Learning Robust Representations for World Models without Reward Signals

Zeqiang Zhang
Institute of Neural Information Processing
Ulm University
Ulm, DE 89081
zeqiang.zhang@uni-ulm.de

Fabian Wurzberger
Institute of Neural Information Processing
Ulm University
Ulm, DE 89081
fabian.wurzberger@uni-ulm.de

Sebastian Gottwald
Institute of Neural Information Processing
Ulm University
Ulm, DE 89081
sebastian.gottwald@uni-ulm.de

Daniel A. Braun
Institute of Neural Information Processing
Ulm University
Ulm, DE 89081
daniel.braun@uni-ulm.de

Abstract

Learning accurate and generalizable world models is a central challenge in model-based reinforcement learning (MBRL), particularly in reward-free settings where no task-specific supervision is available. In this paper, we investigate how different unsupervised objectives, including reconstruction, inverse dynamics, and contrastive learning, capture distinct components of the observation space, such as noise, background, controllable dynamics, and slow-changing factors. Building on this understanding, we introduce a hybrid representation learning approach that integrates the strengths of multiple objectives to better capture predictable and task-relevant structure. We design a controlled shape-based environment with disentangled latent factors to evaluate the robustness and utility of learned representations. Empirical results show that our method yields more informative and generalizable representations.

1 Introduction

Reinforcement learning (RL) has achieved remarkable success across a wide range of domains, from game playing to robotics and real-world decision-making tasks [21, 19, 22]. A key driver of this progress has been the development of powerful representation learning methods and increasingly sophisticated model-based approaches. Compared to model-free methods, model-based RL leverages a learned model of the environment to enable planning and sample-efficient learning. As a result, model-based techniques have attracted growing attention in recent years [14].

Despite these advances, the majority of model-based RL algorithms are inherently task-driven, relying on reward signals to guide both representation learning and model construction [23, 5, 7, 8]. However, in many real-world scenarios, reward feedback is sparse, delayed, or entirely unavailable during the data collection phase. This raises an important open question: how can agents learn a useful and generalizable world model in a reward-free setting? Addressing this challenge is essential for building agents that can autonomously explore and develop robust internal models of their environment, even before being assigned a specific task.

A central component of the world model is the representation of the environment's dynamics. Learning robust and compact representations of observations is crucial, as these representations directly affect

18th European Workshop on Reinforcement Learning (EWRL 2025).

the quality of the learned dynamics model and, consequently, the downstream performance of the agent. Unsupervised representation learning methods offer a promising avenue for this purpose, enabling the agent to structure its understanding of the environment without relying on external rewards [1].

Several unsupervised approaches have been proposed in this context, including reconstruction-based learning (e.g., autoencoders) [10, 11], inverse dynamics models [16], and contrastive learning techniques [17]. While each of these methods has demonstrated effectiveness in isolating certain aspects of the observation space, they inherently emphasize different types of features. For example, reconstruction tends to capture all information, including irrelevant or unpredictable details; inverse models often focus on features that are controllable by the agent, and contrastive methods prioritize temporally stable or discriminative features.

In this work, we analyze these unsupervised learning paradigms from a unified perspective, highlighting their respective biases and limitations in terms of the information they preserve. Building on this analysis, we introduce a novel method that integrates multiple learning objectives to construct representations that better capture the predictable structure of the environment. Our approach is designed to balance the strengths of different representation learning strategies while explicitly avoiding reliance on reward signals.

To evaluate our approach, we design a simple dynamic environment composed of shapes that exhibit diverse types of features, including noise, random features, and both short-term and long-term dynamics. This environment allows for precise analysis of what information is retained in the learned representations. We show that our method produces representations that are more robust, generalizable, and informative for downstream tasks such as world model learning. These results highlight the potential of our approach in reward-free settings. For future work, we plan to integrate our representation learning method into modern world model architectures, such as Dreamer, and evaluate its effectiveness in more complex and high-dimensional environments.

2 Related work

Model-based reinforcement learning (MBRL) aims to improve sample efficiency and generalization by explicitly learning a model of environment dynamics for planning and policy optimization. Early examples include Dyna-style methods [20], which integrate learning and planning, and PILCO [3], which pioneered probabilistic models for data-efficient control. More recently, latent dynamics models such as PlaNet [6] and Dreamer [5, 7, 8] have demonstrated impressive performance by learning compact latent spaces that support long-horizon prediction and imagination-based planning. However, these approaches typically rely on task-specific reward signals to shape both the learned model and the underlying state representation. In contrast, our work seeks to decouple model learning from reward supervision by developing a representation learning framework tailored to constructing effective world models in reward-free settings.

The challenge of learning meaningful representations without reward signals has been a long-standing objective in both supervised and reinforcement learning contexts. Reconstruction-based methods, such as auto-encoders [10, 11], aim to compress high-dimensional observations into lower-dimensional representations that retain information necessary to reconstruct the input. While effective for preserving general features, these methods often encode irrelevant or unpredictable aspects of the input. Inverse dynamics models [16] take a different perspective by predicting the action taken between consecutive observations, encouraging the model to focus on controllable or agent-relevant features. Meanwhile, contrastive learning techniques, including Contrastive Predictive Coding (CPC) [15] and CURL [13], promote representations that capture temporally consistent or discriminative features by distinguishing true future states from distractors. Although each of these methods contributes valuable inductive biases, they inherently emphasize different aspects of the observation space. Our approach builds upon this insight by systematically analyzing the feature preferences of each method and proposing a unified objective that encourages representations optimized for predictability and generalization in the context of world model learning.

While much prior work in reward-free learning focuses on intrinsic motivation, such as curiosity [16, 4], empowerment [12], or novelty-based exploration [2], our aim is not to drive exploration but to improve the quality of internal representations learned in the absence of external rewards. In this regard, our work is more closely related to task-agnostic world models [18], which learn

general-purpose dynamics that support downstream adaptation without reward supervision. Our contribution is complementary, as we focus specifically on the representation learning module and its impact on the quality and robustness of the learned world model in fully unsupervised environments.

3 Background

3.1 Learning through interactions

We formulate the problem of learning a world model as a Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = (\mathcal{O}, \mathcal{S}, \mathcal{A}, \mathcal{P})$, where $\mathcal{O}$ is the observation space, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the state transition function. The agent interacts with the environment by taking actions $a_t \in \mathcal{A}$ and observing the next observation $o_{t+1} \in \mathcal{O}$ at each time step t. The observation o_t is a noisy and high-dimensional representation depending on the underlying state $s_t \in \mathcal{S}$, which is unobservable and may include irrelevant or unpredictable features. The agent's goal is to learn a model of the environment dynamics that can be used for planning and decision-making. The model is learned by interacting with the environment and collecting a dataset of state transitions $\mathcal{D} = \{(o_t, a_t, o_{t+1})\}_{t=1}^{T}$.

3.2 Reconstruction

Reconstruction-based methods aim to learn compact representations of observations by employing an encoder–decoder architecture. The encoder maps a high-dimensional observation o_t into a lower-dimensional latent representation $z_t = f(o_t)$, and the decoder attempts to reconstruct the original observation from this latent code, producing $\hat{o}_t = g(z_t)$. The learning objective is to minimize the reconstruction loss, typically defined as:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathcal{D}} \left[||o_2 - g(f(o_1))||^2 \right], \tag{1}$$

where $|| \cdot ||$ denotes the Euclidean norm. Reconstruction can be performed in two forms: (i) single-step reconstruction, where the model attempts to reconstruct the input observation itself (i.e., $o_2 = o_1 = o_t$); or (ii) predictive reconstruction, where the decoder reconstructs the next observation (i.e., $o_2 = o_{t+1}$, $o_1 = o_t$) as a forward dynamics model.

3.3 Inverse dynamics

Inverse dynamics models learn representations by predicting the action that caused a transition between two observations. Given consecutive observations o_t and o_{t+1}, the encoder produces latent representations $z_t = f(o_t)$ and $z_{t+1} = f(o_{t+1})$. An inverse model h then predicts the action $\hat{a}_t$ from these latent features. The objective is to minimize the prediction loss

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_{\mathcal{D}} \left[\ell(a_t, h(z_t, z_{t+1})) \right], \tag{2}$$

where ℓ denotes a task-appropriate loss function. This encourages the latent space to preserve information about agent-controllable features.

3.4 Contrastive learning

Contrastive learning aims to learn representations by distinguishing between similar (positive) and dissimilar (negative) pairs of observations. In our setting, we use temporal proximity to define these relationships: observations that are temporally close (e.g., o_t and o_{t+1}) form positive pairs, while randomly sampled observations serve as negative pairs.

Let $z_1 = f(o_1)$ and $z_2 = f(o_2)$ be latent representations produced by an encoder. A contrastive discriminator $h(z_1, z_2)$ estimates the probability that the pair is positive. We employ a binary-NCE loss, also known as InfoMax loss [9]:

$$\mathcal{L}_{\text{con}} = -\mathbb{E}_{\mathcal{D}} \left[y \log h(z_1, z_2) + (1 - y) \log(1 - h(z_1, z_2)) \right], \tag{3}$$

where $y = 1$ for positive pairs and $y = 0$ for negative pairs. This objective encourages the encoder to preserve temporally predictable features while discarding irrelevant or unpredictable components.

4 A shape environment

To analyze the representational properties of different unsupervised objectives in a controlled and interpretable setting, we design a synthetic environment, referred to as the Shape Environment. This environment features a discrete action space and structured visual observations, facilitating the disentanglement of different latent factors.

4.1 Environment design

The agent interacts with the environment where the observation is an image with a single geometric object rendered on a noisy background. The object varies along several axes:

- **Shape and Scale:** There are four geometric shapes (circle, triangle, square, and pentagon) arranged in a fixed cycle. Each shape appears in one of three sizes: small, medium, or large. When the agent applies the *forward shape* action, the object increases in size; once it reaches the largest size, it transitions to the next shape in its smallest form. The *backward shape* action reverses this progression.
- **Color:** The object color cycles through red $\rightarrow$ green $\rightarrow$ blue $\rightarrow$ black and vice versa, controlled by the other two discrete actions (*forward color* and *backward color*).
- **Position:** The spatial location of the object is randomized independently at each time step and does not respond to agent actions.
- **Noise:** Gaussian noise is added to the entire observation to introduce stochastic variation

The agent's action space is thus composed of four discrete actions, corresponding to the forward/backward transitions of shape and color. Figure 1 illustrates all 12 shape-scale combinations under all colors with four possible actions. Further implementation details are provided in the appendix A.

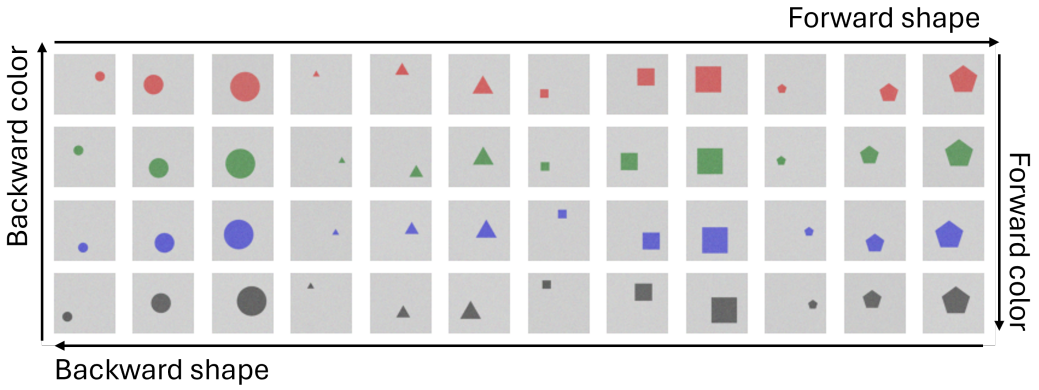

Figure 1: All shape-scale combinations under all colors.

4.2 Features of the environment

The Shape environment contains diverse observational features that differ in predictability and semantic relevance. Noise is entirely stochastic and neither predictable nor reconstructable, acting as irrelevant variation that should ideally be ignored by the representation. Position, though fully reconstructable from the observation, is unpredictable due to random sampling at each step. It is often retained by reconstruction-based methods despite being semantically uninformative. Shape and size form a coupled, action-dependent dynamic; size changes more rapidly and triggers shape transitions, making both features highly predictable and essential for modeling controllable structure. However, note that only size information is essential to predict actions between neighbor observations. In contrast, color evolves independently of shape and size. Its small discrete state space and weak temporal continuity make it harder to reliably model using temporal objectives such as contrastive

learning. This decomposition illustrates the challenges in disentangling useful dynamics from distractors and guides the design of robust representation objectives.

4.3 Forward model fails without robust abstraction

To demonstrate the limitations of using naive forward models in our environment, we train a simple model that predicts the next observation $\hat{o}_{t+1}$ based on the current observation o_t and action a_t. The model is trained by minimizing a standard reconstruction loss between $\hat{o}_{t+1}$ and the ground truth observation o_{t+1}. Implementation details are provided in Appendix B.

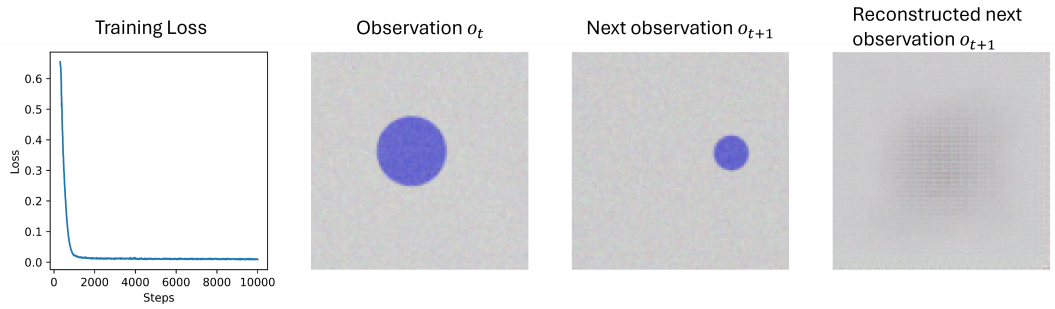

Figure 2: Training loss and an example of the forward model's reconstruction.

As shown in the left panel of Figure 2, the training loss converges rapidly, suggesting that the model successfully optimizes its objective. However, visual inspection of the reconstructed observations (right panel) reveals that the model fails to capture the meaningful structure of the environment. This failure stems from the model's reliance on raw pixel inputs, which are dominated by unpredictable and irrelevant variations such as noise and position. These results underscore the necessity of learning robust, disentangled representations that abstract away nuisance factors and preserve the predictable, controllable structure essential for world modeling.

5 Learning robust representations

Our model comprises six components, as illustrated in Figure 3. It includes two encoders ϕ and ϕ^R, a decoder g, an inverse model h, a forward model f, and a contrastive discriminator d.

The primary encoder $\phi(o_t)$ maps the observation o_t into a latent representation z_t. In this latent space, the inverse model $h(z_t, z_{t+1})$ predicts the action $\hat{a}_t$ taken between two consecutive observations, and the forward model $f(z_t, a_t)$ predicts the next-step latent representation $\hat{z}_{t+1}$. The contrastive discriminator $d(z, z')$ estimates the probability that two latent vectors are temporally adjacent. These three components operate on the latent space produced by ϕ, and collectively encourage it to encode predictable and action-relevant features.

A second encoder ϕ^R is dedicated to reconstruction. It maps the observation o_t to a latent representation z_t^R, which is then combined with z_t by the decoder g to reconstruct the current observation $\hat{o}_t = g(z_t, z_t^R)$.

This dual-encoder design aims to disentangle latent factors: the encoder ϕ focuses on capturing controllable and temporally predictable structure, while the encoder ϕ^R absorbs residual information necessary for high-fidelity reconstruction.

The trainable components are listed in the following:

$$
\begin{array}{ll}
\text{Encoder } \phi\text{:} & z_t = \phi(o_t) \\
\text{Encoder } \phi^R\text{:} & z_t^R = \phi^R(o_t) \\
\text{Decoder:} & \hat{o}_t = g(z_t, z_t^R) \\
\text{Inverse model:} & \hat{a}_t = h(z_t, z_{t+1}) \\
\text{Forward model:} & \hat{z}_{t+1} = f(z_t, a_t) \\
\text{Contrastive discriminator:} & d(z, z') = 1 \text{ if } z \text{ and } z' \text{ are neighbours}
\end{array}
\tag{4}
$$

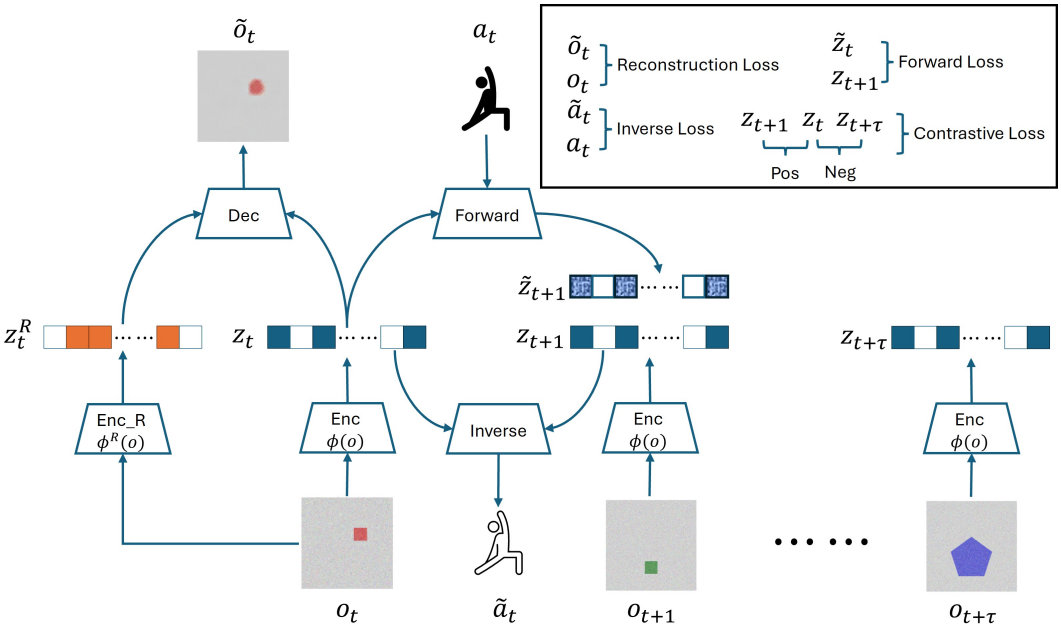

Figure 3: The architecture of our representation learning model.

Given a batch of observation-action-next observation triplets $\mathcal{D} = \{(o_t, a_t, o_{t+1})\}$ and a batch of random observations $\mathcal{D}' = \{(o_{t'})\}$, the entire model is trained end-to-end using a multi-objective loss function that combines reconstruction loss $\mathcal{L}_{\text{rec}}$, inverse dynamics loss $\mathcal{L}_{\text{inv}}$, forward dynamics loss $\mathcal{L}_{\text{fwd}}$, and contrastive learning loss $\mathcal{L}_{\text{con}}$:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{inv}} + \mathcal{L}_{\text{fwd}} + \mathcal{L}_{\text{con}}, \tag{5}$$

where each component is defined as follows:

$$\begin{array}{ll} \text{Reconstruction loss:} & \mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathcal{D}} \left[||o_t - g(z_t, z_t^R)||^2 \right] \\ \text{Inverse dynamics loss:} & \mathcal{L}_{\text{inv}} = \mathbb{E}_{\mathcal{D}} \left[\ell(a_t, h(z_t, z_{t+1})) \right] \\ \text{Forward dynamics loss:} & \mathcal{L}_{\text{fwd}} = \mathbb{E}_{\mathcal{D}} \left[||z_{t+1} - f(z_t, a_t)||^2 \right] \\ \text{Contrastive loss:} & \mathcal{L}_{\text{con}} = -\mathbb{E}_{\mathcal{D},\mathcal{D}'} \left[\log d(z_t, z_{t+1}) + \log(1 - d(z_t, z')) \right] \end{array} \tag{6}$$

where ℓ denotes a task-appropriate loss function, such as cross-entropy for the discrete action prediction in the inverse model. The whole process is summarized in Algorithm 1.

6 Experiments

We evaluate our proposed representation learning method in the Shape Environment. Full implementation details are provided in Appendix D. We compare our method against four baselines: (i) a reconstruction-based encoder, (ii) an inverse dynamics-based encoder, (iii) a contrastive learning-based encoder, and (iv) a classifier-based encoder. Each baseline jointly trains an encoder and a latent forward model in an end-to-end manner. The classifier-based encoder is trained with direct supervision to predict the object's color, size, and shape from the observation. While this approach relies on external knowledge not available to the other approaches, it serves as a useful upper bound for assessing representation quality. Details of all comparison methods are provided in Appendix C.

To quantitatively evaluate the learned representations, we train classifiers on the frozen latent embeddings z_t to predict object properties (shape, size, and color). This allows us to assess how well each method captures semantically meaningful features. In the first subsection, we analyze the structure and content of the latent space; in the second, we examine representation robustness by using the learned embeddings for multi-steps prediction via a latent forward model.

Algorithm 1 Training procedure for the representation learning model.

Initialize: Encoders ϕ, ϕ^R, decoder g, inverse model h, forward model f, contrastive discriminator d with random weights, replay buffer $\mathcal{R}$
while not converged **do**
  for Interaction **do**
    Sample an action $a_t \sim \pi(a_t \mid o_t)$ from the policy π.
    Observe the next observation o_{t+1} from the environment.
    Store the transition (o_t, a_t, o_{t+1}) in the replay buffer $\mathcal{R}$.
  end for
  for Training **do**
    Sample a batch of transitions $\mathcal{D} = \{(o_t, a_t, o_{t+1})\}$ from the replay buffer $\mathcal{R}$.
    Sample a batch of random observations $\mathcal{D}' = \{(o_{t'})\}$ from the replay buffer $\mathcal{R}$.
    Compute latent representations: $z_t = \phi(o_t)$, $z_t^R = \phi^R(o_t)$, $z_{t+1} = \phi(o_{t+1})$, $z' = \phi(o')$.
    Compute predicted variables: $\hat{o}_t = g(z_t, z_t^R)$, $\hat{a}_t = h(z_t, z_{t+1})$, $\hat{z}_{t+1} = f(z_t, a_t)$.
    Compute losses for each component according the Equation 6.
    Compute the total loss $\mathcal{L}$ according to Equation 5.
    Backpropagate gradients and update parameters of all components.
  end for
end while
Return: Learned representations ϕ, ϕ^R, decoder g, inverse model h, forward model f, contrastive discriminator d

6.1 Representation space analysis

We evaluate how well the learned representations capture semantic information by training a classifier to predict the object's shape, size, and color from the latent embeddings z_t. The classification accuracies for each property and method are summarized in Table 1 and visualized in Figure 4. Each entry reports the accuracy achieved by a classifier trained on representations from the corresponding encoding method. The experiments are repeated for 5 times with random initialization.

Table 1: Classification accuracy of different representation learning methods.

Algorithm	Color	Size	Shape
Reconstruction-based encoder	$99.66\% \pm 0.38\%$	$74.78\% \pm 3.10\%$	$43.46\% \pm 4.09\%$
Inverse dynamics-based encoder	$100.00\% \pm 0.00\%$	$100.00\% \pm 0.00\%$	$87.64\% \pm 11.64\%$
Contrastive learning-based encoder	$27.84\% \pm 5.77\%$	$46.22\% \pm 26.92\%$	$40.24\% \pm 29.80\%$
Classifier-based encoder	$100\% \pm 0.00\%$	$100\% \pm 0.00\%$	$99.92\% \pm 0.12\%$
Our model	$99.42\% \pm 1.16\%$	$100\% \pm 0.00\%$	$97.36\% \pm 3.78\%$

Our method achieves consistently high accuracy across all three properties, closely approaching the performance of the supervised classifier-based encoder. This suggests that our approach effectively captures the essential, disentangled features of the environment in an unsupervised manner. The inverse dynamics-based encoder performs well on color and size but shows a significant drop in shape prediction accuracy. Since size alone is sufficient to infer the agent's action, the inverse model lacks incentive to fully encode shape information, explaining its limited generalization. The reconstruction-based encoder captures color relatively well but performs poorly on shape and size. This is likely because reconstruction loss tends to prioritize pixel-level fidelity, and the variability introduced by noise makes structural features harder to reconstruct. The contrastive learning-based encoder yields the lowest performance across all properties, likely due to a misalignment between the contrastive objective and the forward prediction task during end-to-end training. This misalignment also manifests in higher variance in prediction accuracy, which is partially reflected in our method as well. Addressing this incompatibility between objectives remains an important direction for future work.

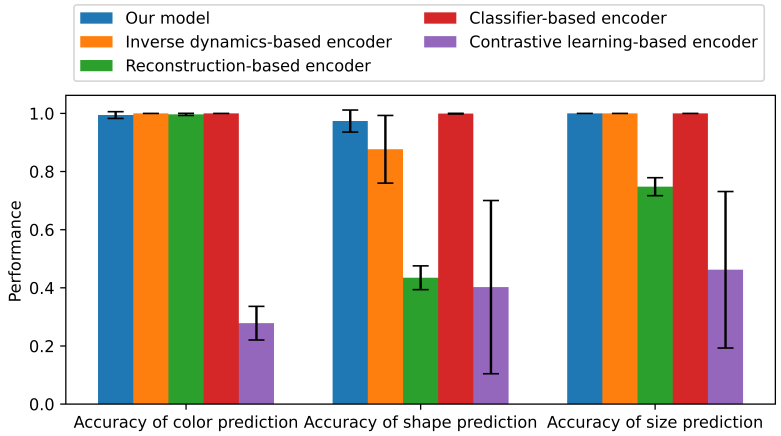

Figure 4: Classification accuracy of different representation learning methods.

7 Latent split check: what is in z vs. z^R

To directly test whether the two encoders specialize as intended, we evaluate how well common attributes can be predicted from each latent separately. Concretely, we freeze the learned encoders and train classifiers or regressors on top of either $z = \phi(o)$ or $z^R = \phi^R(o)$ to predict color, shape, size, as well as positions. For position prediction, we consider it accurate if the mean-squared error between predicted value and real value is less than 0.0025. Heads are trained on frozen embeddings with the same train/validation protocol as in the main experiments. No gradients are propagated into the encoders.

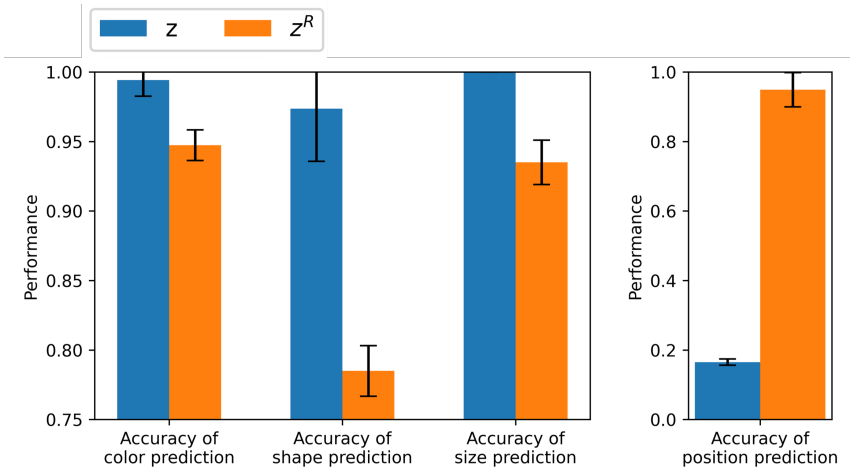

Figure 5: Classification/regression accuracy based on representation z and z^R.

Figure 5 summarizes the outcomes. We observe:

1. **Position is easy from z^R but hard from z.** The regressor trained on z^R attains low error, while the same regressor on z performs near chance, indicating that randomized, non-predictive nuisance factors (position) are largely stored in z^R and suppressed in z.
2. **Predictable and controllable factors are stronger in z.** For color, shape, and size, classifiers trained on z achieve higher accuracy than those trained on z^R, although these attributes are still decodable from z^R to a non-trivial degree. This shows that although z^R retains some parts of residual semantic content that aids reconstruction, z concentrates the predictable and controllable structure more.

Overall, these results support our design assumption: the dual-encoder architecture effects a soft split of information, with z prioritizing temporally predictable, action-coupled structure and z^R absorbing non-predictive variability needed for pixel-space fidelity.

7.1 Representation with forward model

To evaluate the robustness of the learned representations over time, we use a latent forward model to predict the next latent state $\hat{z}_{t+1}$ from the current latent state z_t and action a_t. We then use the predicted latent representation with the trained classifiers to infer the properties of the next observation, specifically the object's shape, size, and color. This process is unrolled for up to five steps to assess how predictive information is preserved across time. The degradation in classification accuracy over prediction horizons is shown in Figure . Detailed numerical results are presented in Appendix E, and confusion matrices for each prediction step are included in Appendix F.

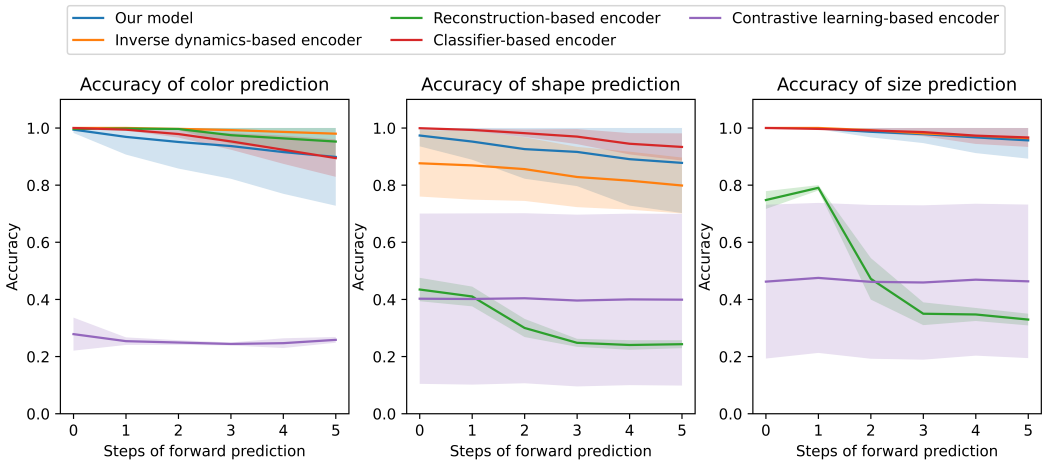

Figure 6: Classification accuracy of different representation learning methods with multi-step predictions.

The results show that the representations learned by our method maintain high predictive accuracy across all three object properties, even at longer prediction horizons. This suggests that our approach yields robust and temporally stable representations. The inverse dynamics-based encoder also demonstrates temporal robustness but consistently underperforms our method on shape prediction, likely due to its limited incentive to model features not directly related to action inference. In contrast, the reconstruction-based encoder suffers a rapid decline in accuracy—particularly for shape and size—as it retains non-predictive factors such as position, which introduce noise and degrade long-term prediction quality. The contrastive learning-based encoder exhibits relatively stable performance over time but consistently achieves the lowest overall accuracy, particularly for color. This is likely because color changes rapidly and it is difficult for the model to identify which observations are temporally related, making it harder to learn consistent color features through contrastive objectives.

8 Conclusion and future work

In this work, we investigated the problem of learning robust and generalizable representations for world models in the absence of reward signals. We introduced a novel representation learning approach that combines multiple unsupervised learning signals to capture the predictable and structured components of the environment. Through a carefully designed synthetic Shape environment, we demonstrated that our method learns latent representations that are not only informative and disentangled, but also predictive over long horizons, outperforming several commonly used baselines.

While our results show promise, several important directions remain for future work. First, we plan to evaluate the scalability and generalization of our method in more complex and diverse environments

with richer dynamics and visual complexity. Second, we aim to integrate our representation learning approach into modern model-based reinforcement learning architectures, such as those using recurrent state-space models (RSSM), to better exploit temporal structure and uncertainty modeling. Finally, although our current study focuses solely on unsupervised representation learning, we intend to explore how these representations can be jointly optimized with policy learning. Specifically, we aim to investigate how active data collection strategies can accelerate world model convergence and improve sample efficiency.

References

[1] Nicolò Botteghi, Mannes Poel, and Christoph Brune. Unsupervised representation learning in deep reinforcement learning: A review. *IEEE Control Systems*, 45(2):26–68, 2025.

[2] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[3] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

[4] Nick Haber, Damian Mrowca, Stephanie Wang, Li F Fei-Fei, and Daniel L Yamins. Learning to play with intrinsically-motivated, self-aware agents. *Advances in neural information processing systems*, 31, 2018.

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

[6] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[7] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[8] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pages 1–7, 2025.

[9] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[10] Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *Second international conference on learning representations, ICLR*, volume 19, page 121, 2014.

[11] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[12] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, volume 1, pages 128–135. IEEE, 2005.

[13] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pages 5639–5650. PMLR, 2020.

[14] Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. A survey on model-based reinforcement learning. *Science China Information Sciences*, 67(2):121101, 2024.

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[16] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[17] Felix Schneider, Xia Xu, Markus R Ernst, Zhengyang Yu, and Jochen Triesch. Contrastive learning through time. In *SVRHM 2021 Workshop@ NeurIPS*, 2021.

[18] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International conference on machine learning*, pages 8583–8592. PMLR, 2020.

[19] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[20] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

[21] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[22] Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28694–28698, 2025.

[23] Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. Solar: Deep structured representations for model-based reinforcement learning. In *International conference on machine learning*, pages 7444–7453. PMLR, 2019.

A Shape environment details

The observations in the Shape environment are rendered as images with an initial resolution of 144×144 pixels and 4 channels (RGB and alpha). These images are subsequently rescaled to 96×96 and normalized to have pixel values in the range $[0, 1]$.

To simulate realistic sensory noise, Gaussian noise with mean 0 and variance 0.1 is added to the rendered image. After noise injection, the image is clipped and renormalized to ensure all pixel values remain within the $[0, 1]$ interval.

The position of the geometric shape in each image is randomized using a uniform distribution, constrained to ensure that the shape is fully contained within the image boundaries. All shapes—circle, triangle, square, and pentagon—are defined such that the same size level (small, medium, large) corresponds to the same radius value. However, due to their differing geometries, the resulting areas vary across shape types.

The mapping between shape type, size, and the corresponding radius and area is summarized in Table 2. Here the number is with respect to the original figure (144×144).

Table 2: Parameters for shapes.

	Small	**Medium**	**Large**
Circle	$r = 15$, $S = 707$	$r = 30$, $S = 2827$	$r = 45$, $S = 6362$
Triangle	$r = 15$, $S = 292$	$r = 30$, $S = 1169$	$r = 45$, $S = 2630$
Square	$r = 15$, $S = 450$	$r = 30$, $S = 1800$	$r = 45$, $S = 4050$
Pentagon	$r = 15$, $S = 535$	$r = 30$, $S = 2140$	$r = 45$, $S = 4815$

B The simple forward model

We implement a simple autoencoding forward model based on convolutional and transposed convolutional layers.

B.1 Model architecture

The model consists of two main components:

- **Encoder**:A convolutional neural network that maps a 4-channel input image of size 96×96 into a latent representation conditioned on the discrete action taken by the agent. The encoder outputs a latent tensor of shape `(hidden, action)`, which is projected via a batch matrix multiplication with a one-hot action vector to produce an action-conditioned latent representation. The hidden dimension is 128.
- **Decoder**: A transposed convolutional network that reconstructs the next observation $\hat{o}_{t+1}$ from the latent representation.

The forward pass can be described as:

$$\hat{o}_{t+1} = \text{Decoder}(\text{Encoder}(o_t) \cdot a_t) \tag{7}$$

where a_t is a one-hot encoding of the discrete action.

B.2 Training procedure

We collect transitions (o_t, a_t, o_{t+1}) into a replay buffer. Once a sufficient number of transitions are gathered, we begin training the forward model using mean squared error (MSE) loss between the predicted and true next observations.

The model is optimized with the Adam optimizer and a learning rate of 0.001. Training proceeds for 10,000 steps, and the model is updated every 10 steps.

C Comparison algorithms

We compare our method against four baselines that represent common paradigms in self-supervised representation learning. Each method consists of an encoder $\phi(o_t)$ that maps the observation o_t to a latent representation z_t, and a latent forward model $f(z_t, a_t)$ that predicts $\hat{z}_{t+1}$ from z_t and action a_t. All models are trained end-to-end by combining the respective representation learning loss with a forward prediction loss. The forward loss is defined as:

$$\mathcal{L}_{\text{fwd}} = \mathbb{E}_{\mathcal{D}} \left[||z_{t+1} - f(z_t, a_t)||^2 \right] \tag{8}$$

C.1 Reconstruction-based encoder

This method learns a representation by reconstructing the original observation. An encoder maps o_t to z_t, and a decoder $g(z_t)$ reconstructs $\hat{o}_t$ from z_t by optimizing a reconstruction loss:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathcal{D}} \left[||o_t - g(z_t)||^2 \right]. \tag{9}$$

The total loss is the sum of reconstruction and forward losses.

C.2 Inverse dynamics-based encoder

In this setup, the encoder maps both o_t and o_{t+1} to z_t and z_{t+1} respectively. An inverse model $h(z_t, z_{t+1})$ is trained to predict the action $\hat{a}_t$ that led from z_t to z_{t+1}, using a cross-entropy loss:

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_{\mathcal{D}} \left[\text{CE}(h(z_t, z_{t+1}), a_t) \right]. \tag{10}$$

A forward model also predicts $\hat{z}_{t+1}$ from z_t and a_t, and the two losses are summed to form the training objective.

C.3 Contrastive learning-based encoder

This approach uses contrastive learning to distinguish between neighboring and non-neighboring observations in latent space. The encoder maps o_t to z_t, and a contrastive discriminator is trained to maximize agreement between temporally adjacent representations while minimizing agreement with negative samples. The contrastive loss is defined as:

$$\mathcal{L}_{\text{con}} = -\mathbb{E}_{\mathcal{D},\mathcal{D}'} \left[\log d(z_t, z_{t+1}) + \log(1 - d(z_t, z')) \right]. \tag{11}$$

A forward model is trained concurrently with a forward loss, and the two losses are optimized jointly.

C.4 Classifier-based encoder

This model incorporates strong supervision by explicitly optimizing for features relevant to the task. The encoder maps o_t to z_t, which is then passed to three classifiers predicting the object's color, shape, and size. The classification loss is the sum of three cross-entropy losses:

$$\mathcal{L}_{\text{classification}} = \mathbb{E}_{\mathcal{D}} \left[\text{CE}_{\text{Color}} + \text{CE}_{\text{Shape}} + \text{CE}_{\text{Size}} \right]. \tag{12}$$

This loss is combined with the forward prediction loss to train the model. Note that this method uses the extra information (the ground truth of the object's color, shape, and size) to train the encoder.

D Implementation details

This section describes the architectural components and training pipeline used across all experiments.

D.1 Model architecture

D.1.1 Encoder ϕ, ϕ^R

The encoder ϕ processes a $4 \times 96 \times 96$ input image using three convolutional layers with ReLU activations. The output is flattened and projected into a latent representation via a fully connected layer. The encoder ϕ^R (used for reconstruction loss) shares the same architecture.

D.1.2 Decoder g

The decoder g reconstructs the original observation from a concatenated latent vector (z^R, z) for our algorithm, and directly from the latent vector z for the reconstruction-based encoder. It uses a linear projection followed by three transposed convolution layers to upsample back to the original resolution.

D.1.3 Inverse model h

The inverse model h predicts the action a_t given two consecutive latent representations (z_t, z_{t+1}). It consists of three fully connected layers with ReLU activations.

D.1.4 Forward model f

The forward model f predicts z_{t+1} from z_t and action a_t. The action is one-hot encoded and used to index the corresponding transformation slice from a reshaped output tensor.

D.1.5 Contrastive discriminator d

The discriminator d in the InfoMax framework takes two latent vectors (z_t, z') and outputs a scalar score indicating similarity. It is implemented as an MLP with ReLU activations.

D.1.6 Classifier

The classifier is defined as a multi-layer MLP. Object attributes (color, shape, size) are predicted using separate classifiers.

D.2 Training procedure

We collect transitions (o_t, a_t, o_{t+1}) into a replay buffer. Once a sufficient number of transitions are gathered, we employ a multi-objective training setup that combines all applicable losses depending on the encoder type. Total loss is backpropagated and used to update all parameters jointly.

The model is optimized with the Adam optimizer and a learning rate of 0.001. Training proceeds for 1,000,000 steps, and the model is updated every 10 steps. The latent dimensions of z and z^R are set to 128 for all of the algorithms.

D.3 Evaluation procedure

To assess the quality of the learned representations, we perform downstream classification tasks on object attributes: color, shape, and size. The evaluation is conducted in two stages:

D.3.1 Latent representation evaluation

We first freeze the encoder trained by either our model or one of the comparison algorithms. Then, we train three independent classifiers to predict object color, shape, and size from the latent representation z_t. The classifiers are trained for 2,000 epochs on a dataset containing 20,000 labeled examples. During testing, we randomly sample 1,000 observations, extract their latent representations using the frozen encoder, and perform attribute prediction using the trained classifiers.

D.3.2 Forward model evaluation

To evaluate the temporal consistency of the learned latent space, we use the forward model trained end-to-end alongside the encoder. We sample 1,000 initial observations and iteratively apply the forward model to generate multi-step predictions $\hat{z}_{t+\tau}$. At each prediction step, the pre-trained classifiers (from the previous evaluation) are used to predict object attributes based on the predicted latent representations.

E Accuracy table of classification based-on multi-step predictions

Table 3: Classification accuracy of different representation learning methods with one-step prediction.

Algorithm	Color	Size	Shape
Reconstruction-based encoder	99.78% ±0.30%	79.06% ±0.94%	41.02% ±3.45%
Inverse dynamics-based encoder	100.00% ±0.00%	100.00% ±0.00%	86.90% ±11.98%
Contrastive learning-based encoder	25.38% ±1.32%	47.52% ±26.25%	40.14% ±29.96%
Classifier-based encoder	99.40% ±0.28%	99.68% ±0.10%	99.32% ±0.46%
Our model	96.88% ±6.19%	99.82% ±0.26%	95.24% ±6.35%

Table 4: Classification accuracy of different representation learning methods with two-step prediction.

Algorithm	Color	Size	Shape
Reconstruction-based encoder	99.64% ±0.37%	47.20% ±7.26%	30.00% ±3.17%
Inverse dynamics-based encoder	99.70% ±0.14%	99.24% ±0.16%	85.60% ±11.13%
Contrastive learning-based encoder	24.90% ±0.81%	46.14% ±26.91%	40.40% ±29.75%
Classifier-based encoder	97.88% ±1.24%	99.14% ±0.81%	98.18% ±1.31%
Our model	95.10% ±9.30%	98.66% ±1.94%	92.60% ±10.35%

Table 5: Classification accuracy of different representation learning methods with three-step prediction.

Algorithm	Color	Size	Shape
Reconstruction-based encoder	97.48% ±2.96%	34.98% ±3.99%	24.80% ±1.40%
Inverse dynamics-based encoder	99.26% ±0.29%	98.12% ±0.07%	82.84% ±10.55%
Contrastive learning-based encoder	24.40% ±0.52%	45.92% ±27.01%	39.60% ±30.06%
Classifier-based encoder	95.30% ±2.90%	98.56% ±1.45%	96.98% ±2.63%
Our model	93.68% ±11.45%	97.84% ±3.13%	91.62% ±11.97%

Table 6: Classification accuracy of different representation learning methods with four-step prediction.

Algorithm	Color	Size	Shape
Reconstruction-based encoder	96.36% ±4.12%	34.72% ±2.28%	24.04% ±1.69%
Inverse dynamics-based encoder	98.62% ±0.24%	97.28% ±0.10%	81.56% ±10.15%
Contrastive learning-based encoder	24.68% ±1.70%	46.90% ±26.85%	40.00% ±30.01%
Classifier-based encoder	92.38% ±4.93%	97.32% ±2.91%	94.48% ±3.72%
Our model	91.56% ±14.60%	96.70% ±5.49%	89.08% ±16.24%

Table 7: Classification accuracy of different representation learning methods with five-step prediction.

Algorithm	Color	Size	Shape
Reconstruction-based encoder	95.24% ±5.52%	32.94% ±2.05%	24.32% ±1.39%
Inverse dynamics-based encoder	98.00% ±0.26%	96.44% ±0.35%	79.86% ±9.78%
Contrastive learning-based encoder	25.84% ±1.01%	46.32% ±26.87%	39.90% ±30.06%
Classifier-based encoder	89.46% ±6.56%	96.68% ±3.31%	93.36% ±4.76%
Our model	89.86% ±17.08%	95.66% ±6.39%	87.78% ±17.55%

F Confusion Matrix

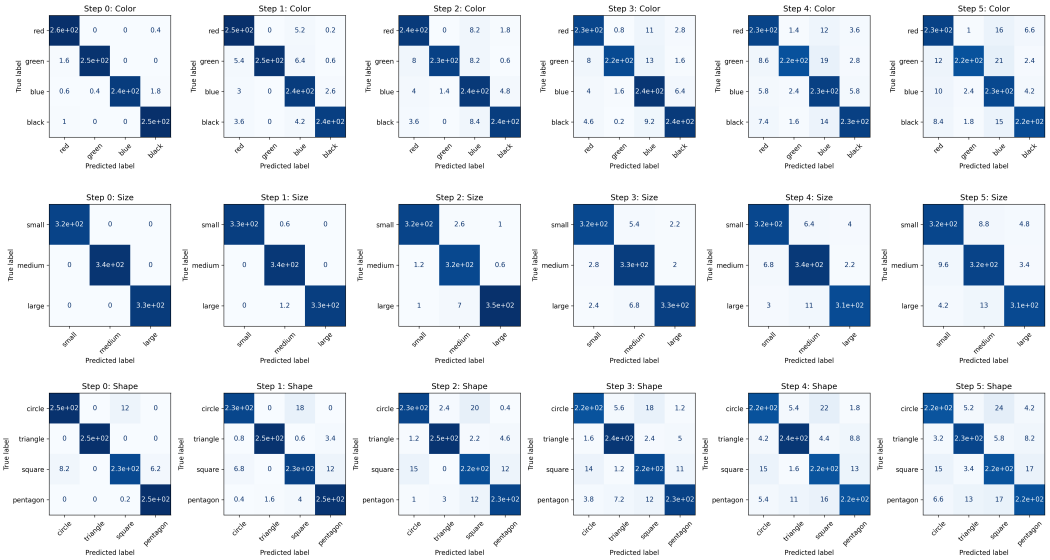

Figure 7: Confusion matrix for our model.

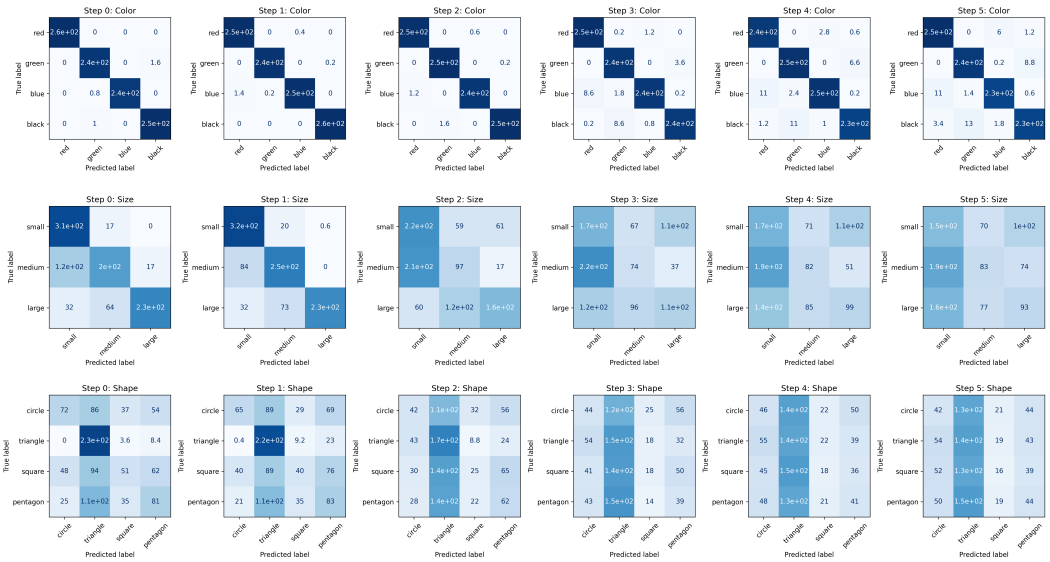

Figure 8: Confusion matrix for reconstruction-based encoder.

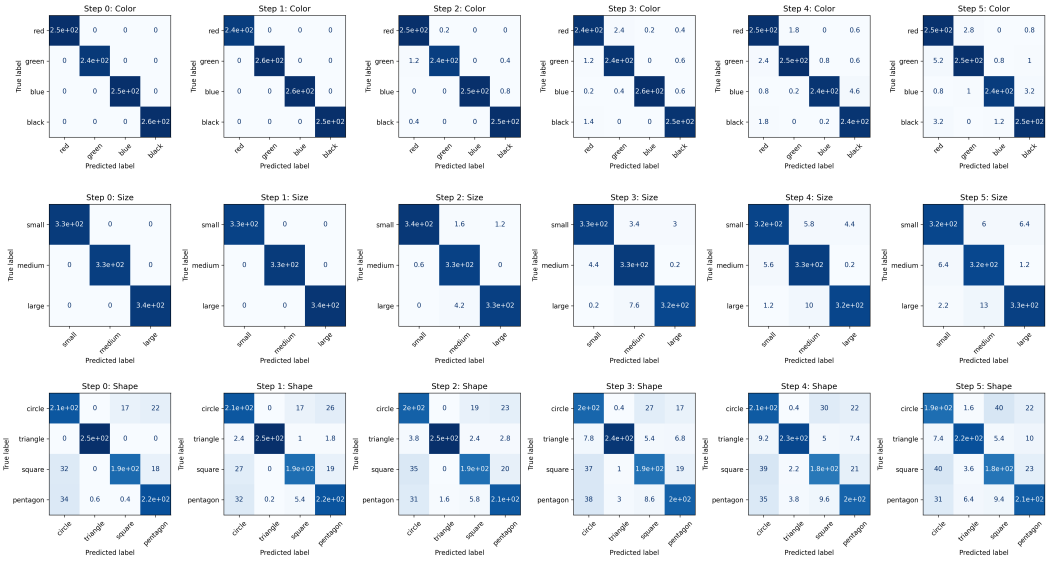

Figure 9: Confusion matrix for inverse dynamics-based encoder.

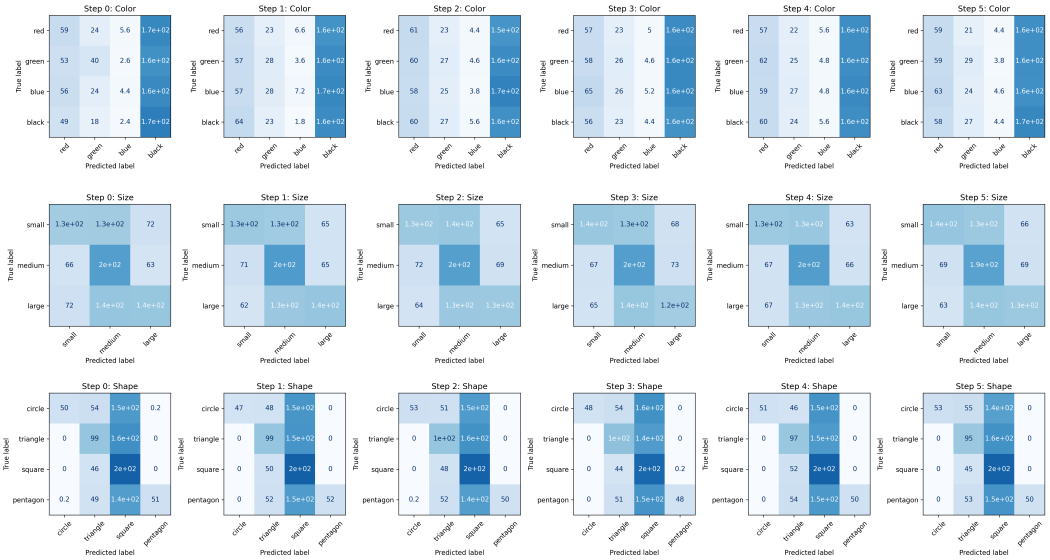

Figure 10: Confusion matrix for contrastive learning-based encoder.

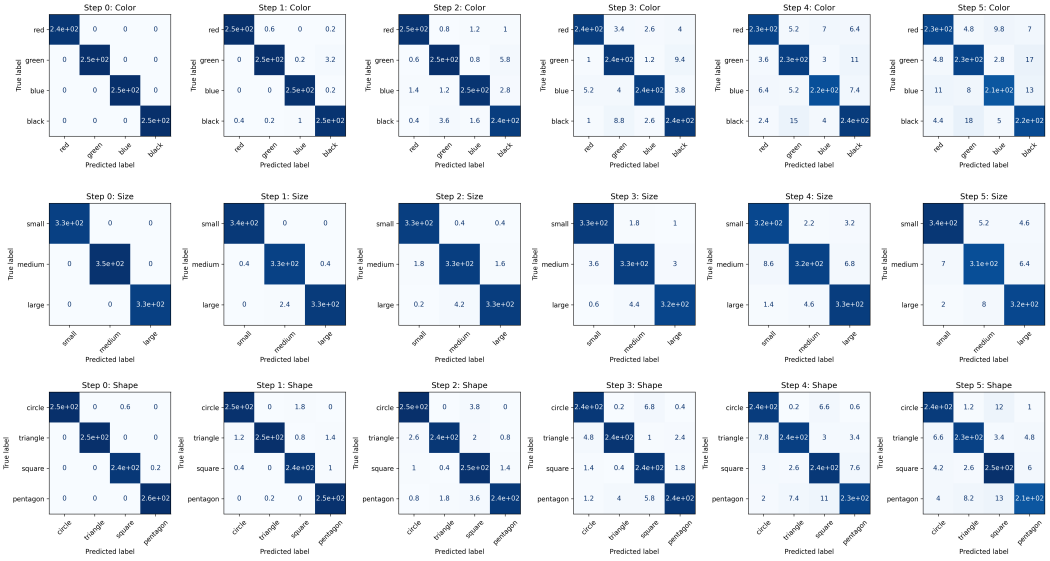

Figure 11: Confusion matrix for classifier-based encoder.