LIBERO-JEPA: Learning Action-Grounded Dynamics in Object-Centric World Models

Thomas Deng
Stanford
450 Jane Stanford Way
tdeng23@stanford.edu

Zubin Carvalho
Stanford
450 Jane Stanford Way
zubinc@stanford.edu

Abstract

We investigate whether object-centric world models produce more semantically grounded predictions than action-agnostic alternatives in a robotic manipulation setting. We present LIBERO-JEPA, a temporal object-centric world model built on the Joint-Embedding Predictive Architecture that operates over object-level slot representations produced by a frozen VideoSAUR encoder. A transformer-based predictor is trained to roll slots forward in time under random object-slot masking, conditioned on the robot's action sequence. We find that standard training recipes, including Hungarian MSE loss, future action conditioning, and inverse dynamics, all fail to produce action-sensitive predictions, with GT–Zero cosine similarity gaps near zero throughout training. We propose a contrastive action loss that explicitly penalizes futures indistinguishable across shuffled action sequences, paired with a dedicated high learning rate for the action encoder. This variant is the only method that develops strong action sensitivity, achieving a GT–Zero gap of +0.299 at rollout horizon $k = 10$. We evaluate semantic fidelity using an ALOE multiple-choice VQA head on a task-match benchmark derived from LIBERO-100, where the model must identify the correct task description from predicted slot trajectories. The contrastive model outperforms all baselines at every prediction horizon, with the largest margin at the hardest settings where 50% of the trajectory must be predicted while also improving qualitative rollout predictions. These results suggest that explicit contrastive pressure on the training objective, not architectural access to action information, is the key factor for learning action-grounded object dynamics.

1. Introduction

Predicting how a visual scene evolves over time is a fundamental problem underlying robotics, autonomous driving, and video understanding. Recent world models have demonstrated strong planning and control capabilities by learning latent dynamics directly from visual observations, including DreamerV3 [3], DreamGen [4], and Direct Video Action models [8]. Yet current world models, despite handling background texture and global motion well, consistently fail in the cases that matter most: when objects collide, interact, or occlude one another. This limitation is not incidental;it reflects a deeper architectural mismatch. Patch-level and pixel-space models treat all regions of the image equally, with no explicit representation of the discrete, manipulable entities that govern the semantics of a scene.

Object-centric representations offer a principled remedy. Inspired by the cognitive observation that humans interpret the world in terms of objects rather than raw pixels, object-centric models decompose a scene into a small set of latent vectors, each corresponding to a distinct entity. Slot Attention [6] implements this idea by routing image features into a fixed number of object slots through a competitive attention mechanism, enabling downstream models to reason over individual entities rather than image patches.

In this paper, we investigate a surprising failure mode of object-centric world models in robotic manipulation: despite explicit access to robot actions, their predictions often become nearly action-invariant. Robotic manipulation provides an ideal setting for studying this phenomenon because future states are fundamentally determined by object interactions induced by actions. We use the LIBERO manipulation benchmark [5], which contains diverse tabletop manipulation tasks with RGB observations and action trajectories, allowing controlled evaluation of whether predicted object-centric representations truly encode action-dependent dynamics.

To study this question, we build LIBERO-JEPA, an object-centric world model based on the Joint-Embedding Predictive Architecture (JEPA) [1]. The model operates over object-level slot representations extracted by a frozen VideoSAUR encoder and predicts future slots using a transformer conditioned on robot actions. We find that standard training objectives, including Hungarian MSE, future ac-

tion conditioning, and inverse dynamics losses, all produce nearly action-invariant predictions. To address this failure mode, we introduce a contrastive action objective that explicitly forces futures generated under different action sequences to diverge. We evaluate both action sensitivity and semantic fidelity using a horizon-sweep benchmark based on a frozen ALOE task-classification head, showing that explicit action-grounding significantly improves the preservation of task-relevant information over long rollout horizons.

Our contributions are threefold:

- We identify a failure mode in object-centric JEPA world models whereby predictions remain nearly action-invariant despite explicit access to robot actions.
- We introduce a contrastive action objective that explicitly encourages futures generated under different action sequences to diverge.
- We show that contrastive action training substantially improves both action sensitivity and semantic task fidelity compared to Hungarian MSE, future action conditioning, and inverse dynamics objectives.

2. Methods

2.1. Overview

Our system follows a similar methodology to C-JEPA's three-stage pipeline [7]. More architectural details can be found in the Supplementary Materials section 4. LIBERO-JEPA consists of a frozen VideoSAUR slot encoder and a transformer predictor conditioned on robot actions. The predictor receives object slots from a history window and predicts future slots under a Hungarian-matched latent prediction objective.

A temporal predictor takes a window of $T_h = 5$ history frames and must predict the slot representations for the next $T_p = 3$ frames, operating at a frameskip of 4 (roughly 8 Hz in wall-clock time at the LIBERO simulation rate). The input at each timestep is a set of N object slots augmented with one additional action slot that encodes the robot's end-effector command.

2.2. Training Objective

Because slots are unordered and the same physical object may occupy different slot indices across runs, a naive per-index MSE would penalize permutations of an otherwise correct prediction. We instead use Hungarian matching: given predicted future slots $\hat{Z} \in \mathbb{R}^{N \times D}$ and EMA target slots $Z^* \in \mathbb{R}^{N \times D}$, we solve the linear assignment problem to find the minimum-cost bijection σ^* and compute

Table 1. Action sensitivity at the latest checkpoint for each model at horizon $k = 10$. GT denotes cosine similarity under ground-truth actions, Δ denotes the GT–Zero gap, and Pers. denotes the persistence baseline.

Method	Step	GT	Δ	Pers.
OC-JEPA	45k	0.7695	-0.0026	0.9384
v6b (Hungarian MSE)	100k	0.7746	-0.0011	0.9402
v6d (FAC)	100k	0.7710	-0.0002	0.9393
v6e (Contrastive)	40k	0.7663	**+0.2989**	0.9397
v6f (InvDyn)	30k	0.7720	+0.0089	0.9483

$$\mathcal{L}\text{pred} = \frac{1}{N \cdot T_p} \sum t = 1^{T_p} \sum_{i=1}^{N} |\hat{z}t, i - z^*t, \sigma^*(i)|^2$$

An equal-weight MSE on the masked history slots is added following C-JEPA's training recipe.

2.3. Contrastive Action Loss (Best-Performing Variant)

Standard Hungarian MSE training produced nearly action-invariant predictions. To encourage action grounding, we introduce a contrastive action loss that penalizes futures generated from shuffled action sequences when they remain too similar. We additionally train the action encoder with a higher learning rate to strengthen gradient flow. Additional implementation details are provided in the supplementary material (Section 5.6).

3. Experiments

3.1. Experimental Setup

We evaluate all methods on LIBERO-100 [5], a benchmark consisting of 100 household manipulation tasks with expert demonstrations recorded from an agentview RGB camera. Demonstrations are sub-sampled using a frame skip of four, producing effective trajectories of approximately 15–40 frames. The final 10% of demonstrations from each task are reserved as a held-out test set, and all reported metrics are averaged over 100 randomly sampled test trajectories.

To evaluate action sensitivity, we perform an action ablation study using three action conditions: (1) ground-truth actions, (2) zero actions, and (3) random actions. We define the GT–Zero gap as $[\Delta = \text{CosSim} * GT - \text{CosSim} * Zero,]$ where larger positive values indicate stronger dependence on action information. Models that ignore actions entirely produce $\Delta \approx 0$.

3.2. Main Quantitative Results

Table 1 compares all methods at their latest available checkpoint across prediction horizons $k \in 1, 3, 5, 10$.

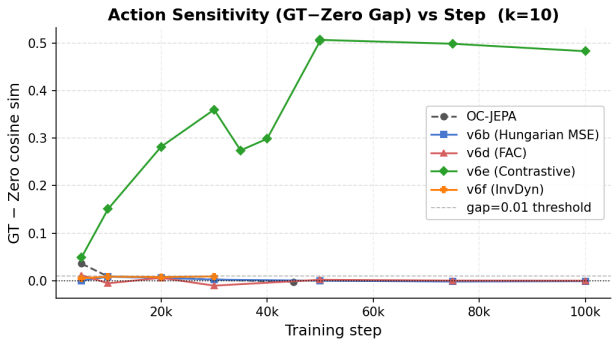

Figure 1. Action sensitivity measured by the GT–Zero gap at horizon $k = 10$ throughout training. OC-JEPA, Hungarian MSE, Future Action Conditioning (FAC), and Inverse Dynamics remain near zero, indicating little dependence on action information. In contrast, the Contrastive Action model develops a large positive GT–Zero gap, demonstrating that its predictions depend strongly on the provided actions.

All methods achieve similar slot prediction quality, with cosine similarity values clustering around 0.77. However, action sensitivity varies dramatically. OC-JEPA, v6b, v6d, and v6f all exhibit GT–Zero gaps near zero across all horizons, indicating that their predictions are largely unaffected by the provided action sequence. In contrast, v6e produces substantially larger positive gaps, reaching +0.299 at horizon $k = 10$, suggesting that it is the only method that consistently incorporates action information into future predictions.

A core result of this work is shown in Figure 1. While all baseline methods remain within approximately ± 0.01 of zero throughout training, v6e diverges monotonically from the earliest checkpoint and continues improving. This behavior suggests that the contrastive objective successfully creates gradient pressure that forces the predictor to distinguish between futures generated by different actions.

By comparison, v6d (Future Action Conditioning) explicitly provides future action embeddings to the Transformer, yet still exhibits near-zero action sensitivity. This result indicates that merely exposing the model to action information is insufficient; the training objective must actively require actions to influence predictions.

To measure how well each model preserves task-relevant semantic content as a function of how much of the trajectory must be predicted, we conduct a horizon sweep. For each model variant and each observation fraction $f \in 0.9, 0.8, 0.7, 0.6, 0.5$, we use the first f fraction of each trajectory as ground-truth context and roll the predictor forward to fill in the remaining $1-f$ fraction. The resulting slot trajectories, partially observed, partially predicted, are then fed to the frozen ALOE head, and task-match accuracy is recorded. Lower observation fractions correspond to harder settings where more of the trajectory must be hallucinated from scratch. The x-axis in Figure 1 reports the percentage of the trajectory that was predicted rather than observed, so curves that remain high toward the right of the plot indicate models that degrade gracefully under long-horizon prediction.

3.3. Baseline Comparison

We compare four model variants against a ground-truth slot upper bound (99.4% accuracy). The **H-MSE Loss** baseline trains the predictor with Hungarian MSE alone and no auxiliary losses, and degrades most severely: at 50% of the trajectory predicted, accuracy collapses to 19.8%, near the 25% chance level. This confirms that the base training recipe produces a predictor that generates geometrically plausible slots but loses task-discriminative information rapidly over time. The **OC-JEPA (No Masking)** ablation, which is identical to the main model but trained without slot masking, recovers meaningfully over H-MSE (34.0% at 50% predicted), demonstrating that the object-centric temporal architecture itself contributes beyond the loss function. The **Inverse Dynamics Loss** variant performs comparably to OC-JEPA across all horizons (32.8% at 50% predicted), suggesting that recovering the action from the predicted transition delta provides a similar inductive bias to the unmasked architecture but does not substantially exceed it.

3.4. Qualitative Analysis and Failure Modes

To better understand model behavior, we visualize slot prediction rollouts on held-out trajectories.

Across all methods, large static structures such as tables and background geometry are predicted reliably. However, small manipulable objects such as mugs, books, and tools remain difficult to track. Predicted slots often collapse into nearby background regions within a few rollout steps. This failure mode persists even in the contrastive object loss variant, suggesting that improved action sensitivity alone does not guarantee accurate object-centric prediction.

3.5. Contrastive Action Loss

Our best-performing model, trained with the **Contrastive Action Loss**, achieves 39.0accuracy at the hardest setting (50% predicted) and 96.6% at the easiest (10% predicted), outperforming all baselines at every horizon point. The margin over H-MSE Loss is largest at hard horizons, a 19-point gap at 50% predicted versus a 5-point gap at 10% predicted, indicating that the benefit of action conditioning grows as the prediction task becomes more demanding. This is the expected behavior of a model that has genuinely internalized action causality: at short horizons, all models can coast on inertia since the scene changes little, but at long horizons only a model that understands how actions shape the future can produce semantically coherent rollouts.

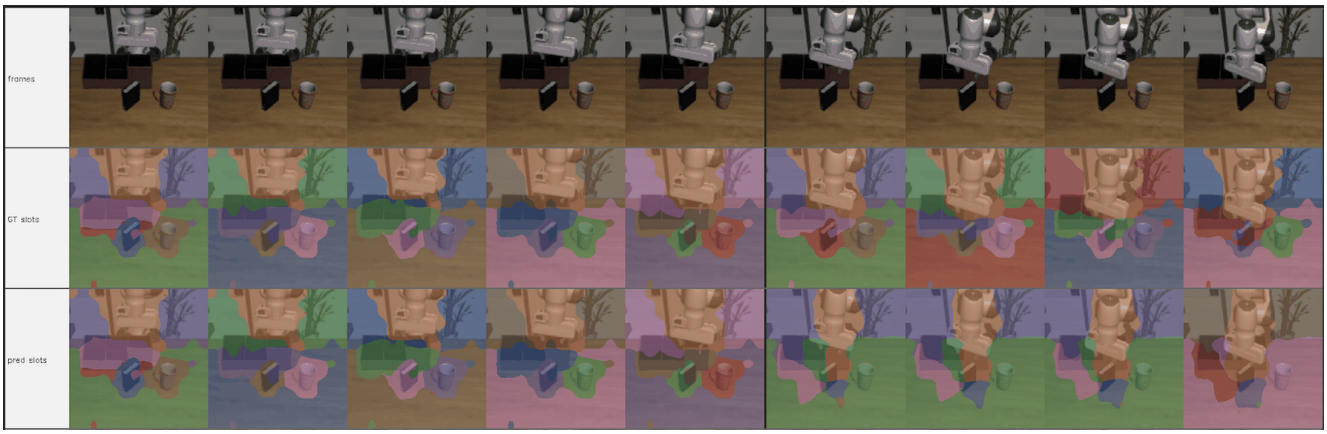

Figure 2. Qualitative slot prediction rollout on a held-out LIBERO trajectory. Rows show RGB observations, ground-truth VideoSAUR slots, predicted slots, and the persistence baseline. The dark line represents the boundary between previous input frames and predicted future frames.Large scene structures remain coherent across the rollout, while small manipulable objects such as the mug gradually disappear or merge into surrounding slots. This failure mode is shared across models and suggests that action sensitivity alone is insufficient for robust object-centric prediction.

The contrastive loss directly targets the failure mode identified in the H-MSE baseline: the action encoder receiving negligible gradient signal and producing nearly action-invariant predictions. By penalizing futures that are indistinguishable across shuffled action sequences, the loss forces the predictor to encode action-specific causal structure, which in turn keeps the predicted slot trajectories aligned with the task being performed.

The horizon sweep 3 reveals two distinct failure modes that the ablations help disentangle. First, training without masking (OC-JEPA) or with an insufficient action signal (H-MSE) both produce predictors that are brittle at long horizons, but for different reasons: OC-JEPA lacks the counterfactual pressure that masking provides, while H-MSE lacks action grounding. Second, the contrastive loss addresses the action-grounding failure specifically and yields the largest gains at the hardest prediction settings, where causal structure matters most. Together, these results support the core hypothesis of this work: that object-level masking combined with explicit action contrastive training produces world model representations that are not only geometrically accurate but semantically meaningful over extended rollout horizons.

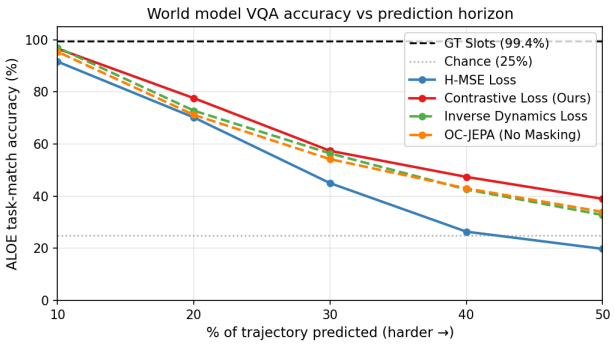

Figure 3. ALOE task-match accuracy as a function of prediction horizon. The x-axis reports the percentage of the trajectory that is predicted rather than observed, with larger values corresponding to more difficult long-horizon rollouts. Ground-truth VideoSAUR slots achieve 99.4% accuracy, while random guessing yields 25%. The proposed Contrastive Action Loss consistently outperforms all baselines and maintains the highest accuracy at long horizons, achieving 39.0% accuracy when 50% of the trajectory must be predicted. The widening performance gap relative to H-MSE at harder horizons suggests that explicit action grounding becomes increasingly important for preserving task-relevant semantic information during extended rollouts.

4. Conclusion

We presented LIBERO-JEPA, an object-centric world model for robotic manipulation and identified a key failure mode of existing training objectives: despite explicit action conditioning, Hungarian MSE, future action conditioning, and inverse dynamics losses all produce nearly action-invariant predictions. To address this issue, we introduced a contrastive action loss that explicitly encourages futures generated under different actions to diverge. This objective was the only method to achieve substantial action sensitivity, reaching a GT–Zero gap of +0.299 at horizon $k = 10$, and consistently achieved the strongest performance on the ALOE horizon-sweep benchmark. These results suggest that learning action-grounded dynamics requires explicit objective-level pressure rather than merely providing action information to the model. Future work should address the persistent challenge of tracking small manipulable objects during long-horizon rollouts.

References

[1] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023. 1

[2] M. Assran et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. 5

[3] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 1

[4] J. Jang, S. Ye, Z. Lin, J. Xiang, J. Bjorck, Y. Fang, F. Hu, S. Huang, K. Kundalia, Y.-C. Lin, L. Magne, A. Mandlekar, A. Narayan, Y. L. Tan, G. Wang, J. Wang, Q. Wang, Y. Xu, X. Zeng, K. Zheng, R. Zheng, M.-Y. Liu, L. Zettlemoyer, D. Fox, J. Kautz, S. Reed, Y. Zhu, and L. Fan. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025. 1

[5] B. Liu et al. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. 1, 2, 7

[6] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538, 2020. 1, 5

[7] H. Nam, Q. L. Lidec, L. Maes, Y. LeCun, and R. Balestriero. Causal-jepa: Learning world models through object-level latent interventions. *arXiv preprint arXiv:2602.11389*, 2026. 2, 5

[8] Rhoda AI. Causal video models are data-efficient robot policy learners: The direct video-action model. Company research note, 2026. 1

[9] A. Zadaianchuk, M. Kleindessner, F. Locatello, T. Brox, and G. Abbati. Object-centric learning for real-world videos by predicting temporal feature similarities. In *Advances in Neural Information Processing Systems*, volume 36, 2023. 5

5. Supplementary Materials

5.1. Object Tokenization with VideoSAUR

We use VideoSAUR [9] as our frozen slot encoder. VideoSAUR extends Slot Attention [6] to video by training on a self-supervised objective that encourages temporally consistent slot assignments: the encoder is trained to predict feature similarities between adjacent frames using a frozen DINOv2-small backbone as the patch-feature source, so slots learn to track objects across time without explicit correspondence supervision.

Each input frame is resized to 196×196 and encoded into N = 7 slots of dimension D = 128. At each timestep the encoder runs two iterations of slot attention (three on the first frame), producing a set of unordered object-centric vectors. Because slots are unordered, downstream losses must be permutation-invariant; we return to this in the training objective. The encoder parameters are locked after pretraining on LIBERO-100; only the slot sequences it produces are consumed by the predictor.

5.2. Transformer architecture

The predictor is a non-causal transformer operating over the flattened spatiotemporal token sequence of shape $(T_h + T_p)(N + 1)$. We use depth 6, 16 attention heads, head dimension 64, and FFN hidden dimension 2048, with 0.1 dropout. Learned time positional embeddings are added to each token before the transformer. This architecture is adapted directly from C-JEPA's Masked Slot AP Predictor [7], modified to exclude only the single action slot (rather than two slots) from masking, since LIBERO uses a unified action representation without a separate proprioception node.

5.3. Object-slot masking

During each training step, k = 3 of the 6 object slots are randomly selected and replaced with a learnable mask token at all future timestep positions. The predictor must infer the masked future slots from the remaining visible slots, the action slot, and the full history. This masking strategy, inherited from Causal-JEPA [7], is the mechanism by which the model is forced to reason about object-level causal dependencies: predicting a masked object's future requires understanding how the surrounding objects and the applied action constrain its trajectory.

5.4. EMA target encoder

Prediction targets are the slot representations produced by an exponential moving average (EMA) copy of the slot encoder, with decay 0.996. The EMA target provides a slowly evolving, stable prediction target without collapse, following V-JEPA 2 [2].

5.5. Action encoding

The raw action at each timestep is a 7-dimensional vector $[\Delta x, \Delta y, \Delta z, \Delta r, \Delta p, \Delta y, \text{gripper}]$. At frameskip 4, four consecutive raw actions are concatenated into a 28-dimensional vector representing the motion between two observed frames. A two-layer MLP with LayerNorm and GELU activation projects this to D = 128 and appends it as an extra slot, so the full per-frame token set has shape (N+1, D) = (8, 128). The action slot is never masked during training, ensuring the predictor always has access to action context.

Proprioceptive state (end-effector position, orientation, and gripper state; 8 dimensions) is encoded by a separate MLP and appended similarly when enabled.

5.6. Contrastive Action Loss Details

The contrastive object loss variant addresses this with two coupled changes:

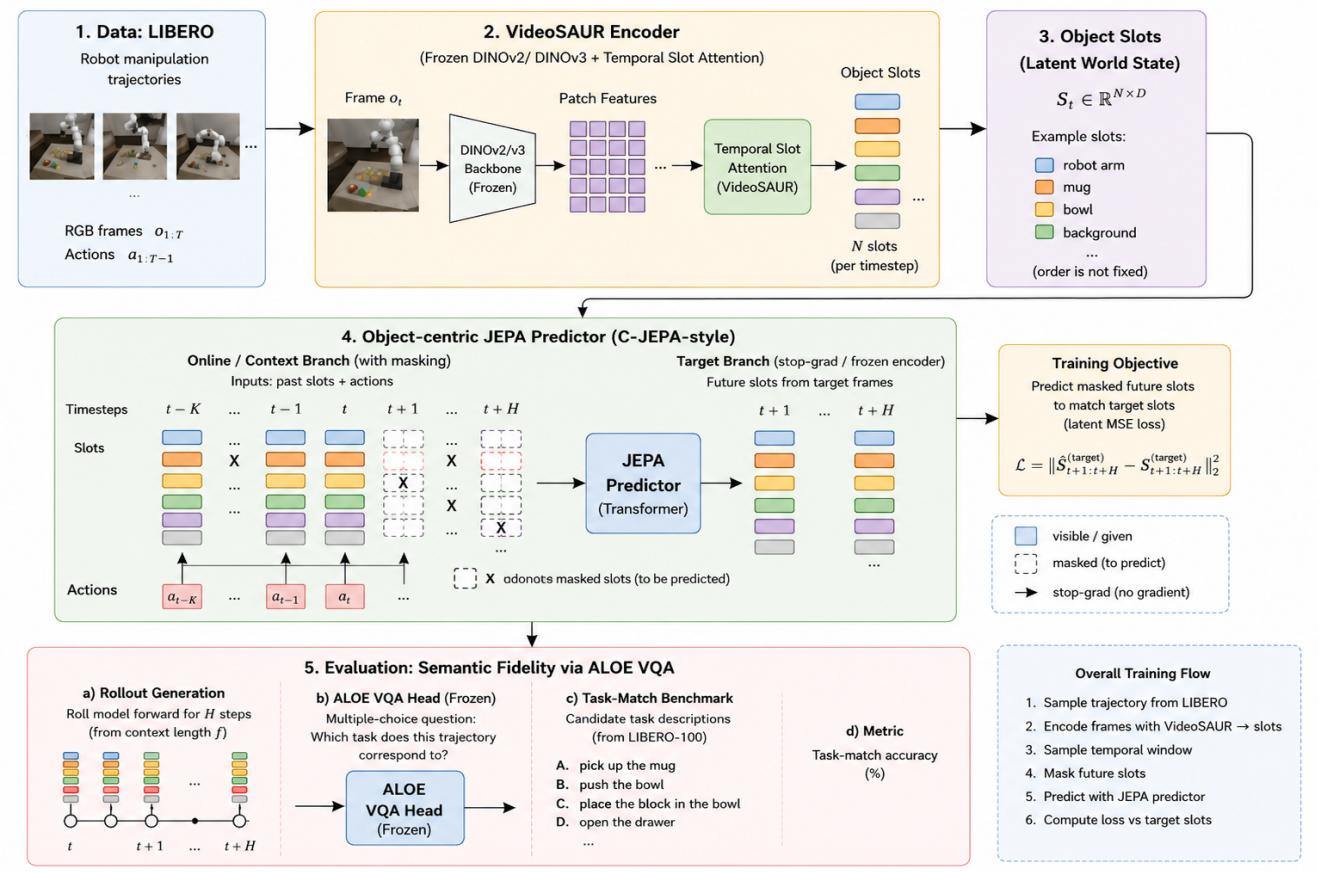

Figure 4. Overview of LIBERO-JEPA.

1. Contrastive action loss. For each batch, we construct negative pairs by shuffling the action sequences across batch elements. The predictor produces two sets of future slots: one conditioned on the real action sequence a, and one conditioned on a shuffled sequence a' drawn from a different trajectory in the same batch. We then apply a hinge loss requiring the mean-slot representations to diverge:

$$\mathcal{L}_{\text{contrast}} = \max\left(0,\ m - |\bar{z}(a) - \bar{z}(a')|^2\right)$$

where $\bar{z}$ denotes the mean over object slots at the first predicted timestep and $m = 0.5$ is the margin. The total loss becomes $\mathcal{L} = \mathcal{L}\text{pred} + \lambda\text{contrast} \cdot \mathcal{L}\text{contrast}$ with $\lambda\text{contrast} = 0.1$.

2. Separate action encoder learning rate. We assign the action encoder its own optimizer parameter group at 5× the base learning rate (base lr $= 210^{-4}$, action encoder lr $= 110^{-3}$). Without this multiplier, the contrastive gradient arriving at the action encoder is too small relative to the predictor's gradient to produce meaningful weight updates even when the contrastive loss is nonzero.

Together, these changes give the action encoder a strong direct training signal and the capacity to respond to it, forcing the model to produce qualitatively different predicted futures for qualitatively different actions. This is the key innovation of our approach relative to the base C-JEPA recipe and the property most directly required for use as a planning component.

5.7. Alternative Loss Variants

We explored two alternative auxiliary losses as ablations.

Inverse dynamics loss. Rather than pushing apart predictions from different actions, the inverse model recovers the action from the predicted transition. Given the mean object slot at time t, $\bar{z}t$, and the predicted mean at t+1, $\bar{z}t + 1^{\text{pred}}$, a lightweight two-layer MLP g predicts the action:

$$\hat{a} = g\left(\bar{z}t,\ \bar{z}t + 1^{\text{pred}} - \bar{z}t\right), \quad \mathcal{L}\text{inv} = |\hat{a} - a|^2$$

Gradients flow through the predictor and action encoder (the history slots from the frozen encoder are detached). This provides a complementary inductive bias: the predicted transition delta must encode enough information to decode the causing action. We use $\lambda_{\text{inv}} = 0.05$.

OC-JEPA baseline. To isolate the contribution of slot masking, we train an identical model with zero masked

slots. This variant, which we call OC-JEPA, sees all object slots in all future frames during training and therefore receives no pressure to infer masked objects from context. Comparing OC-JEPA to V6e measures the combined effect of masking and action contrastive training.

6. Dataset

We train and evaluate on LIBERO-100 [5], a publicly released benchmark for robot manipulation consisting of 100 tabletop manipulation tasks spanning diverse object configurations, spatial arrangements, and goal conditions. Each task is accompanied by approximately 50 expert demonstrations collected via teleoperation in a MuJoCo simulation environment, yielding roughly 5,000 total trajectories. Each demonstration is stored as an HDF5 file containing synchronized RGB video from two cameras (third-person agentview and wrist-mounted hand camera), 7-dimensional delta end-effector actions, proprioceptive state (end-effector position, Euler-angle orientation, and gripper state; 8 dimensions total), object states, and per-step rewards and dones. Demonstrations range from approximately 100 to 300 timesteps at the simulation rate. We use the agentview camera exclusively, as it provides a stable third-person view with consistent object visibility across tasks and matches the distribution on which the VideoSAUR slot encoder was pretrained.

6.1. Temporal Subsampling

Raw demonstrations are subsampled at a frameskip of 4, reducing the effective frame rate while expanding the action horizon per step. At each transition between subsampled frames, the 4 intervening raw actions are concatenated into a 28-dimensional vector, following the C-JEPA convention of stacking multi-step actions into a single effective action token. This gives the predictor a richer action signal per step and reduces the total sequence length the transformer must process.

6.2. Sliding Window Indexing

Rather than treating full trajectories as training samples, we slice each subsampled demonstration into overlapping clips of length $T_h + T_p = 5 + 3 = 8$ frames using a sliding window with stride 1. This maximizes the number of distinct training samples and exposes the model to clips beginning at every point in a trajectory, including mid-task states that are harder to predict than early frames.

6.3. Image Preprocessing

Frames are resized from their native resolution to 196×196 using bicubic interpolation and normalized with ImageNet mean and standard deviation, as required by the DINOv2-small backbone inside VideoSAUR.

6.4. Action and Proprioception Normalization

Dataset-wide mean and standard deviation are computed over all actions and proprioceptive states by scanning every demonstration before training. Actions and proprio are then standardized to zero mean and unit variance per dimension, which stabilizes the action encoder's input distribution and avoids large-magnitude dimensions (e.g., the binary gripper signal) dominating the learned projection.

6.5. Slot Normalization

VideoSAUR slot vectors have raw L2 norms of approximately 6. Before being passed to the predictor, all slots are L2-normalized to unit vectors. This concentrates training on predicting the direction of future slots, which is exactly what the cosine similarity evaluation metric measures, rather than requiring the predictor to also reproduce the correct magnitude. Raw per-slot norms are retained as a proxy for object spatial extent and used to construct inverse-norm loss weights in the ablation, where small objects such as mugs and tools receive amplified gradient contribution relative to large background slots.