \leftarrow 0$
for $i = 0, \ldots, N-1$ **do**
  $D_i \leftarrow D_\theta(x_i, \sigma_i, \Gamma, \bar{W}_i)$; $u_i \leftarrow (x_i - D_i)/\sigma_i$.
  $\Delta_i \leftarrow \sigma_{i+1} - \sigma_i$; $x_{i+1}^{\mathrm{E}} \leftarrow x_i + \Delta_i u_i$.
  if $\sigma_{i+1} > 0$ **then**
    $D_i' \leftarrow D_\theta(x_{i+1}^{\mathrm{E}}, \sigma_{i+1}, \Gamma, \bar{W}_i)$;
    $u_i' \leftarrow (x_{i+1}^{\mathrm{E}} - D_i')/\sigma_{i+1}$.
    $x_{i+1} \leftarrow x_i + \frac{1}{2}\Delta_i(u_i + u_i')$; $\bar{W}_{i+1} \leftarrow D_i'$.
  else $x_{i+1} \leftarrow x_{i+1}^{\mathrm{E}}$, $\bar{W}_{i+1} \leftarrow D_i$.
  end if
end for
return $\widehat{W} = x_N$

Rollout schedule and scoring. Refinement uses $H = 2$ and $M_{\mathrm{roll}} = 4$ independent complete chains per record. At update $k \geq 1$, the scheduled objective is

$$\mathcal{L}_{\mathrm{dyn}}^{(k)} = \mathcal{L}_{\mathrm{fact}} + \delta_k\lambda_2\mathcal{L}_{\mathrm{roll}}, \qquad \delta_k \sim \mathrm{Bernoulli}(q_k), \qquad q_k = \tfrac{1}{4}\min(1, k/2000), \quad \lambda_2 = 1. \tag{S18}$$

The factual term averages the two recorded transitions on every update. The rollout term averages Eq. 4 over the minibatch, with independent code draws and Gaussian noise across complete chains.

Alg. 1 shows an update with $\delta_k = 1$. Here $\ell_{\text{fact}}(j)$ is the factual loss on recorded transition j, and $\text{mean}_{\mathcal{B}}$ averages over minibatch $\mathcal{B}$. Alg. 2 gives the corresponding feedback planner, with learned parameters fixed during execution.

Gradients are clipped to norm 10 before AdamW updates.

Rollout scoring uses all recorded physical coordinates and the non-squared Euclidean norm. The score excludes outcome-dependent masks, and generation runs for the prescribed horizon without success-based gradient truncation. Sensor-validity flags enter the history encoder, while the denoising mask handles physical-target validity.

For predicted and true summary distributions P, Q at the same starting input and commands, let $X, X' \overset{\text{iid}}{\sim} P$ and $Y, Y' \overset{\text{iid}}{\sim} Q$ be independent. Under finite first moments, the energy score is

$$\text{ES}(P, y) = \mathbb{E}\|X - y\|_2 - \tfrac{1}{2}\mathbb{E}\|X - X'\|_2. \tag{S19}$$

Its expected excess over the true distribution equals half the energy distance:

$$\begin{aligned} &\mathbb{E}\,\text{ES}(P, Y) - \mathbb{E}\,\text{ES}(Q, Y) \\ &\quad = \tfrac{1}{2}\left[2\mathbb{E}\|X - Y\|_2 - \mathbb{E}\|X - X'\|_2 - \mathbb{E}\|Y - Y'\|_2\right]. \end{aligned} \tag{S20}$$

This quantity is nonnegative and vanishes exactly when $P = Q$, establishing strict propriety (Gneiting & Raftery, 2007; Pacchiardi et al., 2024). Equation 4 is an unbiased estimator for the finite sampler's summary distribution.

Gradient scope. Fix the representation and write $\vartheta = (\omega, \theta)$ for dynamics parameters. Let $\mathcal{Z}$ collect all categorical draws, ϵ the parameter-independent base noise, and $f(\mathcal{Z}, \epsilon; \vartheta)$ the full particle-batch score. The derivative $\nabla_\vartheta^{\text{path}}$ holds sampled categories fixed. Under differentiability, integrability, and valid interchange of expectation and differentiation (Schulman et al., 2015),

$$\nabla_\vartheta \mathbb{E}[f] = \mathbb{E}\left[\nabla_\vartheta^{\text{path}} f + \sum_{j \in \mathcal{J}} (f - B_j) \nabla_\vartheta^{\text{path}} \log p_\vartheta(Z_j \mid \mathcal{F}_j)\right]. \tag{S21}$$

Here $\mathcal{J}$ indexes categorical draws, $\mathcal{F}_j$ is the information available before draw j, and baseline B_j is measurable with respect to $\mathcal{F}_j$. Zero is a valid baseline. The log-probability derivative includes any dependence on earlier generated continuous states. The coefficient uses the full score because the pairwise distances couple particles. A surrogate retaining the numerical score is

$$\mathcal{A} = \sum_{j \in \mathcal{J}} \text{sg}(f - B_j) \log p_\vartheta(Z_j \mid \mathcal{F}_j), \qquad \widetilde{f} = f + \mathcal{A} - \text{sg}(\mathcal{A}). \tag{S22}$$

T2 uses only the path term, while factual cross entropy trains the code prior. Its rollout gradient omits score terms from categorical draws, including later draws conditioned on earlier generated states. Physics-C with EDM has no categorical draws; mixture-density generators introduce component draws even with continuous states.

Training comparisons and encoder coordination. T0 fits factual transitions. T1a, T1b, T2-stop, and T2 continue from the same paired T0 checkpoint. T1a continues factual training, while T1b fits a direct two-primitive predictor and a success head using real physical outcomes. T1a and T1b match the paired T2 run's measured continuation time. T2-stop uses the same sampling procedure as T2 but detaches each intermediate state before the next transition. These controls compare extra factual updates, direct prediction, and generated-state training with or without continuous cross-primitive gradients. All training conditions use 1,024 evaluation starts per seed.

The T3 variant updates the representation after T2. Each cycle snapshots the system and evaluates factual prediction, independent physical queries, and complete development execution. With the codebook, sensor encoding, continuous interface, normalization, and dynamics frozen, it updates the history encoder and active response heads using

$$\mathcal{L}_{\text{coord}} = \mathcal{L}_{\text{repr}} + \lambda_f \mathcal{L}_{\text{fact}}^{-} + \lambda_r \mathcal{L}_{\text{roll}}. \tag{S23}$$

The weights λ_f, λ_r control consistency terms, and the superscript $-$ denotes next-state labels encoded by the snapshot. Physical targets remain fixed, while frozen dynamics retain input derivatives

under T2's sampled-code restriction. The updated representation is then frozen, recorded histories are re-encoded, and the prior, generator, value, and physical query heads adapt to refreshed labels. Acceptance requires factual error within its preset tolerance, sufficient query improvement, and development success meeting its threshold. A failed cycle restores the complete snapshot; the matched frozen-T2 comparison includes validation, relabeling, adaptation, and rejected work in its time budget.

Continuation value and physical-event evaluation. The continuation value is a sigmoid head fitted to complete-success labels from actual continuations. Its reference controller executes the program's remaining primitives under the fixed low-level controller and remaining budget. Boundary examples assign one after completed success and zero after terminal failure. The default planner averages terminal values using Eq. 5, without a stage reward or discount. The main comparison uses four candidates of horizon $H = 2$ per decision, with $M_{\text{plan}} = 8$ for physical transition models. This deployment count differs from the training count $M_{\text{roll}} = 4$.

The physical-event variant reads generated trajectories in physical units and the given goal. It assigns branch score $C_m + A_m V_g(\widehat{s}^{(m)}_{t+H})$, where C_m indicates completed success and A_m eligibility to continue after success or fatal-event checks. Completed success contributes once, and every sampled branch enters the average.

B Tasks, Data, and Comparison Interfaces

The four main evaluation cells cross seen or unseen object identities with seen or unseen combinations of local transitions. The held-out combinations use component transitions covered during training. Changed-physics and tool-use evaluations have separate conditions and denominators (App. G).

The main learned systems use a common 160k-primitive interaction pool per task. They share observation histories, given geometry and task information, available candidates, and low-level feedback. Representation objectives select different targets from the common records. Simulator physical trajectories are supervision, while estimated poses and other available sensor fields provide decision-time inputs. Physical transition fitting and continuation-value fitting use separate targets and do not update an encoder once its representation stage is frozen. Source-transfer and new-demand adaptation have their own additional-data budgets, stated with their results.

Objects and geometry. Each Geometry task contains 96 objects in 24 families, with four objects per family. Training, development, and unseen testing use 48, 16, and 32 objects from 12, four, and eight families, respectively. Seen-object evaluation uses 32 training objects from eight families. The two action-combination conditions reuse object identities with separate starting scenes. Families reflect design and contact structure, with size changes treated as within-family variation. Table S5 summarizes the geometric variation and task-specific stages.

Primitives and collection. All tasks begin with approach, descent, grasp, and lift, followed by the stages in Table S5. Each primitive uses pose feedback toward a target derived from current observations and given geometry. Ordinary collection follows one precommitted command sequence, while branched collection restores the starting boundary to execute alternatives. Each two-primitive target comes from one executed branch. The supervision comparisons use the same valid-record subset, including physical failures. Usable-data budgets count retained training primitives. Total-acquisition budgets additionally charge prefix and label-collection executions and record scene-restoration overhead.

Table S5: Task geometry and stages following approach, descent, grasp, and lift.

Task	Geometric variation	Subsequent primitives
Peg	Peg dimensions, hole position and clearance	Align above entry, align at entry, insert
Kits	Part contour and matching slot geometry	Align above slot, insert, release, retreat
Poke	Tool length and cross-section, tool–cube contact geometry	Align tool, contact, push, settle

MimicGen data. The external tasks use MimicGen's ThreePieceAssembly_D0 and Threading_D0 environments with Panda control (Mandlekar et al., 2023). Assembly places two parts sequentially into a base, while Threading inserts a needle through a tripod-mounted ring. The official source archives contain ten human demonstrations per task. We extract their environment configuration and object-local geometry, then collect new primitive executions with RGB-D observations and physical trajectory targets. The model reads 256×256 agentview RGB-D images. Two-command programs are fixed before execution, using the same collection protocol as the main tasks. Interaction budgets count these new primitive executions. Each method is trained independently with five seeds and evaluated on 512 new starts per seed using the fixed task assets.

Supervision and common physical predictors. The controlled representation comparisons change the encoder's supervision while retaining the observed fields and representation capacity. REC reconstructs observed history summaries. VIS2 predicts two-primitive future visual features obtained from a fixed target encoder. AFF2 predicts measured binary outcomes of both primitives, such as executability and success. VAL predicts complete task success from the current state, command sequence, and goal. PHY1 and PHY2 predict continuous physical outcomes over one and two primitives, respectively. Each frozen representation receives a newly initialized two-primitive physical predictor with the same architecture and fitting protocol. Generated-state diagnostics use a separately fitted, frozen one-primitive head, shared between the real-state and generated-state queries within that comparison. These predictors are distinct from the representation-training head and the deployed continuation value.

Complete-system baselines. The system baselines use these representations in several different planning procedures. Visual, Affordance, and Physics-C use continuous history features learned with VIS2, AFF2, and PHY2, respectively. They share the EDM transition architecture, self-conditioning, T2 training, and continuation-value planning with Physics-VQ. Their encoders remain frozen during transition training, so the common physical transition loss does not change their original representation supervision. Physics-VQ replaces the continuous history feature with the quantized code described above.

Geometry ranks the common candidates by geometric feasibility and clearance without a learned predictor. Value directly predicts complete success from the current state, entire candidate command sequence, and goal. DFP deterministically predicts the same two-primitive physical measurement vector used by PHY2, trained with mean squared error. A learned success evaluator scores its predicted measurements together with the current state, command sequence, and goal. Thus, DFP also uses physical outcome supervision, although its prediction and planning procedure differ from the generative model. Skill+Q learns an AFF2 representation and skill transitions, then predicts through all but the last command. It scores the last command with a goal-conditioned skill Q head and averages over sampled preceding states.

DINO-WM* and TD-MPC2* are local adaptations evaluated through the common command and task interfaces. DINO-WM* predicts visual and proprioceptive features, with learned terminal-event and continuation heads providing success scores for goal-conditioned planning. TD-MPC2* encodes the current observation and evaluates candidate sequences using its latent dynamics, reward model, and value model. These adaptations compare alternative planning systems, while the frozen-representation controls isolate the effect of representation supervision.

C Statistical Analysis and Primary Contrasts

Main geometry evaluations contain eight families, four objects per family, and 32 starting scenes per object. The primary 160k system comparisons use ten seeds for Physics-VQ, Physics-C, Value, Affordance, and the selected Kits TD-MPC2 adaptation. Other learned systems use five seeds, and geometry has no training randomness. Source-transfer evaluations use six families, two targets per family, and 256 starts per target, crossed with five seeds. Different target budgets reuse the same target identities. Table S6 reports paired confidence intervals and multiplicity-adjusted tests for the primary contrasts.

Means weight families, objects, starts, and training seeds equally at their respective levels. We use 20,000 paired bootstrap resamples of families, objects within families, and starts within objects, with

a separate crossed training-seed axis. Methods share the resampling indices, and repeated queries are averaged within a start. The reported intervals are individual 95% percentile intervals, while Holm adjustment retains all 30 specified comparison positions. The two query comparisons report descriptive ES gaps and confidence intervals. Their entries use $p = 1$ in the 30-comparison Holm adjustment. Query intervals are conditional on the available reference draws.

The additional finite-cluster sensitivity analysis uses two-way family-by-seed variance and a t approximation with $\min(F-1, S-1)$ degrees of freedom. Here F and S count families and training seeds. Six of nine information-retention contrasts remain below the corrected 0.05 threshold, and all nine retain their favorable direction. All four extra source gains over visual supervision remain positive but fail this adjusted sensitivity test.

Table S6: Primary contrasts. Effects are in energy-score units for information and next-query comparisons, and percentage points otherwise. Finite p is the adjusted finite-cluster sensitivity result; dashes denote descriptive comparisons without hypothesis tests.

Question	Task	Contrast	Effect	95% interval	Holm p	Finite p
Information	Peg	PHY2-AFF2	-0.044	[-0.059, -0.031]	< 0.002	0.065
Information	Peg	PHY2-VAL	-0.061	[-0.078, -0.045]	< 0.002	0.038
Information	Peg	PHY2-VIS2	-0.075	[-0.095, -0.055]	< 0.002	0.038
Information	Kits	PHY2-AFF2	-0.040	[-0.047, -0.033]	< 0.002	0.005
Information	Kits	PHY2-VAL	-0.061	[-0.083, -0.043]	< 0.002	0.066
Information	Kits	PHY2-VIS2	-0.083	[-0.097, -0.068]	< 0.002	0.007
Information	Poke	PHY2-AFF2	-0.043	[-0.057, -0.028]	< 0.002	0.068
Information	Poke	PHY2-VAL	-0.052	[-0.068, -0.040]	< 0.002	0.038
Information	Poke	PHY2-VIS2	-0.074	[-0.093, -0.059]	< 0.002	0.021
Transfer	Peg	Source gain, n100	+9.440	[+6.725, +12.064]	< 0.002	0.013
Transfer	Peg	Extra gain, n100	+5.547	[+2.194, +8.763]	0.019	0.102
Transfer	Peg	Source gain, n500	+8.340	[+5.540, +11.087]	< 0.002	0.013
Transfer	Peg	Extra gain, n500	+5.625	[+1.829, +9.343]	0.062	0.090
Selection	Peg	Response + geometry	+0.371	[-0.430, +1.973]	1.000	1.000
Next query	Peg	Generated − reference	+0.011	[+0.007, +0.015]	–	–
Transfer	Kits	Source gain, n100	+6.667	[+3.060, +10.085]	< 0.002	0.148
Transfer	Kits	Extra gain, n100	+3.984	[+0.775, +7.311]	0.190	0.253
Transfer	Kits	Source gain, n500	+5.221	[+1.653, +8.887]	0.065	0.253
Transfer	Kits	Extra gain, n500	+3.372	[+0.280, +6.524]	0.282	0.253
Selection	Kits	Response + geometry	+1.250	[-0.567, +3.353]	1.000	0.493
Next query	Kits	Generated − reference	+0.011	[+0.008, +0.015]	–	–
System	Peg	VQ − Affordance	+6.895	[+3.877, +9.844]	< 0.002	0.007
System	Peg	VQ − Value	+10.957	[+7.842, +14.287]	< 0.002	< 0.002
System	Peg	VQ − Physics-C	+3.760	[+0.576, +6.943]	0.195	0.179
System	Kits	VQ − TD-MPC2*	+4.297	[+0.967, +7.637]	0.135	0.112
System	Kits	VQ − Value	+11.084	[+7.646, +14.492]	< 0.002	< 0.002
System	Kits	VQ − Physics-C	+0.195	[-3.682, +4.014]	1.000	1.000
System	Poke	VQ − Affordance	+1.963	[-0.928, +4.834]	1.000	0.543
System	Poke	VQ − Value	+7.275	[+4.473, +10.264]	< 0.002	0.004
System	Poke	VQ − Physics-C	+0.127	[-2.695, +2.891]	1.000	1.000

D SUPERVISION, STATE, AND DATA ALLOCATION

These controls distinguish supervision, encoder adaptation, state structure, and data allocation. Unless stated otherwise, success is reported in percent as mean ± training-seed SD.

Frozen and adapted representations. The common physical predictor evaluates information accessible from the frozen representation. Table S7 uses a width-128 head, continuous history features, 160k pretraining primitives, and five seeds. Each query distribution contains 1,024 starts with four queries per start.

At 160k primitives, widening this head to 256 units preserves PHY2's lower mean ES than AFF2 on held-out actions: 0.576 versus 0.628 on Peg, 0.543 versus 0.579 on Kits, and 0.495 versus 0.533 on Poke.

Allowing the encoder to update with shared physical supervision can narrow or reverse a frozen comparison. Table S8 uses continuous history features, 160k primitives, five seeds, and 1,024 starts per seed. For example, fine-tuned AFF2 reaches 76.4% on Poke, compared with 74.4% for PHY2.

Table S7: Frozen-representation prediction across query distributions. Entries are mean energy scores over five seeds; lower is better.

	Peg			Kits			Poke		
Target	New starts	Actions	Families	New starts	Actions	Families	New starts	Actions	Families
REC	0.740	0.746	0.744	0.689	0.674	0.679	0.672	0.672	0.666
VIS2	0.652	0.656	0.654	0.639	0.628	0.633	0.573	0.572	0.567
VAL	0.637	0.642	0.641	0.618	0.606	0.611	0.551	0.550	0.545
AFF2	0.620	0.625	0.623	0.596	0.585	0.590	0.541	0.541	0.535
PHY1	0.604	0.609	0.608	0.585	0.575	0.580	0.533	0.533	0.528
PHY2	0.575	0.581	0.580	0.553	0.545	0.550	0.499	0.498	0.494

Table S8: Frozen versus fine-tuned encoders under common physical supervision. Success (%) is mean ± training-seed SD.

	Peg		Kits		Poke	
Target	Frozen	Fine-tuned	Frozen	Fine-tuned	Frozen	Fine-tuned
REC	61.2 ± 1.7	67.2 ± 1.8	47.9 ± 1.4	57.8 ± 1.5	65.2 ± 2.3	72.8 ± 1.2
VIS2	64.7 ± 2.3	69.1 ± 2.1	54.2 ± 3.0	59.6 ± 2.1	69.9 ± 2.1	74.4 ± 1.7
VAL	68.1 ± 2.0	67.7 ± 1.2	58.1 ± 1.5	61.8 ± 2.2	72.3 ± 2.1	73.9 ± 0.4
AFF2	68.3 ± 1.5	68.9 ± 1.3	59.5 ± 2.4	64.2 ± 2.4	72.5 ± 1.5	76.4 ± 1.7
PHY1	70.0 ± 2.4	70.4 ± 1.2	62.0 ± 0.6	63.6 ± 1.4	75.5 ± 1.4	76.0 ± 2.9
PHY2	70.7 ± 1.2	71.1 ± 1.7	66.4 ± 1.9	65.4 ± 2.2	76.0 ± 1.5	74.4 ± 1.2

State and generator controls. Table S9 holds the state and generator fixed while changing supervision. The generators are a mixture-density network (MDN) and EDM with self-conditioning (EDM-SC). Physical supervision remains useful across both state types and both generators.

Table S9: Supervision across state types and generators at 160k primitives. Success (%) is mean ± SD over five seeds. The last two rows give paired PHY2-minus-AFF2 differences (pp) and their SDs.

		Peg		Kits	
Target	State	MDN	EDM-SC	MDN	EDM-SC
VIS2	C	63.5 ± 1.6	64.7 ± 2.3	52.5 ± 1.1	54.2 ± 3.0
VIS2	VQ	65.0 ± 2.4	64.6 ± 1.2	54.6 ± 1.5	54.6 ± 1.2
VAL	C	66.1 ± 0.9	68.1 ± 2.0	58.3 ± 0.9	58.1 ± 1.5
VAL	VQ	67.2 ± 1.4	68.9 ± 1.8	57.6 ± 2.2	58.2 ± 1.9
AFF2	C	69.0 ± 2.3	68.3 ± 1.5	59.3 ± 3.0	59.5 ± 2.4
AFF2	VQ	66.9 ± 1.0	67.7 ± 1.3	57.6 ± 2.1	58.5 ± 2.9
PHY2	C	70.3 ± 1.7	70.7 ± 1.2	63.8 ± 2.6	66.4 ± 1.9
PHY2	VQ	73.9 ± 0.5	75.7 ± 0.9	64.5 ± 1.6	64.2 ± 0.7
PHY2 − AFF2	C	$+1.35 \pm 1.75$	$+2.44 \pm 2.38$	$+4.49 \pm 4.74$	$+6.91 \pm 2.71$
PHY2 − AFF2	VQ	$+6.99 \pm 1.31$	$+7.99 \pm 1.45$	$+6.95 \pm 2.25$	$+5.68 \pm 3.00$

Table S10 compares state implementations with physical supervision, generator family, and training held fixed. A continuous history feature is already competitive on Kits, while the selected SimVQ + Rotation variant improves Peg. The alternatives include finite scalar quantization (FSQ) (Mentzer et al., 2024) and standard VQ, distinguished here from the SimVQ implementation used by Physics-VQ.

With a 512-dimensional continuous history feature and hidden width 512 at 160k primitives, PHY1 exceeds PHY2 in mean success on Peg (73.7% versus 71.7%) and Poke (77.4% versus 76.8%).

Interaction budget and collection. Fig. 3 fixes continuous history features, EDM-SC, and T2 while varying supervision and interaction budget. Fig. S7 instead compares complete systems on

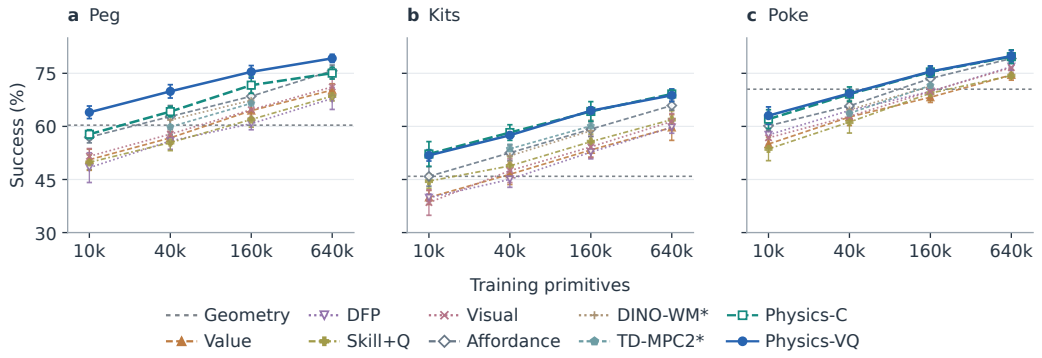

Figure S7: Complete-system data-budget trends on unseen objects and action combinations. Bars show one training-seed SD. Geometry is a fixed reference; local adaptations have two measured budgets.

unseen objects and unseen action combinations across budgets. Its 160k points retain the main system comparison's seed counts; other learned points use five seeds.

Fig. S8 separates the supervision horizon from data collection. Ordinary trajectories supply one command sequence per starting history, while branched collection supplies four. The command-coverage comparison either fixes total acquisition or fixes the starting histories and eight records per history. At fixed total acquisition, each distinct command is executed once. At fixed histories and records, repetitions per command equal eight divided by the number of distinct commands. All settings use frozen continuous history features, EDM-SC, T2, five training seeds, and 1,024 evaluation starts per seed. For the common width-128 prediction head, four queries are averaged within each evaluation start.

Table S10: State choices under the same PHY2 target, EDM-SC generator, and T2 training. Entries report success (%) over five seeds at 160k primitives; quantized history features use 256 codes.

State	Peg mean	Peg SD	Kits mean	Kits SD
Observed fields only	61.7	1.7	55.0	1.3
Continuous history	70.7	1.2	66.4	1.9
Soft mixture	69.7	2.5	62.8	2.8
FSQ	69.7	1.0	62.0	1.1
Standard VQ	67.9	2.5	61.7	2.2
Standard VQ + Rotation	69.4	1.1	63.7	2.3
SimVQ	69.2	1.1	61.7	3.1
SimVQ + Rotation	75.7	0.9	64.2	0.7

PHY2 improves mean success on Kits in every matched collection setting, whereas the smaller Peg differences sometimes reverse. With starting histories fixed, replacing repeats with distinct commands improves mean success on both tasks. With total acquisition fixed, eight commands give no further success improvement over four. The useful number of commands therefore depends on how interaction is allocated.

E Transfer and Adaptation

Cross-object source transfer. Source transfer uses encoders pretrained on other objects while matching target data for downstream training (Fig. 5).

Source A and Source B each provide 160k primitives from designated pools that exclude the target object. Their encoders train on source records only and freeze before any target-side learning. Target-only representations train on the n available target primitives, and the repeated-data control adds updates on those same records. The more-target representation instead uses $n + 160\text{k}$ target primitives. We then freeze each representation and fit fresh common physical predictors, factual dynamics, and continuation values on the same nested n-primitive target subset. Thus, the additional source or target data affect representation learning, while downstream data remain matched.

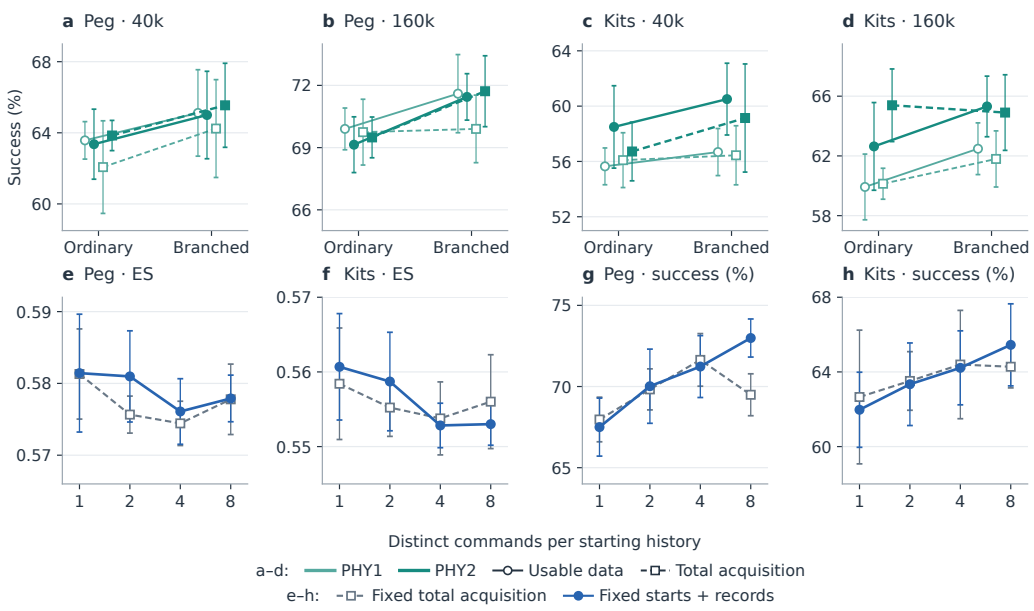

Figure S8: Supervision horizon and data allocation (mean ± SD, five seeds). (a–d) Success under matched usable-data or total-acquisition budgets. (e–h) PHY2 query ES and success with increasing command coverage at 160k.

Source-pool composition. For Peg and Kits, eight source pools group experience by geometry, contact structure, and program similarity. Each source pool combines two 80k-primitive blocks. Source A and Source B share no records, while neighboring pools overlap by half, giving 640k distinct source primitives per task.

These fixed source-pool comparisons are separate from the source-selection test, which chooses among eight pools for each target. The source selector is fitted by cross-validation on development families, with source choices fixed before test evaluation. Adding response information to the geometry selector changes mean success by +0.37 pp on Peg and +1.25 pp on Kits. The individual 95% intervals are $[-0.43, 1.97]$ and $[-0.57, 3.35]$ pp, respectively, with Holm-adjusted $p = 1$ for both.

Table S11: PHY2 source transfer with n target primitives. Encoders remain frozen (Frozen) or adapt to target data (Full). Success (%) is mean ± SD over five seeds, averaging 12 targets per seed with 256 starts per target.

			Target only		Source A	
Task	n	State	Frozen	Full	Frozen	Full
Peg	100	C	43.5 ± 1.0	43.1 ± 1.7	52.9 ± 0.9	52.2 ± 0.4
Peg	100	VQ	44.6 ± 0.6	43.8 ± 0.7	54.3 ± 0.8	53.5 ± 1.0
Peg	500	C	54.2 ± 0.4	55.1 ± 1.3	62.6 ± 1.3	63.9 ± 1.3
Peg	500	VQ	55.3 ± 0.8	55.8 ± 0.8	64.0 ± 0.4	65.1 ± 0.9
Kits	100	C	42.2 ± 0.7	42.4 ± 0.7	48.9 ± 0.8	48.3 ± 0.7
Kits	100	VQ	41.5 ± 0.3	40.6 ± 0.8	48.4 ± 0.7	48.0 ± 0.3
Kits	500	C	53.3 ± 0.7	54.4 ± 0.9	58.5 ± 0.7	59.9 ± 0.6
Kits	500	VQ	52.4 ± 0.8	53.5 ± 1.0	58.5 ± 0.9	59.8 ± 1.1

The source-transfer control additionally permits target-side representation adaptation (Full) before fitting fresh downstream modules on the same n target primitives (Table S11).

Source A retains its mean advantage over target-only training under both adaptation modes.

New demands and adaptation. The new-demand evaluation retains the task parameterization available during training (Table S12). The main comparison tests unseen objects and action combinations under a fixed task requirement. Here the requirement changes, and the adaptation levels specify which learned modules may update. Goal demands change target poses or progress requirements. Program demands change the combinations, order, or parameters of existing primitives. Geometry demands change object, path, or tool constraints. The goal enters the evaluator and con-

tinuous context, while the program enters candidate command sequences with their stage identifiers. Candidate generation uses the existing primitive templates, current observations, and supplied demand fields. The low-level controller stays fixed.

L0 loads the pretrained system unchanged and uses no new data. L1 updates only the high-level evaluator: the continuation value for physical, visual, and affordance models, the direct-value head for Value, the skill-Q head for Skill+Q, and the measurement evaluator for DFP. L2 uses the same new-data budget but also adapts the representation and prediction modules. For physical models, representation adaptation precedes a renewed freeze, factual dynamics fitting, and value fitting. Adaptation records use demand-specific goals and success definitions and are separate from evaluation starts. Budgets count all primitive executions per demand, including those needed for complete-continuation labels.

Table S12: New-demand success (%; five-seed means). Frozen columns use no target interaction; adaptation columns use 100 or 500 primitives per new program.

			Frozen		High-level adaptation		Full adaptation	
Task	Method	Goal	Program	Geometry	100	500	100	500
Peg	Value	65.2	53.6	46.1	56.4	59.1	56.2	60.4
Peg	DFP	61.7	50.4	42.1	54.2	56.8	54.8	58.9
Peg	Skill+Q	61.9	50.1	43.2	54.2	57.1	55.7	59.1
Peg	Affordance	68.8	56.8	49.6	61.9	64.4	61.3	65.5
Peg	Physics-C	70.5	62.0	53.8	66.9	69.0	66.9	69.9
Peg	Physics-VQ	74.6	67.5	58.7	70.9	73.3	71.7	75.4
Kits	Value	53.2	45.1	38.7	48.8	50.6	48.1	50.7
Kits	DFP	52.7	44.1	37.4	47.6	50.4	47.4	51.4
Kits	Skill+Q	56.6	48.2	42.4	50.3	53.9	50.8	54.6
Kits	Affordance	58.6	50.1	44.3	54.5	56.7	55.2	57.5
Kits	Physics-C	65.4	60.7	53.8	61.8	65.5	62.7	65.0
Kits	Physics-VQ	62.1	57.6	51.9	61.6	64.4	62.0	64.6
Poke	Value	68.8	61.0	53.0	63.2	64.4	63.3	67.3
Poke	DFP	69.7	62.3	54.9	64.2	68.0	65.1	68.4
Poke	Skill+Q	69.2	61.4	53.4	61.7	64.8	63.5	65.8
Poke	Affordance	71.5	64.0	55.3	66.4	71.0	67.7	71.7
Poke	Physics-C	74.1	70.1	61.9	72.9	75.6	72.3	76.8
Poke	Physics-VQ	73.1	68.0	60.9	72.0	75.3	73.0	76.5

Fig. S9 extends the program comparisons in Table S12 to all three demand categories and four adaptation budgets. Within each seed, we average success across four demands before computing the paired L2-minus-L1 difference. Each demand has 256 evaluation starts. Positive differences favor updating representation and dynamics as well as high-level evaluation. The largest observed gains occur under geometry changes, while gains for goals and programs are smaller and sometimes negative.

F Prediction through Generated States

Joint two-primitive queries. The two-primitive query test scores a joint six-coordinate physical summary, with three coordinates per primitive. In expectation, joint ES can distinguish cross-primitive dependence errors within this summary even when the single-primitive marginals agree. It uses 1,024 starting scenes, four held-out subsequent-command queries per scene, and five training seeds. Each prediction combines 32 first-primitive branches with two subsequent samples per branch from the same frozen one-primitive physical head. The model samples its first branches from the transition generator. The marginal reference uses independently repeated real first-primitive outcomes from the same restored physical state and commands, varying future execution randomness. Each branch pairs its first physical summary with the subsequent head prediction.

Both distributions are scored against a separate recorded joint outcome, excluding the reference executions from that scoring record. Energy-score pair terms use different outer branches because

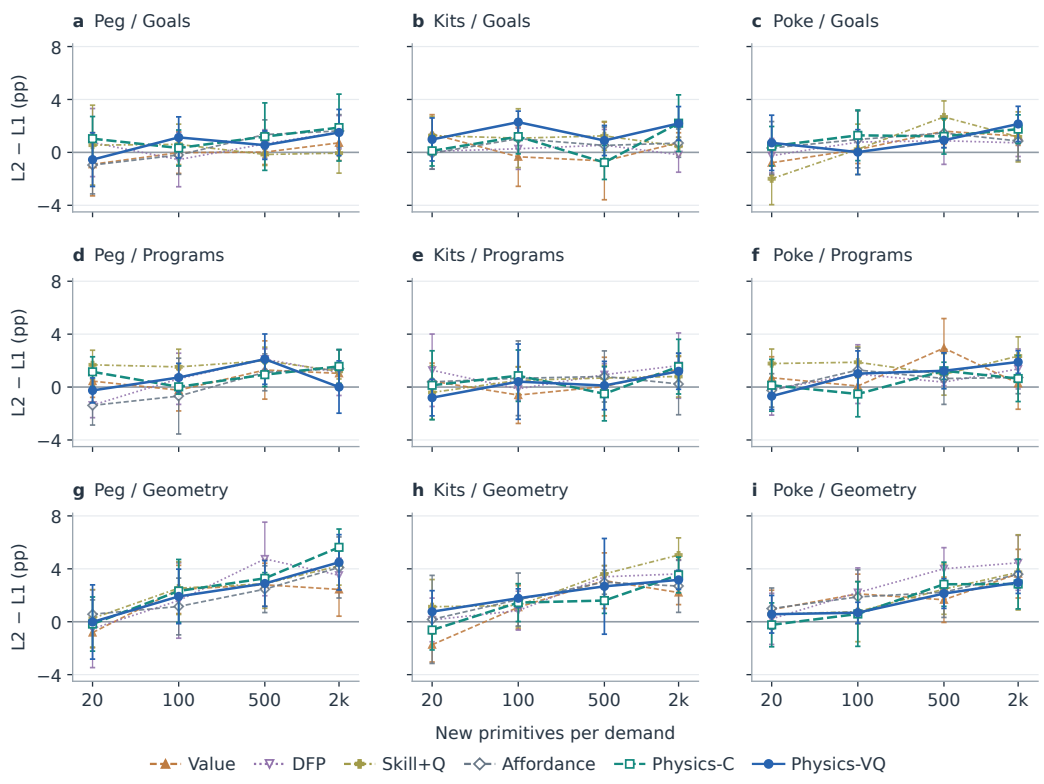

Figure S9: Success gain from full (L2) over high-level (L1) adaptation, in percentage points. Points and bars show the mean and one SD of paired differences across five seeds. Zero indicates equal success.

the two inner continuations share an intermediate state. The observed-fields-only control fits a width-128 two-primitive head from continuous observation fields and both commands. The action-only control sees only those commands, and neither control reads the history representation.

Table 3a reports the joint query scores and generated-minus-reference gaps. The joint scores include first-action prediction and error from the common query head.

Third-primitive queries. This diagnostic uses 512 starting scenes, four queries per scene, and five seeds. It scores a three-coordinate final-primitive summary against a reference that observes the real second state. Peg ES increases from 0.3820 to 0.4085 (+0.0265, 6.95%), and Kits from 0.3696 to 0.3917 (+0.0222, 6.00%). These final-primitive scores concern a different summary from the joint two-primitive test.

Categorical sampling gradients. For discrete sampling, Table S13 adds a score-function gradient to the categorical draw while retaining continuous cross-primitive gradients. Its scoring changes are small and depend on the task and codebook size. The variogram score averages squared discrepancies between observed square-root absolute differences and their predicted expectations across the three corresponding coordinate pairs. The Brier score measures squared probability error for attaining the final primitive's physical progress threshold, indicated by a nonnegative margin. Its predicted probability is the event frequency among eight generated evaluation samples, distinct from the two particles used for training. Each score is averaged over four queries per starting scene before aggregation across scenes and seeds.

Table S13: Categorical-gradient diagnostic with small codebooks. Scores are means ± training-seed SD for five seeds, T2, 160k primitives, and two training particles. Lower is better.

Task	Codes	Categorical gradient	Energy	Variogram	Brier
Peg	4	Stopped	0.591 ± 0.002	0.068 ± 0.001	0.128 ± 0.001
Peg	4	Score function	0.590 ± 0.005	0.068 ± 0.001	0.128 ± 0.002
Peg	8	Stopped	0.589 ± 0.007	0.068 ± 0.001	0.128 ± 0.001
Peg	8	Score function	0.586 ± 0.006	0.067 ± 0.001	0.128 ± 0.003
Kits	4	Stopped	0.565 ± 0.010	0.060 ± 0.002	0.116 ± 0.002
Kits	4	Score function	0.568 ± 0.004	0.060 ± 0.000	0.116 ± 0.001
Kits	8	Stopped	0.563 ± 0.004	0.059 ± 0.000	0.115 ± 0.001
Kits	8	Score function	0.563 ± 0.004	0.059 ± 0.001	0.115 ± 0.001

Generator outputs and path requirements. Table S14 adds terminal holding (Hold) and contact safety throughout execution (Process) to ordinary task success. Hold checks insertion depth, lateral and angular tolerances, and safety over consecutive terminal control steps. Process checks summed contact-force magnitudes against task-specific, asset-scaled safety limits at every executed control step. Monitored contacts involve the peg and hole fixture in Peg, and the part with the tray, slot, or assembled components in Kits. Contact within these limits is allowed, without an additional holding requirement. Both criteria use actual execution trajectories, independently of absorbing extensions in the prediction target.

Trajectory predictors improve Hold and Process success over the endpoint mixture, while ordinary success remains similar. Event targets recover part of these gains, and self-conditioning has mixed effects. This comparison varies prediction targets and generators under terminal-value planning.

Table S14: Generator comparison with a frozen Physics-VQ encoder and factual transition training. Success (%) is mean ± SD over five seeds at 160k primitives, with VQ states and 512 starts per condition.

	Peg			Kits		
Generator	Ordinary	Hold	Process	Ordinary	Hold	Process
Endpoint mixture	74.2 ± 0.7	56.7 ± 1.2	56.4 ± 1.3	64.5 ± 1.7	49.9 ± 2.5	49.5 ± 2.4
Endpoint + events	74.6 ± 1.4	60.5 ± 1.6	62.7 ± 2.5	62.6 ± 2.2	53.1 ± 1.8	54.0 ± 2.1
Trajectory mixture	75.4 ± 1.4	61.6 ± 2.0	61.3 ± 1.7	62.7 ± 2.3	53.6 ± 2.2	53.9 ± 1.6
EDM	75.1 ± 1.4	63.4 ± 2.5	64.0 ± 2.1	63.4 ± 1.3	55.4 ± 2.0	56.4 ± 2.3
EDM-SC	74.8 ± 2.2	61.4 ± 4.2	63.2 ± 3.6	64.6 ± 0.9	54.9 ± 1.3	55.8 ± 0.8

Training and encoder coordination. Table 3b reports the 160k training comparison and paired T2-minus-T2-stop differences. Table S15 extends this comparison to 40k primitives.

Table S15: Training at 40k primitives (success %, mean ± SD; five seeds). Δ is T2 − T2-stop (pp, paired-seed SD). Gray columns train through generated states. C/VQ denotes continuous/quantized history features.

Task (state)	T0	T1a	T2-stop	T2	Δ
Peg (C)	62.5 ± 1.4	62.7 ± 2.4	63.9 ± 2.3	**64.2** ± 1.5	$+0.3 \pm 2.2$
Peg (VQ)	67.1 ± 1.5	67.2 ± 2.7	69.3 ± 1.7	**69.9** ± 1.9	$+0.6 \pm 1.4$
Kits (C)	55.1 ± 1.9	57.1 ± 2.4	**58.3** ± 2.9	**58.3** ± 2.2	-0.1 ± 0.9
Kits (VQ)	55.2 ± 1.0	56.4 ± 1.6	57.2 ± 1.1	**57.5** ± 0.8	$+0.3 \pm 1.2$

T2 versus T2-stop isolates continuous cross-primitive gradients, whereas the comparison with T1b also changes prediction structure. Table S16 evaluates encoder coordination (App. A) against an equal-time continuation of rollout training. Its mixed results support retaining T2 as the reference model.

Table S16: Encoder coordination versus equal-time rollout training. Success (%) is mean ± SD over five seeds at 160k primitives.

Task	State	More rollout training	Coordination	Difference (pp)
Peg	C	71.3 ± 1.2	70.7 ± 1.5	-0.66
Peg	VQ	74.4 ± 1.8	75.7 ± 0.9	+1.33
Kits	C	65.8 ± 1.2	66.1 ± 2.3	+0.29
Kits	VQ	65.1 ± 0.9	65.7 ± 1.4	+0.57

G Runtime, External Tasks, and Failure Boundaries

Candidate choice. The candidate-choice diagnostic holds the continuation controller fixed (Table S17). Light physical scores candidate continuations with a frozen continuous-state physical predictor and learned success evaluator. The observed-history reference uses only available histories, whereas the privileged reference also accesses the simulator state.

Each diagnostic start has four candidates, each with 32 estimation executions and 32 independent verification executions. All six selectors share those candidates and verification records. The selected-candidate denominator is 8,192 verifications per task and selector, clustered within 256 starts; this diagnostic has no training-seed axis.

Table S17: Candidate-choice diagnostic under a common continuation controller. Success (%) averages independent verification of the selected candidate over all 256 starting scenes per task.

Selector	Peg	Kits	Poke
Geometry	72.01	64.89	78.03
Value	74.17	66.53	78.49
DFP	74.57	65.64	78.78
Light physical	73.73	67.22	77.44
Observed-history reference	75.52	67.76	79.88
Privileged reference	74.91	68.14	79.46

The observed-history reference exceeds geometry by 3.52, 2.87, and 1.86 pp on Peg, Kits, and Poke, with smaller differences from Value and DFP. These finite-sample references show scope for improving candidate choice under the shared controller.

Runtime. The runtime comparison uses a separate five-seed evaluation of deployment configurations under a common decision-time limit. Runtime evaluations use one NVIDIA H200 GPU per run. As in the main comparison, the physical-model scorer uses $M_{\text{plan}} = 8$, $H = 2$, and $N = 4$, giving 112 scalar-sample denoiser evaluations per candidate. This count excludes candidate search and prior/value-network computation. In Table S18, SR@100 is the fraction of executions that succeed and whose recorded decisions all meet the cap.

Table S18: Runtime under a 100 ms decision cap on U/U, with 1,024 starts per seed. SR denotes success (%); p95 is mean per-run 95th-percentile decision latency (ms), over five seeds.

	Peg			Kits			Poke		
Method	SR	SR@100	p95	SR	SR@100	p95	SR	SR@100	p95
Value	64.5	64.5	23.8	54.0	54.0	23.8	69.8	69.8	23.9
Affordance	67.5	65.7	79.1	58.7	57.1	79.7	72.1	70.3	79.2
DINO-WM*	67.0	65.0	80.0	57.8	56.5	79.9	70.7	69.0	79.7
TD-MPC2*	66.0	64.4	79.6	59.2	57.6	79.7	70.8	69.1	79.9
Physics-C	69.9	68.2	79.8	65.4	64.0	78.6	74.2	72.5	79.6
Physics-VQ	75.0	73.2	79.6	63.5	62.1	79.6	74.9	72.9	79.9

Changed operating conditions. Fig. S10 compares nominal execution (Nom), new command combinations (Comb), changed physics (Phys), changed viewpoints (View), and stronger low-level feedback (Feed). The decline under changed physics is consistent with reuse depending on learned transition conditions, even when new task requirements can use frozen dynamics. The Poke tool-use

check requires tool–target contact and sufficient tool-attributed displacement while limiting direct robot pushing, together with the specified terminal task criterion. Its task-placement and stricter tool-use scores use the same tool-test trials.

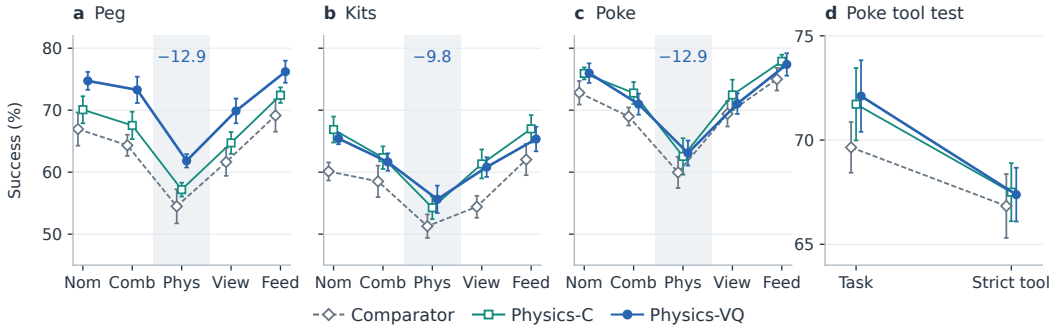

Figure S10: Success under changed conditions (mean ± SD, five seeds, 512 starts per seed and condition). (a–c) Lines connect discrete conditions. Shading marks changed physics, with VQ's change from nominal annotated in blue (pp). (d) Task-placement and strict-tool scores use the same Poke trials, with a tighter success axis. Comparator: TD-MPC2* on Kits, Affordance otherwise.

External tasks and robot evaluation. Table S19 reports unmodified simulator tasks, separately retrained MimicGen tasks, and the real peg assembly counts visualized in Fig. 6.

Table S19: External evaluation. Simulator entries are mean success (%) over five seeds. PickCube uses 40k primitives and other simulated conditions use 160k. Robot columns report successes out of 480 trials.

	Official simulator tasks (%)				MimicGen (%)		Robot successes	
Method	Pick	Peg	Kits	Poke	Assembly	Threading	Frozen	Adapted
Geometry	96.7	69.9	45.8	73.4	–	–	–	–
Value	88.7	68.2	50.4	71.5	49.8	76.8	238/480	268/480
Affordance	90.0	76.0	58.1	74.0	50.7	81.0	257/480	288/480
Physics-C	90.7	77.2	58.0	78.3	56.8	84.5	255/480	290/480
Physics-VQ	91.9	79.0	58.9	79.1	59.6	87.9	294/480	304/480

Real-world peg assembly experiments use a UFACTORY xArm 6 robotic arm. All four robot methods start from their corresponding development-selected PegInsertionSide-Geometry model trained on 160k simulation primitives.

Common calibration of geometry and sensors precedes deployment, using no target behavioral outcomes. No additional real-data training of the representation or physical dynamics precedes either protocol. The frozen robot protocol keeps all learned modules fixed. The adapted protocol starts from the same deployment model and updates only the evaluator using 100 new primitives per object from a common 1,200-primitive pool shared across methods. Each method and protocol has 480 trials: six families, two objects per family, four sessions, and ten repetitions per object-session cell. These trials have no training-seed axis. Robot intervals use 20,000 paired bootstrap

Table S20: Paired robot success differences (percentage points) and individual 95% confidence intervals (CIs). Both protocols use 480 trials per method.

	Frozen (pp)		Adapted (pp)	
Comparison	Difference	95% CI	Difference	95% CI
VQ minus Value	+11.7	[2.3, 21.0]	+7.5	[-1.7, 15.8]
VQ minus Affordance	+7.7	[-2.5, 18.3]	+3.3	[-6.2, 13.3]
VQ minus Physics-C	+8.1	[-2.5, 18.5]	+2.9	[-6.9, 13.3]

resamples of families, objects within families, and crossed sessions, followed by repetitions within each object–session cell. The six individual 95% intervals are exploratory and are outside the 30-contrast Holm family.

H Extended Related Work

Predictive knowledge and state sufficiency. The successor representation describes a state through expected discounted future state occupancy, allowing values to be computed for different rewards under the same policy (Dayan, 1993). Horde extends predictive knowledge to many general value functions, each specifying a policy, termination condition, and prediction target (Sutton et al., 2011). Predictive-state theory describes dynamical systems through predictions of action-observation tests (Singh et al., 2004), and spectral learning connects such representations to planning from sampled interaction sequences (Boots et al., 2011). Our training queries concern physical measurements over specified primitive sequences. The information criterion in App. A therefore concerns the measured targets and sampled commands. Whether generated states support further predictions is tested separately through subsequent primitives and feedback execution.

Interaction features and visual planning. Learning to Poke jointly trains forward and inverse models from object interactions, using action prediction to shape visual features (Agrawal et al., 2016). Curiosity-driven exploration uses inverse-dynamics features to define an intrinsic reward from forward-prediction error (Pathak et al., 2017). Causal InfoGAN couples a generative observation model to a low-dimensional planning representation (Kurutach et al., 2018). Visual Foresight learns action-conditioned video prediction from robot experience and uses it for model-predictive control across goals and objects (Ebert et al., 2018). We study the supervision target itself by comparing physical measurements with visual, feasibility, and value targets under matched downstream learning. The physical outcomes supervise the encoder during training, while execution uses observed history and task context.

Trajectory generation and imagined control. Diffuser generates state-action trajectories and guides denoising toward desired behavior (Janner et al., 2022). Dreamer learns behavior by propagating value gradients through imagined latent trajectories (Hafner et al., 2020). Our generator conditions on candidate primitive commands and jointly samples physical outcomes and the next state. A continuation value evaluates sampled terminal states, and execution receives real observations after each selected primitive. The encoder remains fixed while dynamics and the continuation value are learned, supporting the comparison of representation supervision with common downstream targets.

Training with generated context. Scheduled Sampling exposes recurrent predictors to their own outputs during training (Bengio et al., 2015). Professor Forcing uses an adversarial objective to align hidden-state and output dynamics between teacher-forced and freely sampled sequences (Goyal et al., 2016). Self Forcing brings self-generated context and sequence-level distribution matching to autoregressive video diffusion (Huang et al., 2025). Our two-primitive training shares the concern that later predictions must work from generated context. Its loss scores the joint distribution of complete physical outcome chains from a recorded starting state, allowing each sampled intermediate state to remain paired with its subsequent outcome. The rollout comparisons separate exposure to generated states from gradients propagated through them.