LEARNING LATENT 3D ROTATIONS FROM UNLABELED VISUAL SEQUENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning how visual states transform under physical transformations is important for visual reasoning, but transformation-aware models typically assume access to transformation labels. We ask whether $\mathrm{SO}(3)$-parameterized actions, learned from unlabeled visual sequences, can recover meaningful structure of the underlying physical 3D rotations. Given object-centric sequences of rotating objects, we jointly train an action inference network and an action-conditioned transition model in the latent space of a frozen visual encoder. To discourage degenerate predictive codes, we introduce a reverse-prediction constraint that requires inferred actions to support consistent inverse transitions. Without using labeled data, the learned actions recover meaningful structure of the underlying 3D rotations. As a complementary evaluation, we show that the learned dynamics support object matching across 3D viewpoint changes through latent-action planning, outperforming static feature-similarity baselines on abstract shapes with limited semantic shortcuts.

1 INTRODUCTION

Understanding how visual states change under physical transformations is important for reasoning and planning. Prior work has studied transformation-aware visual learning from two complementary perspectives. Action-conditioned dynamics models learn to predict how a state changes given an observed action or transformation (Ha & Schmidhuber, 2018; Hafner et al., 2019; Hansen et al., 2022; Zhou et al., 2025), while equivariant representation learning seeks visual representations whose changes systematically reflect transformations of the input (Qi et al., 2019; Garrido et al., 2023; Konstantinou et al., 2025). These approaches have traditionally assumed access to the transformation labels relating observations during training.

An important question is whether transformation-aware learning can instead occur without access to transformation labels. Answering this question could enable self-supervised algorithms that eliminate the need for explicit transformation supervision and rely solely on visual observations. STL (Yu et al., 2024) has explored removing explicit transformation labels from training for equivariant representation learning, but does so for image-level augmentation such as cropping or color jitter. Related work on latent action learning has shown that action-like variables can be inferred from observation-only sequences. For example, Genie (Bruce et al., 2024) learns a small set of discrete latent actions for interactive video generation, while LAPO (Schmidt & Jiang, 2024) recovers latent actions that largely align with discrete control actions in Procgen environments, and DiLA (Zhang et al., 2026) studies continuous latent action manifolds associated with transformations such as translation, scaling, and in-plane rotation.

In this work, we consider the setting of rigid 3D rotations. Given unlabeled sequences of rotating objects, we ask whether inferred latent actions recover meaningful geometric structure of the underlying physical rotations. We study this question using synthetic object-centric sequences, an experimental setting also adopted in prior work (Garrido et al., 2023; Konstantinou et al., 2025). This controlled setting enables a systematic analysis of how learned actions relate to the underlying transformation geometry.

To investigate this, we jointly train a *Transition Network* and an *Action Inference Network* in the latent space of a frozen visual encoder. The action network infers the transformation relating two

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.

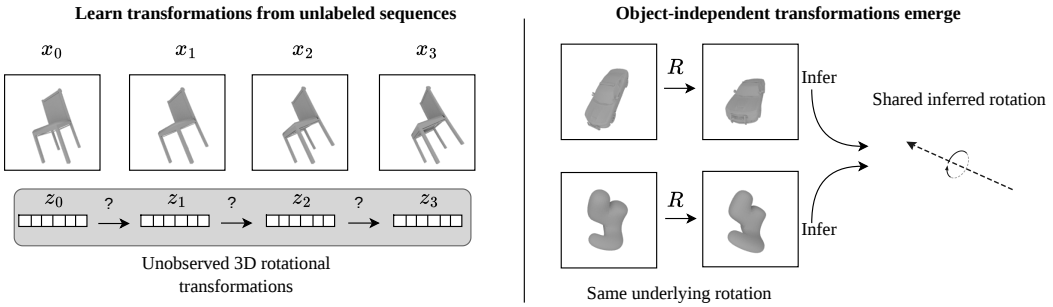

Figure 1: We learn latent 3D rotations from unlabeled sequences of rotating objects. The inferred transformations capture shared, object-independent rotational structure.

consecutive latent states, with the inferred transformation parameterized in $\mathrm{SO}(3)$, while the transition network predicts the next latent state conditioned on the current state and the inferred action. In this way, the model jointly learns a continuous latent action space and its associated dynamics using only sequential visual observations.

A central challenge in jointly learning actions and their corresponding transitions is avoiding degenerate solutions. The two networks may co-adapt to encode rotations into arbitrary predictive codes that need not correspond to meaningful physical transformations. To address this, we introduce a reverse-prediction constraint that enforces consistency under inversion. Specifically, if an inferred action maps the current state to the next state, then its inverse should map the next state back to the current one. We find that this constraint is critical for aligning the learned actions with physically meaningful 3D rotations.

The resulting latent actions recover substantial structure of the underlying physical rotations: inferred axes are consistent across objects (Figure 1) and align with ground-truth axes up to a global coordinate transformation, inferred magnitudes vary almost linearly with physical rotation magnitude, and inferred actions approximately respect inverse and composition operations. Finally, to complement our geometric analysis, we evaluate whether the learned dynamics can be used for 3D object shape matching, a classic task in cognitive science for probing 3D perception (Shepard & Metzler, 1971). On abstract shapes, where semantic shortcuts are limited, this approach outperforms static feature-similarity baselines.

2 Related Work

Equivariant learning. Equivariant representation learning seeks representations that transform predictably under changes to the input. Equivariance can be imposed through known architectural symmetries (Cohen & Welling, 2016; Esteves et al., 2019; Fuchs et al., 2020) or encouraged through self-supervised objectives and learned transformation operators (Qi et al., 2019; Dangovski et al., 2021; Lee et al., 2021; Park et al., 2022; Devillers & Lefort, 2023; Garrido et al., 2023; Konstantinou et al., 2025). These methods typically assume a specified symmetry, known applied transformations, or a predefined augmentation family. STL (Yu et al., 2024) removes explicit transformation labels but operates over image-space augmentations, while Flow Equivariance (Xiong et al., 2021) learns from video using a pretrained optical flow estimator. Our work is complementary to this line of research, focusing on latent transformation inference and dynamics learning in a fixed pretrained representation space.

Latent action models. World models predict how future states evolve under actions and can support planning over candidate action sequences (Hansen et al., 2022; 2024; Zhou et al., 2025), but typically rely on observed actions during training. For example, DINO-WM (Zhou et al., 2025) learns dynamics over pretrained visual features using ground-truth actions. A related line of work instead learns action-like variables from observation-only sequences (Edwards et al., 2019; Menapace et al., 2021; Bruce et al., 2024; Schmidt & Jiang, 2024; Ye et al., 2025; Gao et al., 2025).

Genie (Bruce et al., 2024) learns a small discrete latent action space for controllable video generation, while LAPO (Schmidt & Jiang, 2024) learns latent actions that largely align with discrete control actions in Procgen environments. DiLA (Zhang et al., 2026) analyzes continuous latent action manifolds associated with translation, scaling, and in-plane rotation, as well as relative yaw in navigation. Separately, Ko et al. (2024) estimate explicit 3D rigid transformations from generated videos using optical-flow correspondences and depth for robot control. We instead ask whether latent actions inferred from unlabeled visual sequences become geometrically aligned with the unobserved 3D rotations about arbitrary axes that generated the observed transitions.

3D shape perception. Multiple recent studies have investigated the 3D capabilities of vision models (El Banani et al., 2024; Danier et al., 2025; Chen et al., 2025). For 3D shape perception, recent work has reported that vision models can lag behind human performance (O'Connell et al., 2023; Bonnen et al., 2024; Ollikka et al., 2024). Cognitive science has long studied 3D shape judgment and mental rotation in humans, yet the mechanisms underlying these abilities remain an open question (Rock & DiVita, 1987; Tarr & Bülthoff, 1998; Ullman, 1998; Shepard & Metzler, 1971). Interestingly, when human decisions are time-restricted, their performance becomes comparable to that of vision models (Ollikka et al., 2024; Bonnen et al., 2023), suggesting that temporal processing may play an important role in 3D shape judgment.

3 METHOD

3.1 PROBLEM SETUP

We consider object-centric video sequences $x_{0:T} = (x_0, x_1, \ldots, x_T)$ where each sequence depicts a rigid object undergoing a rotation. During training, no ground-truth rotation information is provided. The learning signal comes solely from the temporal structure of the observation stream, in which consecutive frames typically differ by small rotations.

Our goal is to recover a latent action variable a_t for each transition (x_t, x_{t+1}) such that a_t corresponds to the underlying physical rotation between the two frames.

Each frame is mapped to a latent state $z_t = E(x_t)$ using a frozen pretrained encoder E. In our main experiments, E is a frozen encoder from a distilled Stable Diffusion VAE (Bohan, 2023; Rombach et al., 2022). The resulting latent representation z_t serves as the state space in which transformations are learned.

Latent Action Inference. We introduce an action inference network A_ϕ that predicts the transformation that explains the change between two consecutive latent states:

$$a_t = A_\phi(z_t, z_{t+1}) \tag{1}$$

The inferred action $a_t \in \mathbb{R}^6$ is parameterized using the continuous 6D rotation representation (Zhou et al., 2019). Specifically, the 6D output is interpreted as two 3D vectors and mapped to a valid rotation matrix in $\mathrm{SO}(3)$ via Gram–Schmidt orthogonalization. This mapping enforces that the representation corresponds to a valid rotation. Subsequently, the first two columns of the resulting rotation matrix are extracted to form a 6D vector that is then provided as input to the Transition Network. While this mapping enforces consistency with $\mathrm{SO}(3)$, it does not guarantee that the learned actions correspond to meaningful physical rotations. We implement A_ϕ as a convolutional encoder with a 6D prediction head. Full architectural details are provided in Appendix B.

Transition Network. The Transition Network T_θ models state transitions in latent space. Concretely, its role is to predict the subsequent state z_{t+1} conditioned on the current state z_t and the inferred action a_t:

$$\hat{z}_{t+1} = T_\theta(z_t, a_t) \tag{2}$$

The transition network is implemented as a FiLM-conditioned U-Net (Ronneberger et al., 2015) with residual blocks. The inferred action is injected through FiLM (Perez et al., 2018) modulation of intermediate feature activations. Full architectural details are provided in Appendix B. Training uses two-step autoregressive prediction rather than pure teacher forcing.

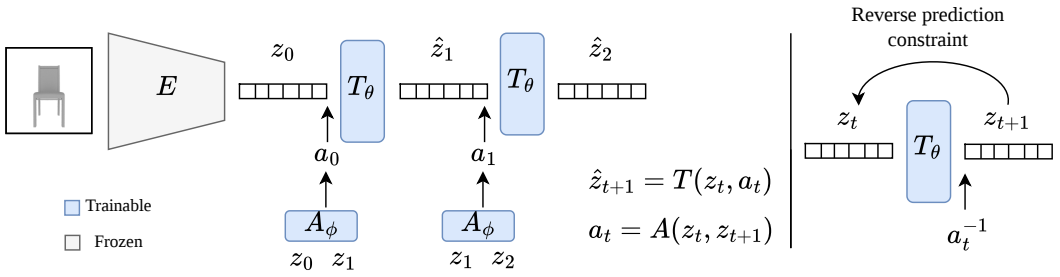

Figure 2: Training overview. A transition network T_θ and action inference network A_ϕ are jointly trained on unlabeled rotating-object sequences in a frozen latent space. A_ϕ predicts latent transformations between consecutive frames, while T_θ predicts the next state conditioned on the current state and inferred action. A reverse prediction constraint enforces consistency under inversion.

3.2 Self-Supervised Training

An overview of our training procedure is provided in Figure 2. During training, we sample sequences of objects rotating about a fixed axis, with transition magnitudes drawn uniformly from 5° to 20°. The primary learning signal is next-state prediction in latent space:

$$\mathcal{L}_{\text{fwd}} = \sum_{t=0}^{T-1} \|T_\theta(z_t, A_\phi(z_t, z_{t+1})) - z_{t+1}\|_2^2 . \tag{3}$$

This objective alone encourages actions that are useful for prediction. However, it does not guarantee that the inferred transformations align with real rotations.

Reverse Prediction Constraint. A significant challenge in jointly training A_ϕ and T_θ is the existence of degenerate solutions. Without further constraints, the networks can co-adapt to encode rotations into arbitrary internal codes that minimize $\mathcal{L}_{fwd}$ but do not correspond to the $SO(3)$ geometry of physical 3D rotations.

To prevent this, we introduce a reverse prediction constraint ($\mathcal{L}_{rev}$), which is critical for the alignment with meaningful physical transformations. We exploit the analytical invertibility of 3D rotations. If an action a_t transforms z_t to z_{t+1}, then the inverse action a_t^{-1} should transform z_{t+1} back to z_t.

To obtain the inverse a_t^{-1}, we first map the inferred 6D code a_t to a rotation matrix $\mathbf{R}(a_t) \in SO(3)$, compute the matrix inverse $\mathbf{R}^{-1}$, and then map back to the 6D representation which we treat as a_t^{-1}.

The reverse prediction loss is defined as:

$$\mathcal{L}_{rev} = \sum_{t=0}^{T-1} \|T_\theta(z_{t+1}, a_t^{-1}) - z_t\|_2^2 \tag{4}$$

The reverse constraint enforces that actions must support bidirectional consistent transitions. This requirement strongly biases the learned representation toward true geometric transformations. Empirically, we find this term to be crucial. Without reverse prediction, the discovered actions fail to correspond to physical rotations even when forward prediction error is low. With reverse prediction added, the learned transformations become semantically aligned with actual object rotations.

Rotation Magnitude Prior. We further regularize the magnitude of inferred rotations. Let θ_t denote the rotation angle associated with a_t. We penalize inferred rotations whose angle θ_t exceeds a maximum $\theta_{\max}$. This prior reflects the typical temporal continuity of the data, where consecutive frames exhibit moderate viewpoint changes. We implement it as an L2 penalty ($\mathcal{L}_\alpha$) for angles that exceed the maximum value. We set $\theta_{\max} = 30^\circ$, making the constraint a weak regularizer rather than tightly enforcing the observed transition distribution.

The complete training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{fwd}} + \lambda_{\text{rev}}\mathcal{L}_{\text{rev}} + \lambda_{\alpha}\mathcal{L}_{\alpha} \tag{5}$$

Full training details and hyperparameters are provided in Appendix C.

3.3 Datasets

We train on synthetic object-centric rotation sequences from ShapeNet Chang et al. (2015) and a procedurally generated Abstract dataset designed to limit semantic shortcuts and emphasize geometric reasoning. We render each sequence by sampling a single 3D rotation axis and rotating the object through a full 360° at 1° increments. Unless otherwise stated, the analyses in Section 4 use held-out objects from both datasets. Dataset sizes, splits, and rendering details are provided in Appendix C.1.

4 Geometric Analysis of the Inferred Actions

The central question of our work is whether latent actions inferred without rotation supervision become aligned with the physical transformations underlying the observed visual transitions. In this section, we provide direct empirical evidence addressing this question.

4.1 Geometry of the Learned Rotation Axes

We first analyze the directional structure of inferred actions by converting each predicted 6D action into a rotation matrix and extracting the corresponding axis of rotation. Figure 3 visualizes these axes on the unit sphere for three settings: an action-supervised baseline, self-supervised training without reverse prediction, and our full self-supervised model. Each color denotes a different sequence, with individual points corresponding to inferred rotation axes for transitions within that sequence.

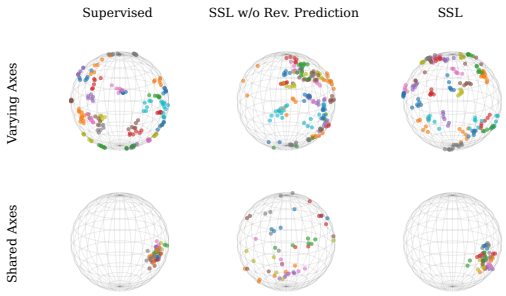

Figure 3: Rotation-axis visualization for supervised training, SSL without reverse prediction, and our full model. Colors denote sequences, and points denote inferred axes for transitions within each sequence. Top: sequences with different physical axes. Bottom: different sequences sharing the same physical axis.

The top row visualizes different object sequences, where each sequence rotates around its own physical axis. In this setting, a correct model should assign a consistent axis within each sequence, while allowing different sequences to occupy distinct regions of the sphere. In our SSL model, the inferred axes form tight sequence clusters that closely match the supervised baseline. This indicates that the learned actions capture a stable rotational structure. While the SSL model without the reverse prediction constraint shows some within-sequence clustering, the overall axis distribution is limited to a subregion of the unit sphere, lacking the diversity observed in the other models.

The bottom row considers multiple object sequences that share the same underlying axis of rotation. Here, an ideal representation should map these sequences to a common point on the sphere, reflecting independence from object identity. Again, the full model aligns with the supervised baseline, with inferred axes concentrating around a single shared direction. In contrast, the SSL model without the reverse prediction constraint exhibits a dispersed set of axes, indicating that the learned actions are not consistent with the underlying physical transformations. These results suggest that the reverse consistency constraint is essential for recovering an object-agnostic rotational action space.

4.2 Agreement with Ground-Truth Rotation Axes

To assess whether the inferred actions correspond to physical 3D rotations, we compare their predicted rotation axes to ground-truth axes. Since the model is trained in a fully self-supervised manner, its learned coordinate system need not align with the dataset reference frame. As a result, the inferred rotations are only identifiable up to a global rotation.

We account for this by estimating a global alignment rotation $\mathbf{Q} \in SO(3)$ between the learned axes and the ground-truth axes using orthogonal Procrustes alignment. The alignment is computed on the training set and evaluated on held-out test data. Let $\hat{\mathbf{u}}_i \in \mathbb{R}^3$ denote the inferred unit rotation axis for sample i, and let $\mathbf{u}_i \in \mathbb{R}^3$ denote the corresponding ground-truth axis. We solve

$$\mathbf{Q}^\star = \arg\min_{\mathbf{Q}\in SO(3)} \sum_i \|\mathbf{Q}\hat{\mathbf{u}}_i - \mathbf{u}_i\|_2^2 . \qquad (6)$$

After applying this single global transformation, the mean axis error drops substantially, from $113.9°$ to $12.7°$, as shown in Figure 4. This indicates that the inferred actions are not arbitrary SO(3)-parameterized latent codes. After a single global change of coordinate frame, their rotation axes closely align with the underlying physical rotation axes. The aligned self-supervised model approaches the supervised model, which obtains $11.1°$ error, despite never observing ground-truth rotations during training.

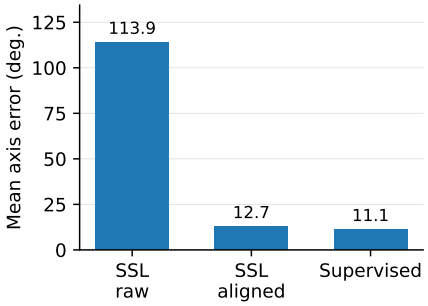

Figure 4: Mean rotation-axis error to ground truth. After a global frame rotation, the SSL-learned axes approach the supervised model.

4.3 Scaling of Rotation Magnitude

A desirable property of a learned action representation is that larger object rotations should correspond to larger inferred rotation magnitudes. To evaluate whether the learned action space captures rotational magnitude, we use sequences with fixed rotation sizes ($5°$, $10°$, $15°$, and $20°$) and measure the mean magnitude of the inferred action for each group. Figure 5 shows an approximately linear relationship between true and inferred rotation magnitudes. Larger physical rotations produce correspondingly larger inferred magnitudes, with the predictions differing from the ground-truth magnitudes primarily by an approximately constant offset.

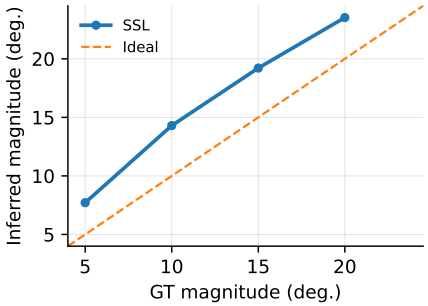

Figure 5: Mean inferred action magnitude versus ground-truth magnitude. Predicted magnitudes increase linearly with the true angular step size.

This result demonstrates that the learned action space captures more than rotational direction alone. It also organizes transformations according to their scale, assigning larger magnitudes to larger physical rotations. Importantly, this structure emerges without access to ground-truth rotation magnitudes, indicating that the model has internalized a meaningful notion of rotational distance.

4.4 Inverse and Composition Consistency

We test whether independently inferred actions respect rotation-group operations on held-out data. For this evaluation, rotations are sampled from the same $5°$–$20°$ range used during training. Let $\hat{\mathbf{R}}_{ij}$ denote the rotation matrix inferred between states z_i and z_j, where $d_{\mathrm{SO}(3)}$ denotes geodesic distance on $\mathrm{SO}(3)$. We measure

$$e_{\mathrm{inv}} = d_{\mathrm{SO}(3)}(\hat{\mathbf{R}}_{10}, \hat{\mathbf{R}}_{01}^{-1}), \qquad e_{\mathrm{comp}} = d_{\mathrm{SO}(3)}(\hat{\mathbf{R}}_{02}, \hat{\mathbf{R}}_{12}\hat{\mathbf{R}}_{01}). \qquad (7)$$

For composition, held-out triplets use successive rotations about different physical axes, so the test is not restricted to co-axial transformations. Table 1 reports the mean errors. The self-supervised

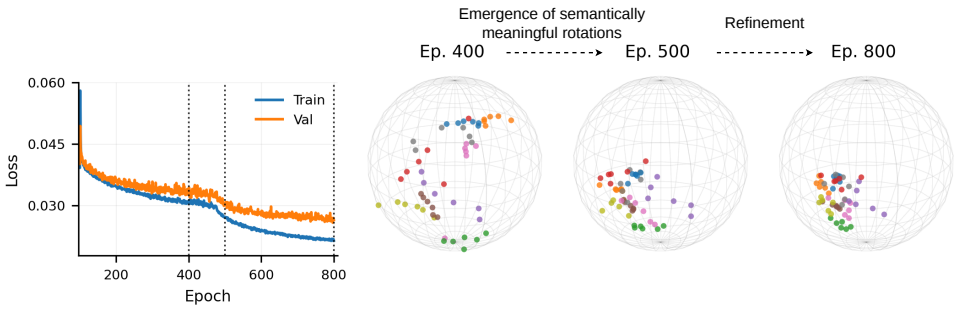

Figure 6: Transition behavior. We observe a sharp drop in next-state prediction loss that coincides with the inferred actions becoming semantically meaningful rotations. The sphere visualizations show inferred rotation axes for different object sequences rotating around the same underlying physical axis. After the drop, the inferred axes begin to form a cluster, indicating the alignment of the inferred actions with true physical rotations.

model achieves 2.5° inverse error and 5.0° composition error, compared with 2.2° and 3.7° for the action-supervised model. Thus, independently inferred actions approximately respect inverse and composition structure, with errors close to those obtained under direct action supervision.

Table 1: Consistency of inferred rotations under inverse and composition operations.

Model	Inverse error (°)	Composition error (°)
Action-supervised	2.2	3.7
Self-supervised	2.5	5.0

4.5 Transition Behavior During Training

We analyze how rotational structure emerges over the course of training. Figure 6 plots the next-state prediction loss together with qualitative visualizations. We visualize the axes of the corresponding inferred actions for different object sequences undergoing rotation around the same physical axis.

Notably, we observe a sharp transition during training. After an initial period of gradual improvement, the prediction error undergoes a sharp drop, which coincides with the inferred actions becoming semantically meaningful rotations. To better understand this transition, we visualize the inferred rotation axes at different stages of training (before the drop, immediately after, and at convergence). Before the transition, the inferred axes are scattered, indicating that the model has not yet learned a consistent representation of the underlying rotation. Immediately after the drop in prediction error, the inferred axes form a cluster, revealing the emergence of a shared rotational structure. Subsequent training primarily refines this representation, further tightening the cluster without qualitatively changing its structure.

5 Visual Planning with Learned Dynamics

As a complementary evaluation, we test whether the learned action space and dynamics can support transformation-based visual reasoning beyond static feature similarity. We use image-based object matching across 3D viewpoint changes as a proof-of-concept task.

Evaluation protocol. We construct a total of 700 trials using test objects from both the ShapeNet and Abstract datasets. For each trial, we sample a source image and construct a matching target by selecting a view of the same object rotated by 60° to 100° around a randomly selected axis. To construct the distractor (lure), we randomly sample a different object from the test set. For ShapeNet, the lure is additionally constrained to belong to the same object category as the source object.

Baselines. We compare our method against several variants of DINOv2 (Oquab et al., 2023) and DINOv3 (Siméoni et al., 2025) as static feature-similarity baselines, as well as a representation-

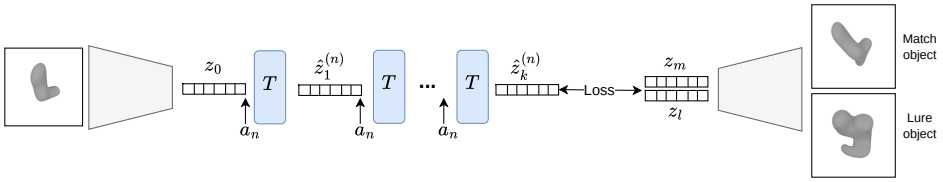

Figure 7: Visual planning. Our model optimizes a sequence of latent actions that progressively transforms the source image representation toward the target representation. The target object with the lowest planning loss is chosen as the match.

matched VAE baseline using the same frozen encoder as our model. These baselines test whether exploiting the learned transformation dynamics provides benefits beyond static feature similarity. We also report a supervised variant of our model trained with ground-truth actions to contextualize the self-supervised setting. For this supervised variant, we additionally evaluate a DINOv2 feature space to assess the effect of the underlying visual representation.

Planning by action search. Given source and target images, we encode $z_s = E(x_s)$ and $z_{tgt} = E(x_{tgt})$. Since the composition of 3D rotations is itself a 3D rotation, any net transformation between the two views can be represented by a single rotation axis and angle. We therefore search over candidate axes and realize different rotation magnitudes by repeatedly applying a fixed-magnitude action. Specifically, we sample candidate actions $\{a_n\}_{n=1}^{N}$, each corresponding to a different axis but the same angular step, and roll out

$$\hat{z}_k^{(n)} = T_\theta(\hat{z}_{k-1}^{(n)}, a_n), \qquad \hat{z}_0^{(n)} = z_s, \qquad k = 1, \ldots, K_{\max}. \tag{8}$$

For each candidate axis, we evaluate the matching loss across the rollout and retain its minimum:

$$\ell_n = \min_{1 \le k \le K_{\max}} \left\| \hat{z}_k^{(n)} - z_{\text{tgt}} \right\|_2^2. \tag{9}$$

The target candidate with the lowest planning loss is selected as the match. A visualization of the procedure is provided in Figure 7. We use up to $K_{\max} = 10$ steps, with 10° actions for supervised models and 15° inferred-action magnitudes for the self-supervised model, corresponding approximately to 10° physical rotations (Figure 5). Candidate actions are refined using CEM Rubinstein (1999); Zhou et al. (2025). Details of the optimization algorithm are provided in Appendix A.

Bidirectional planning. Inference-time optimization in our setting is an open-loop procedure. Actions are optimized entirely in latent space by repeatedly applying the learned transition model, without access to intermediate ground-truth observations. As a result, multi-step rollouts can accumulate model error over time. To reduce this sensitivity, we evaluate a bidirectional variant of the planning objective. Instead of optimizing only forward rollouts from the source representation, we simultaneously propagate from both the source and target representations and optimize the two trajectories to meet at an intermediate latent state. Intuitively, this shortens the effective rollout horizon from either side and provides a stronger consistency constraint on the predicted transformation.

Results. Table 2 compares static feature-similarity baselines with our latent action planning approach. The representation-matched VAE baseline performs near chance on both datasets (53.7% on Abstract and 55.0% on ShapeNet), indicating that the VAE representation alone provides limited matching signal. On the Abstract dataset, where procedurally generated shapes provide few semantic or appearance-based shortcuts, our self-supervised planning model reaches 88.3% accuracy, substantially outperforming the strongest static DINO baseline at 73.3%. In this setting, explicitly reasoning over viewpoint geometry provides a clear advantage for matching.

On ShapeNet, however, strong pretrained visual representations provide a competitive alternative. DINOv3-h reaches 83.1%, compared with 76.8% for our self-supervised model and 79.4% for the action-supervised VAE model. We interpret this difference as reflecting the stronger object-level cues available in ShapeNet, which allow static feature similarity to solve many trials without explicitly modeling the intervening transformation. Thus, the planning advantage is most pronounced when matching depends primarily on geometric viewpoint reasoning rather than semantic similarity.

Table 2: Object matching accuracy across different models.

Model (params.)	Abstract	ShapeNet
Static feature similarity		
VAE (1.2M)	53.7	55.0
DINOv2-s (21M)	69.5	70.9
DINOv2-l (300M)	70.5	75.9
DINOv3-s (21M)	71.5	75.0
DINOv3-l (300M)	69.8	78.4
DINOv3-h (840M)	73.3	**83.1**
Latent Action Planning		
Ours (Sup. actions) (7M)	**92.0** $\pm$ 0.1	79.4 $\pm$ 0.9
Ours (SSL) (7M)	88.3 $\pm$ 0.7	76.8 $\pm$ 0.5

Table 3: Backbone comparison for action-supervised models.

Backbone	Abstract	ShapeNet
VAE	92.0 $\pm$ 0.1	79.4 $\pm$ 0.9
DINOv2	**95.0** $\pm$ 0.4	**95.6** $\pm$ 0.8

Table 4: Comparison of unidirectional $\rightarrow$ bidirectional planning performance.

Model	Abstract	ShapeNet
SSL	82.7 $\pm$ 0.3 $\rightarrow$ **88.3** $\pm$ 0.7	66.3 $\pm$ 0.4 $\rightarrow$ **76.8** $\pm$ 0.5
Sup. VAE	82.3 $\pm$ 0.3 $\rightarrow$ **92.0** $\pm$ 0.1	69.6 $\pm$ 0.8 $\rightarrow$ **79.4** $\pm$ 0.9
Sup. DINOv2	82.5 $\pm$ 0.6 $\rightarrow$ **95.0** $\pm$ 0.4	82.8 $\pm$ 0.1 $\rightarrow$ **95.6** $\pm$ 0.8

However, Table 3 shows that the action-supervised variant of our approach benefits substantially from replacing the VAE representation used in our main experiments with stronger pretrained features. Using our action-supervised variant trained on DINOv2 features substantially improves performance, reaching 95.6% on ShapeNet and 95.0% on Abstract. This suggests that high-quality pretrained representations can strongly benefit performance. However, self-supervised training in the DINOv2 feature space proved more challenging than in the VAE space, with the discovered actions failing to converge to semantically meaningful rotations. We hypothesize that this is due to DINOv2's stronger invariance to fine-grained viewpoint changes. Finally, Table 4 compares the performance of our bidirectional planning objective to unidirectional planning, clearly showing that bidirectional planning consistently outperforms unidirectional.

6 Discussion and Conclusion

We presented a self-supervised method for learning latent actions that recover 3D rotational structure from unlabeled image sequences. By jointly training an action inference network and an action-conditioned transition model in a frozen latent space, the model learns transformations that align with 3D rotational structure. We introduce a reverse-prediction constraint that we find critical for preventing degenerate predictive codes and encouraging the learned actions to align with physically meaningful rotations.

Our analyses show that the inferred actions recover several properties of 3D rotation geometry without access to rotation labels: learned axes are consistent across objects and align with ground-truth axes up to a global coordinate transformation, inferred magnitudes vary systematically with physical rotation magnitude, and inferred actions approximately respect inverse and composition operations. These results suggest that structured continuous action spaces reflecting 3D rotation geometry can emerge from temporal prediction when appropriate geometric constraints are imposed. As a complementary evaluation, we show that the learned dynamics can also support latent-space planning for viewpoint-aware object matching, with substantial gains over static feature-similarity baselines on abstract shapes where semantic shortcuts are limited.

Limitations Our approach is evaluated primarily on synthetic data. While this synthetic setting enables controlled study of 3D rotations, it does not capture the complexity of real-world visual data. Additionally, we restrict dynamics to rigid 3D rotations, and it remains unclear how well the framework extends to more complex transformations. Finally, our method relies on a pretrained encoder, and performance may depend on the quality and inductive biases of this representation. Addressing these limitations is an important direction for future work.

AI USE STATEMENT

We used generative AI tools for literature search, feedback on research methodology and experimental results, and substantial assistance with code implementation. We also used these tools to refine and edit author-written text. The research question and core method were developed by the authors. We did not use generative AI tools for other tasks requiring disclosure under the ICLR 2027 AI policy. We take full responsibility for the final work.

REPRODUCIBILITY STATEMENT

We provide the model architecture, training procedure, dataset construction, and evaluation protocols in the main text and appendices. Appendix A presents the latent-action planning algorithm and CEM optimization procedure, Appendix B provides the network architectures, and Appendix C gives training hyperparameters and dataset-generation details. We will release the implementation and data-generation pipeline to facilitate reproduction of the reported results.

REFERENCES

Ollin Boer Bohan. Tiny autoencoder for stable diffusion (taesd). `https://github.com/madebyollin/taesd`, 2023.

Tyler Bonnen, Anthony D Wagner, and DL Yamins. Medial temporal cortex supports compositional visual inferences. 2023.

Tyler Bonnen, Stephanie Fu, Yutong Bai, Thomas O'Connell, Yoni Friedman, Nancy Kanwisher, Joshua B Tenenbaum, and Alexei A Efros. Evaluating multiview object consistency in humans and image models. *Advances in Neural Information Processing Systems*, 37:43533–43548, 2024.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Xuweiyi Chen, Markus Marks, and Zezhou Cheng. Probing the mid-level vision capabilities of self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30095–30105, 2025.

Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999. PMLR, 2016.

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljačić. Equivariant contrastive learning. *arXiv preprint arXiv:2111.00899*, 2021.

Duolikun Danier, Mehmet Aygün, Changjian Li, Hakan Bilen, and Oisin Mac Aodha. Depthcues: Evaluating monocular depth perception in large vision models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20049–20059, 2025.

Alexandre Devillers and Mathieu Lefort. Equimod: An equivariance module to improve visual instance discrimination. In *International Conference on Learning Representations*, 2023.

Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1755–1763. PMLR, 2019.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21795–21806, 2024.

Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant multi-view networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1568–1577, 2019.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. AdaWorld: Learning adaptable world models with latent actions. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18744–18771. PMLR, 2025.

Quentin Garrido, Laurent Najman, and Yann LeCun. Self-supervised learning of split invariant equivariant representations. *arXiv preprint arXiv:2302.10283*, 2023.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 8387–8406. PMLR, 2022.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Mhb5fpA1T0.

Athinoulla Konstantinou, Georgios Leontidis, Mamatha Thota, and Aiden Durrant. EquiCaps: Predictor-free pose-aware pre-trained capsule networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7947–7957, 2025.

Hankook Lee, Kibok Lee, Kimin Lee, Honglak Lee, and Jinwoo Shin. Improving transferability of representations via augmentation-aware self-supervision. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17710–17722, 2021.

Willi Menapace, Stephane Lathuiliere, Sergey Tulyakov, Aliaksandr Siarohin, and Elisa Ricci. Playable video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10061–10070, 2021.

Thomas P O'Connell, Tyler Bonnen, Yoni Friedman, Ayush Tewari, Josh B Tenenbaum, Vincent Sitzmann, and Nancy Kanwisher. Approaching human 3d shape perception with neurally mappable models. *arXiv preprint arXiv:2308.11300*, 2023.

Netta Ollikka, Amro Abbas, Andrea Perin, Markku Kilpeläinen, and Stéphane Deny. A comparison between humans and ai at recognizing objects in unusual poses. *arXiv preprint arXiv:2402.03973*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Jung Yeon Park, Ondrej Biza, Linfeng Zhao, Jan-Willem van de Meent, and Robin Walters. Learning symmetric embeddings for equivariant world models. *arXiv preprint arXiv:2204.11371*, 2022.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Guo-Jun Qi, Liheng Zhang, Chang Wen Chen, and Qi Tian. AVT: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8130–8139, 2019.

Irvin Rock and Joseph DiVita. A case of viewer-centered object perception. *Cognitive psychology*, 19(2):280–293, 1987.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations*, 2024.

Roger N. Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. doi: 10.1126/science.171.3972.701.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Michael J Tarr and Heinrich H Bülthoff. Image-based object recognition in man, monkey and machine. *Cognition*, 67(1-2):1–20, 1998.

Shimon Ullman. Three-dimensional object recognition based on the combination of views. *Cognition*, 67(1-2):21–44, 1998.

Yuwen Xiong, Mengye Ren, Wenyuan Zeng, and Raquel Urtasun. Self-supervised representation learning from flow equivariance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10191–10200, 2021.

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

Jaemyung Yu, Jaehyun Choi, Dong-Jae Lee, HyeongGwon Hong, and Junmo Kim. Self-supervised transformation learning for equivariant representations. In *Advances in Neural Information Processing Systems*, 2024.

Tianqiu Zhang, Muyang Lyu, Yufan Zhang, Fang Fang, and Si Wu. DiLA: Disentangled latent action world models. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=BRBHruBDkb`.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning, 2025. URL `https://openreview.net/forum?id=GARbxyCV13`.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5745–5753, 2019.

A APPENDIX A

Algorithm 1 Planning for object matching with CEM refinement

Require: Query x_q, match x_m, lure x_ℓ, encoder E, transition model T
Require: $K_{\max} = 10$, initial candidates $N = 256$, CEM refinement iterations $I = 2$, samples per refinement $M = 256$, elite count $M_e = 26$
Require: Rotation angle θ, scoring mode $\mathrm{mode} \in \{\mathrm{uni}, \mathrm{bi}\}$; $d(z, z') = \|z - z'\|_2^2$

Encode $z_q = E(x_q)$, $z_m = E(x_m)$, $z_\ell = E(x_\ell)$.
for $z_t \in \{z_m, z_\ell\}$ **do**
  Sample N random unit rotation axes and form actions $\{a_n\}_{n=1}^N$ with angle θ.
  for each action a_n **do**
    if $\mathrm{mode} = \mathrm{uni}$ **then**
      For each $k \le K_{\max}$, compute

$$L_{n,k} = d\Big(T^k(z_q, a_n), z_t\Big).$$

    else if $\mathrm{mode} = \mathrm{bi}$ **then**
      For each $k \le K_{\max}$, compute

$$L_{n,k} = d\Big(T^{\lceil k/2 \rceil}(z_q, a_n), T^{k - \lceil k/2 \rceil}(z_t, a_n^{-1})\Big).$$

    end if
  end for
  Assign each initial action its best loss:

$$\bar{L}_n = \min_{k \le K_{\max}} L_{n,k}.$$

  Select the M_e initial actions with the smallest $\bar{L}_n$ and initialize the CEM distribution from their rotation axes.
  for $i = 1, \ldots, I$ **do**
    Sample M candidate actions $\{a_j^{(i)}\}_{j=1}^M$ from the current CEM distribution.
    for each action $a_j^{(i)}$ **do**
      if $\mathrm{mode} = \mathrm{uni}$ **then**
        For each $k \le K_{\max}$, compute

$$L_{j,k}^{(i)} = d\Big(T^k(z_q, a_j^{(i)}), z_t\Big).$$

      else if $\mathrm{mode} = \mathrm{bi}$ **then**
        For each $k \le K_{\max}$, compute

$$L_{j,k}^{(i)} = d\Big(T^{\lceil k/2 \rceil}(z_q, a_j^{(i)}), T^{k - \lceil k/2 \rceil}(z_t, (a_j^{(i)})^{-1})\Big).$$

      end if
    end for
    Assign each action its best loss over all steps:

$$\bar{L}_j^{(i)} = \min_{k \le K_{\max}} L_{j,k}^{(i)}.$$

    Select the M_e elite actions with smallest $\bar{L}_j^{(i)}$.
    Update the CEM sampling distribution from the elite rotation axes.
  end for
  Select the best action and rollout length over all evaluated candidates:

$$(a_t^\star, k_t^\star) = \arg\min_{a,k} L(a, k).$$

  Score target by the best matched step:

$$s(q, t) = -L(a_t^\star, k_t^\star).$$

end for
Predict match iff $s(q, m) > s(q, \ell)$.

CEM implementation details. The search uses three candidate generations of 256 rotation axes each: one initial global generation followed by two CEM refinement generations, for a total of $3 \times 256 = 768$ evaluated axes per source–target comparison. In each generation, the best 10% of candidates are retained as elites ($M_e = 26$). Initial rotation axes are sampled as

$$\mathbf{u} = \frac{\boldsymbol{\epsilon}}{\|\boldsymbol{\epsilon}\|_2}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I_3),$$

which yields uniform samples on S^2. For subsequent CEM generations, given the current mean direction $\boldsymbol{\mu}$ and spread σ, we sample

$$\boldsymbol{\xi} = (I - \boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{\epsilon}, \qquad \mathbf{u} = \frac{\boldsymbol{\mu} + \sigma\boldsymbol{\xi}}{\|\boldsymbol{\mu} + \sigma\boldsymbol{\xi}\|_2}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I_3).$$

Thus, the refinement distribution is a Gaussian in the tangent plane followed by normalization back onto S^2. The distribution parameters are updated from the elite samples using a smoothing coefficient of 0.2. We initialize the spread to $\sigma = 1.0$ and lower-bound it at 0.05.

B Network Architectures

Action inference network. The action inference model A_ϕ takes as input a pair of latent states (z_t, z_{t+1}), each represented as a $4 \times 28 \times 28$ latent tensor from the frozen VAE encoder. The two tensors are concatenated channel-wise to form an 8-channel input. The network consists of an initial convolutional stem followed by three residual convolutional stages with progressive spatial downsampling. Feature dimensionality increases from 64 to 256 channels across the network. After the final stage, global average pooling is applied and the resulting feature vector is processed by a two-layer MLP with GELU activations to predict a 6D rotation representation Zhou et al. (2019). The predicted representation is normalized and mapped to a valid rotation matrix using Gram–Schmidt orthogonalization.

Transition network. The transition model T_θ operates directly in the latent space of the frozen VAE encoder and predicts the next latent state conditioned on the current state and inferred action. The architecture follows a U-Net-style encoder–decoder design with skip connections. The input latent tensor is first projected to a higher-dimensional feature space using a convolutional stem. The encoder then progressively downsamples spatial resolution from 28×28 to 7×7 while increasing channel dimensionality.

Action conditioning is implemented using FiLM-modulated residual blocks throughout both the encoder and decoder. The inferred action vector is first normalized and then transformed through small MLPs that generate feature-wise affine modulation parameters for intermediate activations. The decoder mirrors the encoder using nearest-neighbor upsampling and skip connections from corresponding encoder stages.

C Training Details

All models were trained using the Adam optimizer Kingma & Ba (2014) with a learning rate of 10^{-3} and batch size 16. Training was performed for 800 epochs using two-step autoregressive prediction, where the predicted latent state $\hat{z}_{t+1}$ was recursively fed back into the transition model to predict $\hat{z}_{t+2}$.

To stabilize optimization, models were first trained for 100 epochs using only the forward prediction objective $\mathcal{L}_{\text{fwd}}$. The reverse prediction and angle regularization losses were introduced afterward. The forward and reverse prediction losses were weighted equally with coefficient 1.0, while the rotation magnitude regularization term used weight $\lambda_\alpha = 0.1$.

Training was performed on a single NVIDIA RTX 3080 Ti GPU with approximately 7 GB of GPU memory usage. One training epoch required roughly one minute.

C.1 Dataset Generation

We use two synthetic object-centric datasets. For ShapeNet Chang et al. (2015), we sample 800 objects from eight categories: airplane, birdhouse, car, chair, microwave, piano, pistol, sofa. The

Abstract dataset contains 1000 procedurally generated shapes designed to reduce semantic and category-level cues. For both datasets, we generate five sequences per object by sampling a 3D rotation axis and rendering views at 1° increments over a full 360° rotation. We reserve 10% of objects for testing, ensuring that test objects are not observed during training.