CAN WEAK QUANTIZATION MAKE WORLD MODELS PHYSICALLY INTERPRETABLE?

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning models are increasingly employed for perception, prediction, and control in autonomous systems. For achieving realistic and consistent outputs, it is crucial to embed physical knowledge into their learned representations. However, doing so is difficult due to high-dimensional observation data, such as images, particularly under conditions of incomplete system knowledge and imprecise state sensing. To address this, we propose *Physically Interpretable World Models*, a novel architecture that aligns learned latent representations with real-world physical quantities. To this end, our architecture combines a physical interpretable image autoencoding model and a partially known learnable dynamical model. We conduct an in-depth analysis of the latent space, evaluating the effects of continuous versus discrete representations, as well as intrinsic versus extrinsic physical interpretable encodings. The training incorporates weak distributional supervision to eliminate the impractical reliance on ground-truth physical knowledge. Through three case studies, we demonstrate that our approach not only provides physical interpretability but also achieves state prediction accuracy superior to state-of-the-art models, thus advancing interpretable representation learning.

1 INTRODUCTION

Accurate and robust trajectory prediction from high-dimensional sensor data, such as camera images, is a fundamental challenge for the safe operation of autonomous systems. A dominant paradigm for this task is to learn a compact latent representation of the environment and evolve it over time. This principle forms the basis of modern *world models* (Ha & Schmidhuber, 2018), which extend Variational Autoencoders (VAEs) (Kingma, 2013) by integrating predictive components like recurrent neural networks (RNNs) to capture temporal dependencies (Mao et al., 2024b; 2025b). Recent advancements in hierarchical latent modeling and transformer architectures have further enhanced the fidelity and long-horizon capabilities of these predictive models (Micheli et al., 2022; Hafner et al., 2023; Seo et al., 2023). However, while these models achieve impressive predictive performance, their learned representations often function as a "black box," lacking a clear connection to the underlying physical state of the system.

Bridging this gap would greatly improve the trustworthiness and controllability of autonomous systems operating in complex, high-risk scenarios. For instance, in autonomous driving, linking latent representations to physical states could generate causal explanations for decisions (e.g., slowing down due to occlusions) and enable high-assurance methods like formal verification (Hasan & Tahar, 2015) and run-time shielding (Waga et al., 2022). Moreover, physical states are essential for high-assurance methods such as formal verification (Katz et al., 2022) and run-time shielding (Alshiekh et al., 2018). For instance, if a predicted position lies in an occupied lane, a shield can intervene to prevent a collision. Physically interpretable representations also support causal explanations. For example, "that car won't slow down because it doesn't see you" which enhances trust in autonomous behavior. Finally, physical priors improve generalization and sample efficiency by guiding the model toward plausible trajectories.

So far, several approaches for achieving physically interpretable latent spaces from high-dimensional observations can be categorized into two fundamental paradigms: intrinsic and extrinsic encoding methods (Figure 1a). Extrinsic approaches adopt a two-stage strategy, first learning abstract latent representations from images through standard autoencoders, then mapping these representations to

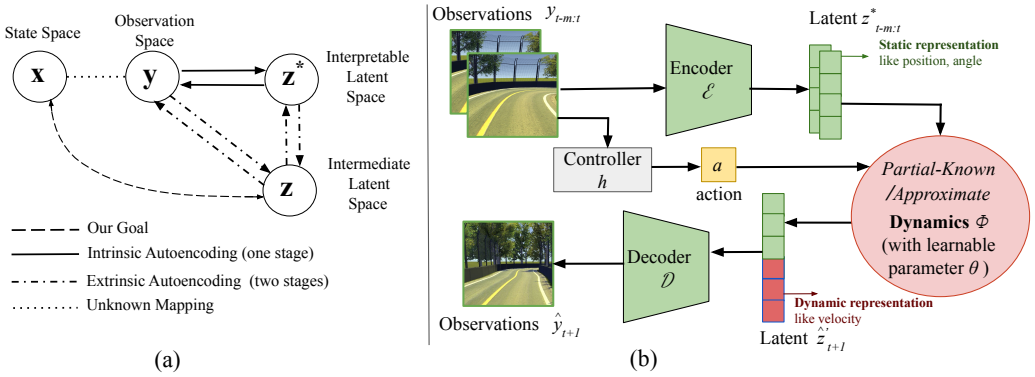

Figure 1: Overview of Physically Interpretable World Models (PIWM). (a) Two approaches to learn a physically meaningful latent space (z^*): Intrinsic and Extrinsic Autoencoding. (b) The PIWM architecture learns a structured latent representation z^* from images, uses a learnable dynamics model ϕ with physical priors to predict future latent states, and decodes them into future images.

physical quantities via additional neural network layers. While this decoupling provides flexibility, it typically requires ground-truth physical states for training the mapping function and may lose physical consistency during the intermediate representation stage. Intrinsic approaches attempt to directly encode physical structure into the image encoder itself, ensuring that the latent representations extracted from visual inputs inherently correspond to meaningful physical quantities (Le et al., 2025). However, both extrinsic and intrinsic methods face significant limitations: they require large-scale datasets with precise physical annotations, need exact supervision of physical variables during training, or are limited to scenarios where object decomposition is feasible and meaningful with object-centric representation learning method (Mosbach et al., 2025).

We aim to address the problem of learning physically meaningful representations *without exact supervision of physical labels* in the context of trajectory prediction. Instead, we rely on weak supervisory signals, which contain valuable information such as approximate speed bounds inferred from coarse or noisy position sequences. As a natural representation of such uncertain signals, we use *distributions* over the true values, which tend to come from sensing and perception pipelines in autonomous systems. In practice, many such pipelines (e.g., GPS, radar) produce probabilistic estimates, such as confidence ellipses or ranges, rather than precise values. Distributions offer a simple, expressive, and widely used abstraction for representing such uncertainty, especially in field robotics, where full state information is rarely available. Thus, we use weak-supervision distributions to steer the encoding of high-dimensional images into physically interpretable representations. This creates an opportunity for physics-informed prediction. Since the governing dynamics of many real-world systems are known or can be well-approximated, we can now make physically plausible predictions by embedding this known structure and learning only its unknown internal parameters.

Our central contribution is the *Physically Interpretable World Model* (PIWM), a novel and flexible architecture designed to learn physically meaningful representations from high-dimensional observations. The core principle of PIWM is to align its learned latent space with real-world physical quantities. It achieves this by integrating a learnable dynamics model, which incorporates partially known physical equations as structural priors, with a powerful image autoencoder. A key aspect of our approach is the use of weak distributional supervision to guide this alignment, which eliminates the impractical reliance on ground-truth physical state annotations.

Within the PIWM architecture, we conduct an in-depth experimental analysis to answer a central question: Can weak quantization make world models physically interpretable, and how to construct a more interpretable latent space? To this end, we systematically evaluate several key design choices, including continuous versus discrete latent spaces and the use of intrinsic versus extrinsic physical encodings. We validate the proposed architecture across three autonomy case studies: cart pole, lunar lander, and autonomous racing donkey car. The experiments demonstrate that the PIWM framework not only achieves superior physical grounding and lower prediction error compared to purely data-driven baselines but also provides clear insights into the optimal design of physically interpretable latent spaces. More specifically, our results reveal that extrinsic architectures, which decouple visual perception from physical interpretation, significantly outperform their end-to-end intrinsic counterparts in both predictive accuracy and the plausibility of the learned physical parameters.

2 PRELIMINARIES

2.1 AUTONOMOUS SYSTEM AND WORLD MODELS

Definition 1 (Autonomous System). *A* autonomous system $s = (X, I, A, \phi_\theta, g, Y)$ *models the evolution of an image-based control system, where the state set* X *defines the finite-dimensional space of physical states, the initial set* $I \subset X$ *specifies possible starting states, and the action set* A *contains all possible control actions. The system dynamics* $\phi_\theta : X \times A \times \Theta \to X$ *govern state transitions under physical parameters* $\theta \in \Theta$*, the observation function* $g : X \to Y$ *maps states to observations, and the controller* $h : Y \to A$ *selects actions based on observations.*

Consider an autonomous system with observation-based control, the states of which evolve as $x_{t+1} = \phi(x_t, a_t, \theta^*)$, where θ^* are the true (unknown) physical parameters. The known controller h relies on observations y from sensors (e.g., cameras). Such controllers can be trained by imitation learning (Hussein et al., 2017) or reinforcement learning (Kaelbling et al., 1996; Lillicrap, 2015). Given an initial state $x_0 \in I$, a *state trajectory* is defined by iteratively applying the dynamics function ϕ, yielding a sequence $(x_0, x_1, \ldots, x_{i+1})$, where each state is $x_{t+1} = \phi(x_t, a_t, \theta^*)$ and the action is generated by the controller based on the observation: $a_t = h(y_t) = h(g(x_t))$. Correspondingly, an *observation trajectory* $(y_0, y_1, \ldots, y_{t+1})$ is generated by applying the observation function g to each state: $y_t = g(x_t)$.

We consider the setting where the state is not directly observable, and to anticipate hazards and adapt, the system needs to forecast observations. This is done by combining a world model conditioned on past observations with an observation-based controller h, essentially forming an efficient online simulator of future observations.

Definition 2 (World Model). *A* world model $\mathcal{W} = (\mathcal{E}, f, \mathcal{D})$ *predicts the future evolution of observations in a robotic system* s*, where the encoder* $\mathcal{E} : Y \to Z$ *compresses high-dimensional observations into latent representations, the predictor* $f : Z \times A \to Z$ *forecasts future latents conditioned on actions, and the decoder* $\mathcal{D} : Z \to Y$ *reconstructs predicted observations. The actions* a_t *are generated by a controller* $h : Y \to A$ *based on the current observation* y_t*, and together with the encoded latent* $z_t = \mathcal{E}(y_t)$*, are used to obtain the future latent* $z_{t+1} = f(z_t, a_t)$*. The predicted observation* $\hat{y}_{t+1}$ *is then obtained by decoding* z_{t+1} *with* $\mathcal{D} : \hat{y}_{t+1} = \mathcal{D}(z_{t+1})$*.*

While world models enable diverse tasks (e.g., controller training), they are usually evaluated with *predictive accuracy* — how closely the predicted observations $\hat{y}_{t+1:t+n}$ match the true observations $y_{t+1:t+n}$. The typical metrics include mean squared error (MSE) and structural similarity index (SSIM). Unfortunately, these metrics are inherently limited: similar pixel-wise observations may correspond to very different underlying states. This means that the similarity (low MSE and high SSIM) between $\hat{y}_{t+1:t+n}$ and $y_{t+1:t+n}$ does not guarantee similarity between the underlying physical states corresponding to the predicted observations and the true physical ones. Thus, the error in predicted observations is not physically interpretable. This severely limits the predictions' utility for downstream tasks like safety monitoring and planning.

2.2 INTERPRETABLE REPRESENTATION LEARNING AND PREDICTION

The autoencoder components $\mathcal{E}$ and $\mathcal{D}$ in world models are primarily based on either VAE, VQ-VAE, or their variants. VAEs (Kingma, 2013) encode high-dimensional inputs y into continuous latent vectors $z = \mathcal{E}(y)$, which are then decoded back into reconstructed inputs $\hat{y} = \mathcal{D}(z)$. A standard VAE assumes a simple prior distribution over the latent space, typically an isotropic Gaussian, which facilitates tractable inference but imposes a strong inductive bias. A VQ-VAE (Van Den Oord et al., 2017) discretizes the latent space by mapping a high-dimensional observation y_t to a continuous latent vector $z_t = \mathcal{E}(y_t)$, then quantizing it via a *codebook* $\{\mathbf{e}_k\}_{k=1}^K$ to obtain a discrete latent *index* $z_t^* = \arg\min_k \|z_t - \mathbf{e}_k\|_2^2$ closest to the continuous latent z_t. The decoder then reconstructs the observation via $\hat{y}_t = \mathcal{D}(\mathbf{e}_{z_t^*})$. While discretization introduces structure and regularization, the learned codebook vectors $\mathbf{e}_k$ themselves are typically unstructured and lack interpretation. Therefore, merely discretizing the space is insufficient; hence, an explicit mechanism is needed to align these latent representations with physical semantics.

Our setting assumes that general knowledge about the dynamics function ϕ, namely the structure of its equations. However, the dynamics parameters θ, such as the mass or friction coefficient, are

unknown, making the system s only partially specified. In practice, it is often hard to find the true parameters θ^* because true physical states x cannot be measured precisely. However, it is typical to compute a distribution $p(x)$ from high-dimensional observations. For instance, the robot's pose may be estimated from images as a range/distribution — and serve as a weak supervisory signal.

Thus, to train a world model, we are given a dataset $\mathcal{S}$ of trajectories, consisting of N sequences of length M, where each sequence contains tuples of images, actions, and state distribution labels:

$$\mathcal{S} = \Big\{ \big(y_{1:M}, a_{1:M}, p_{1:M}(x)\big) \Big\}_{1:N} \tag{1}$$

The weak supervision is provided by state distributions $p_t(x)$, for which we assume no analytical form is known. Instead, we can only draw a finite set of samples from them. This represents a challenging yet practical scenario, as it is a weaker assumption than knowing the distribution's type (e.g., Gaussian), its parameters (e.g., mean and variance), or even a definitive interval.

Our ultimate task is to learn a representation z^* that accurately approximates the system's true, unobserved physical state x for prediction. We formalize this as two distinct but related problems:

Definition 3 (Interpretable Representation Learning Problem)**.** *Given a dataset $\mathcal{S}$ and a controller h, the objective is to* learn a state representation $z^* = \mathcal{E}(y)$ *that minimizes the mean squared error to the true physical state x:*

$$\min_{\mathcal{E}} \ \mathbb{E}\big[\|x - \mathcal{E}(y)\|_2^2\big] \tag{2}$$

The prediction problem is to forecast the evolution of these interpretable representations over time.

Definition 4 (Prediction Problem for Interpretable Representations)**.** *Given a dataset $\mathcal{S}$, a dynamics function ϕ with unknown parameters, and a controller h, the objective is to* train a predictor *that maps a history of representations to a future representation, $\hat{z}^*_{t+k} = \mathcal{P}(z^*_{t-m:t})$, by minimizing the future state prediction error:*

$$\min_{\mathcal{P}} \ \mathbb{E}\big[\|x_{t+k} - \hat{z}^*_{t+k}\|_2^2\big]. \tag{3}$$

The objectives in Definitions 3 and 4 are formulated with respect to the true physical state x, which serves as the ultimate ground truth for evaluating physical interpretability. However, since x is not accessible during training, these objectives cannot be optimized directly. Our proposed method, detailed in Section 3, addresses this challenge by constructing a tractable surrogate objective that leverages the weak supervision by sampling state distributions $p(x)$.

3 Architecture

We introduce the *Physically Interpretable World Model* (PIWM), a flexible prediction architecture with physically-grounded representations of high-dimensional observations. It consists of two core components: (1) a physically interpretable autoencoder responsible for learning the state representation, and (2) a learnable dynamics model that predicts the evolution of this representation.

Learning Physically Interpretable Representations. The primary goal of the autoencoder is to map a high-dimensional observation y to a low-dimensional interpretable latent state z^*. This representation must be explicitly aligned with the true physical state x. Per Section 2, we cannot access x or the analytical form of its supervisory distribution $p(x)$. Instead, we are given access to a set of L state proxy samples, $\Xi = \{\xi^{(l)}\}_{l=1}^{L}$, drawn from $p(x)$. To leverage these proxy samples Ξ for training, we formulate a general interpretability loss, $\mathcal{L}_{\text{interp}}$, which measures the discrepancy between a predicted physically interpretable state z^*_p and the sample set Ξ. In our experiments, we primarily use a Mean Squared Error (MSE) formulation that penalizes the distance to the empirical mean of the samples:

$$\mathcal{L}_{\text{interp}}(z^*_p, \Xi) = \|z^*_p - \hat{\mu}_\xi\|_2^2, \quad \text{where} \quad \hat{\mu}_\xi = \frac{1}{L}\sum_{l=1}^{L} \xi^{(l)}. \tag{4}$$

An alternative could be to use the Kullback-Leibler (KL) Divergence against a Gaussian fitted to the samples. This serves as the mechanism for enforcing physical grounding throughout our models.

A central design question in learning representation z^* is how to manage two competing objectives: reconstructing high-dimensional observations versus aligning the latent space with low-dimensional

physical states. We consider two approaches. The *Intrinsic* approach attempts to achieve both objectives simultaneously with a single, end-to-end encoder. While potentially more efficient, this forces one network to both capture fine-grained visual details (for reconstruction) and ignore those same details to extract the underlying physical state — a difficult optimization task. In contrast, the *Extrinsic* approach follows a two-stage process: a vision autoencoder first learns an intermediate representation focused on reconstruction, and then a second, physical encoder extracts the interpretable state from it. This modularity may stabilize training but risks information loss in the intermediate step. Given the fundamental trade-offs, we will systematically investigate both approaches.

Orthogonal to this architectural choice, a second key design decision is the nature of the latent space Z^*. *Continuous spaces* can represent high-fidelity physical quantities but lack a built-in organizational prior, requiring explicit regularization for disentanglement. Conversely, *discrete spaces* enforce regularity by design through their finite codebook but lose precision due to quantization error. Below, we detail the combinations of intrinsic/extrinsic and continuous/discrete approaches.

Intrinsic Autoencoding. This approach employs a single, end-to-end encoder $\mathcal{E} : Y \to Z^*$ that directly maps an observation y to the final interpretable latent state z^*. This unified representation must disentangle visual features from physical semantics, a known challenge where information may leak between different physical attributes and hinder the desired learning (Peper et al., 2025). For the case of *continuous* latent space, the encoder outputs the parameters for a posterior distribution $q(z^* \mid y)$ over the latent space Z^*. Sampled from this distribution, a latent vector is then partitioned in a fixed manner: $z^* = [z_p^*, z_v^*]$, where z_p^* is the physically interpretable part and z_v^* captures the remaining visual information necessary for reconstruction. The full objective follows the β-VAE formulation, with the interpretability loss applied only to z_p^* and KL loss applied to z_v^*:

$$\mathcal{L}_{\text{intrinsic-cont}} = \mathcal{L}_{\text{recon}}(y, \hat{y}) + \lambda_{\text{interp}}\mathcal{L}_{\text{interp}}(z_p^*, \Xi) + \beta D_{\text{KL}}\big(q(z_v^*|y) \,\|\, \mathcal{N}(0, I)\big) \tag{5}$$

For the case of *discrete* latent space, we structure the codebook to be interpretable. Each vector $\mathbf{e}_k$ is partitioned as $\mathbf{e}_k = [\mathbf{e}_k^p, \mathbf{e}_k^v]$. Only the visual part $\mathbf{e}_k^v$ is a typical learnable VQ-VAE codebook vector. The physical part $\mathbf{e}_k^p$ is a constant vector representing a specific point in a discretized grid of physical values (e.g., specific positions). Then, the interpretable state is computed as the average of the physical portions of the codebook vectors, $z_p^* = \frac{1}{|I|}\sum_{i \in I} \mathbf{e}_i^p$. The full objective combines the standard VQ loss with our interpretability loss:

$$\mathcal{L}_{\text{intrinsic-disc}} = \mathcal{L}_{\text{VQ}}(y, \hat{y}) + \lambda_{\text{interp}}\mathcal{L}_{\text{interp}}(z_p^*, \Xi), \tag{6}$$

where $\mathcal{L}_{\text{VQ}}$ includes reconstruction, codebook, and commitment losses:

$$\mathcal{L}_{\text{VQ}} = \|y - \hat{y}\|_2^2 + \big\|\text{sg}[z_{\text{cont}}] - z_q\big\|_2^2 + \beta\big\|z_{\text{cont}} - \text{sg}[z_q]\big\|_2^2 \tag{7}$$

Here, z_{cont} is the continuous output of the encoder $\mathcal{E}$, and z_q is its nearest vector from the codebook. The $\text{sg}[\cdot]$ is the stop-gradient operator, which ensures that gradients are routed correctly for the codebook loss (updating z_q) and the commitment loss (updating z_{cont}). The hyperparameter β weights this commitment loss, controlling how strongly the encoder's output is encouraged to match the chosen codebook vector.

Extrinsic Autoencoding. This approach utilizes a two-stage training process to decouple perception from interpretation. First, a general-purpose vision autoencoder $(\mathcal{E}_v, \mathcal{D}_v)$ is trained to map an observation y to an intermediate latent vector $z = \mathcal{E}_v(y)$. This stage is trained with a standard objective, independent of physical supervision. For the *continuous* case, this is a β-VAE trained to minimize:

$$\mathcal{L}_{\text{vision-cont}} = \mathcal{L}_{\text{recon}}(y, \hat{y}) + \beta D_{\text{KL}}\big(q(z \mid y) \,\|\, \mathcal{N}(0, I)\big) \tag{8}$$

For the *discrete* case, a VQ-VAE is trained to minimize the standard VQ loss $\mathcal{L}_{\text{vision-disc}} = \mathcal{L}_{\text{VQ}}(y, \hat{y})$.

After the first stage, the vision encoder $\mathcal{E}_v$ is frozen. The second stage trains a separate, auxiliary physical autoencoder $(\mathcal{E}_p, \mathcal{D}_p)$ to map the intermediate representation $z = \mathcal{E}_v(y)$ to a final, purely physical representation $z^* = \mathcal{E}_p(z)$. The training objective for this stage is:

$$\mathcal{L}_{\text{physical}} = \lambda_{\text{interp}}\mathcal{L}_{\text{interp}}(z^*, \Xi) + \lambda_{\text{latent}}\mathcal{L}_{\text{recon}}\big(z, \mathcal{D}_p(z^*)\big) \tag{9}$$

This same loss $\mathcal{L}_{\text{physical}}$ is used for both continuous and discrete intermediate representation z.

Learnable Dynamics Model. The second core component of our PIWM is a latent dynamics model ϕ that predicts the temporal evolution of the physically interpretable state z^*. Rather than using

black-box sequence models, our prediction is based on known dynamics equations, $\phi(z_t^*, a_t, \theta)$, where the form of ϕ (e.g., kinematics) is fixed and only its parameters θ are learnable. This allows our model to reflect the underlying physical laws while adapting to unknown system parameters θ such as friction and mass.

The dynamics model is responsible for estimating physical quantities that depend on a history of states, such as velocity, in order to predict the system's evolution. To do this, the model is initialized with a short window of consecutive representations, (z_t^*, z_{t+1}^*), produced by the encoder from observations. These two states, along with the control action a_{t+1} that causes the transition from state $t+1$ to $t+2$, are used by the learnable dynamics model ϕ to predict the next state: $\hat{z}_{t+2}^* = \phi(z_t^*, z_{t+1}^*, a_{t+1}, \theta)$. By taking two consecutive states as input, the model can internally compute velocity and other time-derivative quantities necessary for an accurate physical prediction. After this initialization phase, the model can operate recursively for multi-step rollouts, taking its own prediction from the previous step as input to generate a future trajectory. This enables efficient, long-horizon forecasting without needing a sequence of observations at every step.

The learnable parameters θ of the dynamics model ϕ are trained by minimizing a dynamics loss, $\mathcal{L}_{\text{dyn}}$. This objective ensures that the predicted state $\hat{z}_{t+k}^*$, generated by recursively applying the dynamics model $\phi(\cdot, \cdot, \theta)$, aligns with the weak supervision available for that future time step. We use a Mean Squared Error (MSE) loss, which compares the prediction against the empirical mean of the proxy labels Ξ_{t+k}. The loss is defined as a function of the parameters θ:

$$\mathcal{L}_{\text{dyn}}(\theta) = \|\hat{z}_{t+k}^* - \hat{\mu}_{\xi_{t+k}}\|_2^2, \tag{10}$$

where $\hat{z}_{t+k}^*$ is the state predicted by the dynamics model parameterized by θ, and $\hat{\mu}_{\xi_{t+k}} = \frac{1}{L}\sum_{l=1}^{L} \xi_{t+k}^{(l)}$ is the empirical mean of the L proxy samples for the future state. The full training algorithm, including gradient backpropagation through the dynamics, can be found in the Appendix.

4 Experimental Evaluation

To validate our PIWM approach, we experiment on three environments: CartPole, Lunar Lander, and the DonkeyCar autonomous racing platform (Brockman et al., 2016; Viitala et al., 2021). These environments differ in the observation dimensionality, action space, and underlying dynamics.

Experimental Setup. For each environment, we collect a dataset of 60,000 trajectories, each with at least 50 time steps. To ensure diverse state space coverage, trajectories are generated by executing both random actions and those generated by well-trained neural controllers. The weak supervision signals are generated by perturbing the ground-truth physical states for different noise levels $\delta \in \{0, 5\%, 10\%\}$. For each, the weak supervision given by a uniform distribution over an interval constructed as follows. Its width is equal to the δ fraction of the full range of the respective state dimension. The interval's center is randomly shifted from the ground truth by an amount drawn from a uniform distribution over $[-\delta/2, \delta/2]$ of the full state range.

We evaluate our PIWM variants (Intrinsic/Extrinsic, Continuous/Discrete) against a suite of strong baselines under 5-fold cross-validation. We first include data-driven sequence models, an LSTM and a Transformer, which serve as non-physical benchmarks. Our primary comparisons are to state-of-the-art models in two categories. For the intrinsic approach, we compare against Vid2Para (Asenov et al., 2019) and GokuNet (Linial et al., 2021). For the extrinsic approach, we evaluate against DVBF (Karl et al., 2016). To specifically isolate and compare the performance of the latent dynamics predictors, we also include SindyC (Brunton et al., 2016) — a classic state-based method for dynamics discovery. For a fair comparison, both the DVBF and SindyC dynamics models operate on the interpretable latent representations produced by our continuous autoencoder. All models are configured to have a comparable parameter count to focus the comparison on architectural efficacy rather than model capacity. Further details and the results of CartPole can be found in the Appendix.

Predictive Performance. We first evaluate the primary task of long-horizon trajectory prediction. Figure 2 shows the root mean square error (RMSE) for 30-step future state prediction in the challenging Donkey Car environment, comparing both extrinsic and intrinsic methods across different levels of supervision noise, δ. The results for extrinsic methods (Fig. 2) show a clear performance hierarchy. Our PIWM variants consistently outperform all baselines. The quantized extrinsic model (purple line) achieves the lowest and most stable prediction error, maintaining accuracy even as the

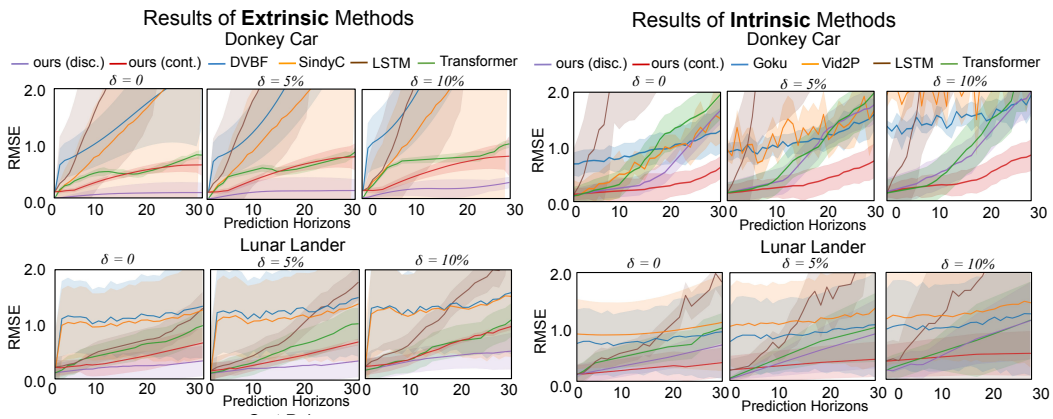

Figure 2: Prediction performance. The Root Mean Square Error (RMSE) of our PIWM variants (extrinsic methods, left; intrinsic methods, right) is compared against baselines over a 30-step prediction horizon in the Donkey Car and Lunar Lander across varying levels of weak supervision (δ).

prediction horizon extends. Our continuous extrinsic model (red line) is the second-best performer. Both significantly surpass the extrinsic baselines (DVBF, SindyC) and the purely data-driven models (LSTM, Transformer), whose errors escalate rapidly.

A more nuanced picture emerges for the intrinsic methods (Fig. 2, right). Our PIWM models again demonstrate a clear advantage over the Vid2Para and GokuNet baselines. Strikingly, our continuous intrinsic model (red line) achieves a predictive accuracy that is highly competitive with, and in some cases even surpasses, our top-performing extrinsic models. This suggests that a well-regularized, end-to-end continuous architecture can be highly effective. In contrast, the quantized intrinsic model (purple line) exhibits less stability and higher error in this configuration, indicating that the optimization challenge of aligning a discrete codebook within a single, unified encoder is considerable. Nevertheless, both of our intrinsic variants outperform the baseline models, confirming the overall benefit of our training methodology. A key insight from these results is that decoupling visual perception from physical state inference (the extrinsic approach) is a critical design choice for achieving robust, long-term prediction. Furthermore, across both architectures, the quantized (discrete) latent space provides a powerful regularization effect, leading to more stable predictions than the continuous alternative, especially under noisy supervision.

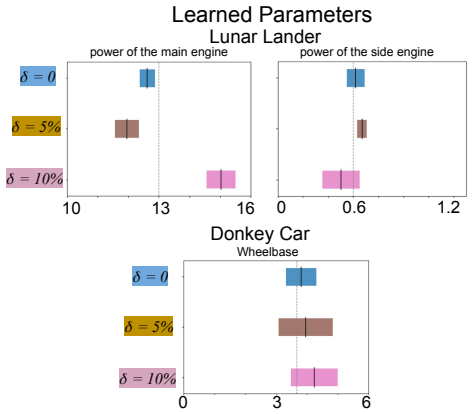

Figure 3: Learned Physical Parameters vs. Ground Truth. The parameters learned by our model (colored bars) are compared against the ground-truth values (dashed lines) under varying noise levels (δ).

Quality of the Learned Physical Representation. Strong predictive performance should stem from an accurate underlying state representation. We validate this by evaluating the static encoding quality and the model's ability to recover the true physical parameters of the simulation.

Table 1 presents the static encoding RMSE, measuring how accurately each model can infer the physical state from a single observation. The results strongly correlate with the predictive findings. Our quantized extrinsic PIWM achieves the highest encoding accuracy, confirming that its superior representation is the foundation of its predictive power. The intrinsic continuous models, while competitive, exhibit higher encoding error, consistent with their weaker predictive performance.

Finally, we assess whether the dynamics model can learn the true physical parameters (e.g., car length, pole mass) within the learned latent space. As reported in Figure 3, PIWM successfully recovers the ground-truth parameters with low relative error across all environments. This provides

direct evidence that our framework learns a genuinely interpretable representation that is not merely correlated with the physics but is structured in a way that is consistent with them.

Table 1: Static Encoding RMSE (mean $\pm$ std) Comparison of MSE and KL Losses

Case	Model	$\delta = 0\%$		$\delta = 5\%$		$\delta = 10\%$	
		MSE	**KL**	**MSE**	**KL**	**MSE**	**KL**
Donkey Car	Intrinsic (Cont.)	0.13 ± 0.05	0.45 ± 0.06	0.16 ± 0.06	0.52 ± 0.07	0.20 ± 0.07	0.58 ± 0.08
	Intrinsic (Disc.)	0.18 ± 0.04	0.42 ± 0.05	0.21 ± 0.05	0.55 ± 0.06	0.22 ± 0.06	0.64 ± 0.07
	Extrinsic (Cont.)	0.11 ± 0.05	0.24 ± 0.04	0.15 ± 0.04	0.28 ± 0.05	0.16 ± 0.05	0.33 ± 0.06
	Extrinsic (Disc.)	0.03 ± 0.02	0.19 ± 0.03	0.05 ± 0.03	0.23 ± 0.04	0.06 ± 0.04	0.28 ± 0.05
Lunar Lander	Intrinsic (Cont.)	0.07 ± 0.05	0.56 ± 0.26	0.08 ± 0.15	0.58 ± 0.37	0.21 ± 0.07	0.77 ± 0.38
	Intrinsic (Disc.)	0.06 ± 0.04	0.44 ± 0.11	0.03 ± 0.05	0.53 ± 0.12	0.16 ± 0.06	0.64 ± 0.17
	Extrinsic (Cont.)	0.11 ± 0.07	0.33 ± 0.14	0.14 ± 0.04	0.39 ± 0.16	0.15 ± 0.05	0.45 ± 0.26
	Extrinsic (Disc.)	0.03 ± 0.02	0.24 ± 0.14	0.09 ± 0.03	0.29 ± 0.11	0.12 ± 0.04	0.35 ± 0.23

Analysis and Discussion. Our experiment demonstrates that the extrinsic architecture with a discrete latent space is optimal for learning physically interpretable world models from weak supervision. This approach achieves superior prediction accuracy (Figure 2) by decoupling perception from physical state abstraction and leveraging quantization as a powerful regularizer against visual noise. The learned representations are not only predictive but also genuinely physically grounded, as evidenced by the model's ability to recover true system parameters (Figure 3) and generate qualitatively plausible visual rollouts that far exceed baseline performance (Figure 4). Crucially, the substantial gains in interpretability and prediction accuracy come at a minimal cost to downstream controller performance (Table 2, Appendix), validating our architecture as a robust and practical solution.

While our framework is effective, future work should focus on scaling its representational capacity and enhancing its use of supervision. For complex, open-world scenarios like autonomous driving, our approach could be extended from predicting simple state vectors to building structured world representations, such as dynamic 3D occupancy grids, where physical priors can be applied to multiple agents. Furthermore, the rich temporal nature of weak supervision signals is currently underutilized. Future methods could process sequences of noisy supervisory signals using filtering or sequence modeling techniques to produce a more refined, temporally coherent learning target, thereby improving the model's robustness and accuracy.

5 Related Work

Trajectory Prediction. Predicting trajectories is critical for safe planning and control (Fridovich-Keil et al., 2020), but existing methods present trade-offs. While approaches like Hamilton-Jacobi (HJ) reachability offer formal guarantees (Li et al., 2021; Nakamura & Bansal, 2023), they are computationally expensive for online settings. Conversely, mainstream deep learning models are powerful but often rely on handcrafted scene representations (Salzmann et al., 2020) or high-precision maps (Itkina & Kochenderfer, 2023; Hsu et al., 2023), and typically do not produce physically interpretable predictions (Lu et al., 2024; Lindemann et al., 2023; Ruchkin et al., 2022). In contrast, our approach learns directly from raw images with distribution-based weak supervision without requiring handcrafted inputs or goal conditioning.

Representation Learning. A central challenge in learning from high-dimensional sequences is creating compact and meaningful latent representations (Shi et al., 2015; Bai et al., 2018). While Variational Autoencoders (VAEs) (Kingma, 2013) are foundational, their standard form learns unstructured latents. A significant line of research attempts to impose structure, either by encouraging disentanglement in continuous spaces with methods like β-VAE, FactorVAE, and TCVAE (Higgins et al., 2017; Kim & Mnih, 2018; Chen et al., 2018), or by enforcing causality (Yang et al., 2021). However, these methods often fall short of ensuring a direct correspondence with real-world physical states or require precisely labeled data. An alternative is to impose structure via discretization with Vector-Quantized VAEs (VQ-VAEs) and their extensions (Van Den Oord et al., 2017; Razavi et al., 2019; Xue et al., 2019). While effective, their application to physical prediction has been limited due to abstract latent codes and reliance on exact supervision. Even in robotic applications like DVQ-VAE that model structured systems, encoding external environmental factors remains a challenge (Zhao et al., 2024). GOKU-net constrains variables to plausible physical ranges but does not tie them to specific, interpretable quantities (Linial et al., 2021).

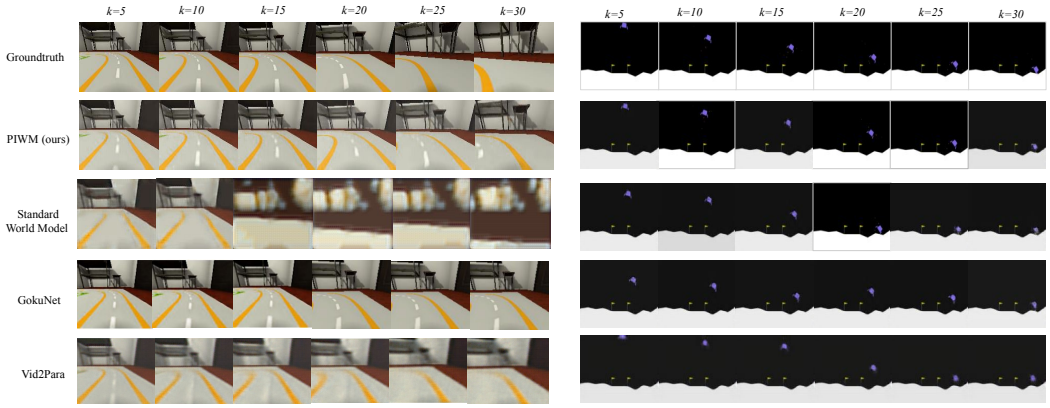

Figure 4: Qualitative Comparison of 30-Step Trajectory Rollouts under $\delta = 5\%$.

Other works incorporate structured priors to enhance learning. Object-centric world models like FOCUS and foundation world models improve efficiency by structuring latents around discrete entities (Ferraro et al., 2023; Mao et al., 2024a). Models like SPARTAN provide outputs that are interpretable by construction but do not incorporate known physical dynamics or action-conditioned prediction (Lei et al., 2024). Priors can also be introduced as task-related rules in Bayesian neural networks (Sam et al., 2024) or through human and self-supervision to guide abstraction (Fu et al., 2021; Wen et al., 2023; Konidaris et al., 2018; Peng et al., 2024; Chen et al., 2022). These approaches, however, often yield abstract representations that lack physical grounding without strong external priors. Closer to our work, Deep Variational Bayes Filters (DVBFs) extend VAEs with a latent dynamics module (Karl et al., 2016). Yet, without explicit physical supervision, they often fail to recover interpretable variables—a limitation our work addresses by leveraging weak supervision.

World Models. Classical world models like Dreamer (Hafner et al., 2020) and DayDreamer (Wu et al., 2023) excel at learning from experience for policy learning, but their latent representations are typically uninterpretable (Peper et al., 2025). Many approaches seek to improve physical grounding by incorporating priors, such as bounds on states and actions (Tumu et al., 2023; Sridhar et al., 2023), physics-aware loss functions (Djeumou et al., 2023), or kinematics-inspired layers (Cui et al., 2020). However, these methods are often designed for low-dimensional systems and do not scale well to learning from noisy, high-dimensional images. Other physics-informed methods leverage differential equations to stabilize learning (Zhong & Meidani, 2023; Linial et al., 2021), with frameworks like Phy-Taylor using Taylor monomials to structure the latent dynamics (Mao et al., 2025a). Approaches like sparse identification and differentiable physics require access to the underlying state variables and are not designed to learn from raw visual inputs (Yao et al., 2024; Brunton et al., 2016; de Avila Belbute-Peres et al., 2018).

Recent advances have produced powerful but distinct world models. For instance, 3D occupancy-based models improve forecasting but their internal states are not explicitly aligned with physical variables (Zheng et al., 2024; Min et al., 2023; Yan et al., 2024; Zuo et al., 2024). Concurrently, neuro-symbolic models enhance generalization but require predefined symbolic inputs not available from raw sensor data (Balloch et al., 2023; Liang et al., 2024). Our approach is distinct from these paradigms as it learns physically interpretable representations directly from images using weak supervision. This also contrasts with the most closely related work, Vid2Param (Asenov et al., 2019), which requires full supervision and struggles with dynamics prediction.

6 Conclusion and Broader Impact

We presented the Physically Interpretable World Model, a framework that learns physically-grounded latent representations from images using only weak distributional supervision. Our systematic evaluation demonstrates that an extrinsic architecture with a discrete latent space yields accurate and robust predictions and successfully recovers the system's true physical parameters. This work not only provides direct evidence of a genuinely interpretable model but also offers a practical path toward more trustworthy and reliable autonomous systems in high-stakes applications.

Ethics Statement

This work does not involve human subjects, sensitive personal data, or potentially harmful applications. No ethical concerns regarding discrimination, bias, privacy, or security arise in the scope of this research. All experiments were conducted using publicly available datasets under appropriate licenses, and all methods follow the ICLR Code of Ethics.

Reproducibility Statement

To ensure reproducibility, we provide implementation details in the supplementary materials, including code. The main paper describes the model architecture and experimental settings, while additional training details are documented in the appendix. Anonymous source code is included as supplementary materials. With the provided resources.

Use of Large Language Models (LLMs)

Large language models were only used for language polishing and code debugging. They did not contribute to research ideation, experimental design, or the writing of scientific content.

References

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Martin Asenov, Michael Burke, Daniel Angelov, Todor Davchev, Kartic Subr, and Subramanian Ramamoorthy. Vid2param: Modeling of dynamics parameters from video. *IEEE Robotics and Automation Letters*, 5(2):414–421, 2019.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Jonathan Balloch, Zhiyu Lin, Robert Wright, Xiangyu Peng, Mustafa Hussain, Aarun Srinivas, Julia Kim, and Mark O Riedl. Neuro-symbolic world models for adapting to open world novelty. *arXiv preprint arXiv:2301.06294*, 2023.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Sparse identification of nonlinear dynamics with control (sindyc). *IFAC-PapersOnLine*, 49(18):710–715, 2016. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2016.10.249. URL https://www.sciencedirect.com/science/article/pii/S2405896316318298. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016.

Boyuan Chen, Kuang Huang, Sunand Raghupathi, Ishaan Chandratreya, Qiang Du, and Hod Lipson. Automated discovery of fundamental variables hidden in experimental data. *Nature Computational Science*, 2(7):433–442, 2022.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Jeff Schneider, David Bradley, and Nemanja Djuric. Deep kinematic models for kinematically feasible vehicle trajectory predictions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10563–10569. IEEE, 2020.

Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in neural information processing systems*, 31, 2018.

Franck Djeumou, Cyrus Neary, and Ufuk Topcu. How to learn and generalize from three minutes of data: Physics-constrained and uncertainty-aware neural stochastic differential equations. *arXiv preprint arXiv:2306.06335*, 2023.

Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: Object-centric world models for robotics manipulation. *arXiv preprint arXiv:2307.02427*, 2023.

David Fridovich-Keil, Andrea Bajcsy, Jaime F Fisac, Sylvia L Herbert, Steven Wang, Anca D Dragan, and Claire J Tomlin. Confidence-aware motion prediction for real-time collision avoidance1. *The International Journal of Robotics Research*, 39(2-3):250–265, 2020.

Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pp. 3480–3491. PMLR, 2021.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

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

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Osman Hasan and Sofiene Tahar. Formal verification methods. In *Encyclopedia of Information Science and Technology, Third Edition*, pp. 7162–7170. IGI global, 2015.

Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

Kai-Chieh Hsu, Karen Leung, Yuxiao Chen, Jaime F Fisac, and Marco Pavone. Interpretable trajectory prediction for autonomous vehicles via counterfactual responsibility. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5918–5925. IEEE, 2023.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Masha Itkina and Mykel Kochenderfer. Interpretable self-aware neural networks for robust trajectory prediction. In *Conference on Robot Learning*, pp. 606–617. PMLR, 2023.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick Van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.

Sydney M Katz, Anthony L Corso, Christopher A Strong, and Mykel J Kochenderfer. Verification of image-based neural network controllers using generative models. *Journal of Aerospace Information Systems*, 19(9):574–584, 2022.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International conference on machine learning*, pp. 2649–2658. PMLR, 2018.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018.

Long Le, Ryan Lucas, Chen Wang, Chuhao Chen, Dinesh Jayaraman, Eric Eaton, and Lingjie Liu. Pixie: Fast and generalizable supervised learning of 3d physics from pixels. *arXiv preprint arXiv:2508.17437*, 2025.

Anson Lei, Bernhard Schölkopf, and Ingmar Posner. Spartan: A sparse transformer learning local causation. *arXiv preprint arXiv:2411.06890*, 2024.

Anjian Li, Liting Sun, Wei Zhan, Masayoshi Tomizuka, and Mo Chen. Prediction-based reachability for collision avoidance in autonomous driving. In *2021 Intl. Conf. on Robotics and Automation (ICRA)*, 2021.

Yichao Liang, Nishanth Kumar, Hao Tang, Adrian Weller, Joshua B Tenenbaum, Tom Silver, João F Henriques, and Kevin Ellis. Visualpredicator: Learning abstract world models with neuro-symbolic predicates for robot planning. *arXiv preprint arXiv:2410.23156*, 2024.

TP Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lars Lindemann, Xin Qin, Jyotirmoy V Deshmukh, and George J Pappas. Conformal prediction for stl runtime verification. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, pp. 142–153, 2023.

Ori Linial, Neta Ravid, Danny Eytan, and Uri Shalit. Generative ODE modeling with known unknowns. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, pp. 79–94, New York, NY, USA, April 2021. Association for Computing Machinery. ISBN 978-1-4503-8359-2. doi: 10.1145/3450439.3451866. URL https://dl.acm.org/doi/10.1145/3450439.3451866.

Juanwu Lu, Wei Zhan, Masayoshi Tomizuka, and Yeping Hu. Towards generalizable and interpretable motion prediction: A deep variational bayes approach. In *International Conference on Artificial Intelligence and Statistics*, pp. 4717–4725. PMLR, 2024.

Yanbing Mao, Yuliang Gu, Lui Sha, Huajie Shao, Qixin Wang, and Tarek Abdelzaher. Phy-Taylor: Partially Physics-Knowledge-Enhanced Deep Neural Networks via NN Editing. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):447–461, January 2025a. ISSN 2162-2388. doi: 10.1109/TNNLS.2023.3325432. URL https://ieeexplore.ieee.org/document/10297119.

Zhenjiang Mao, Siqi Dai, Yuang Geng, and Ivan Ruchkin. Zero-shot safety prediction for autonomous robots with foundation world models. *arXiv preprint arXiv:2404.00462*, 2024a.

Zhenjiang Mao, Carson Sobolewski, and Ivan Ruchkin. How safe am i given what i see? calibrated prediction of safety chances for image-controlled autonomy. In *6th Annual Learning for Dynamics & Control Conference*, pp. 1370–1387. PMLR, 2024b.

Zhenjiang Mao, Mrinall Eashaan Umasudhan, and Ivan Ruchkin. How safe will i be given what i saw? calibrated prediction of safety chances for image-controlled autonomy. *arXiv preprint arXiv:2508.09346*, 2025b.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Chen Min, Dawei Zhao, Liang Xiao, Yiming Nie, and Bin Dai. Uniworld: Autonomous driving pre-training via world models. *arXiv preprint arXiv:2308.07234*, 2023.

Malte Mosbach, Jan Niklas Ewertz, Angel Villar-Corrales, and Sven Behnke. Sold: Slot object-centric latent dynamics models for relational manipulation learning from pixels, 2025. URL https://arxiv.org/abs/2410.08822.

Kensuke Nakamura and Somil Bansal. Online update of safety assurances using confidence-based predictions. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12765–12771. IEEE, 2023.

Andi Peng, Mycal Tucker, Eoin Kenny, Noga Zaslavsky, Pulkit Agrawal, and Julie A Shah. Human-guided complexity-controlled abstractions. *Advances in Neural Information Processing Systems*, 36, 2024.

Jordan Peper, Zhenjiang Mao, Yuang Geng, Siyuan Pan, and Ivan Ruchkin. Four principles for physically interpretable world models. *arXiv preprint arXiv:2503.02143*, 2025.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Ivan Ruchkin, Matthew Cleaveland, Radoslav Ivanov, Pengyuan Lu, Taylor Carpenter, Oleg Sokolsky, and Insup Lee. Confidence composition for monitors of verification assumptions. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 1–12. IEEE, 2022.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pp. 683–700. Springer, 2020.

Dylan Sam, Rattana Pukdee, Daniel P Jeong, Yewon Byun, and J Zico Kolter. Bayesian neural networks with domain knowledge priors. *arXiv preprint arXiv:2402.13410*, 2024.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pp. 1332–1344. PMLR, 2023.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Kaustubh Sridhar, Souradeep Dutta, James Weimer, and Insup Lee. Guaranteed conformance of neurosymbolic models to natural constraints. In *Learning for Dynamics and Control Conference*, pp. 76–89. PMLR, 2023.

Renukanandan Tumu, Lars Lindemann, Truong Nghiem, and Rahul Mangharam. Physics constrained motion prediction with uncertainty quantification. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–8. IEEE, 2023.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ari Viitala, Rinu Boney, Yi Zhao, Alexander Ilin, and Juho Kannala. Learning to drive (l2d) as a low-cost benchmark for real-world reinforcement learning. In *2021 20th International Conference on Advanced Robotics (ICAR)*, pp. 275–281. IEEE, 2021.

Masaki Waga, Ezequiel Castellano, Sasinee Pruekprasert, Stefan Klikovits, Toru Takisaka, and Ichiro Hasuo. Dynamic shielding for reinforcement learning in black-box environments. In *International Symposium on Automated Technology for Verification and Analysis*, pp. 25–41. Springer, 2022.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pp. 2226–2240. PMLR, 2023.

Yifan Xue, Michael Ding, and Xinghua Lu. Supervised vector quantized variational autoencoder for learning interpretable global representations. *arXiv preprint arXiv:1909.11124*, 2019.

Ziyang Yan, Wenzhen Dong, Yihua Shao, Yuhang Lu, Liu Haiyang, Jingwen Liu, Haozhe Wang, Zhe Wang, Yan Wang, Fabio Remondino, et al. Renderworld: World model with self-supervised 3d label. *arXiv preprint arXiv:2409.11356*, 2024.

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9593–9602, 2021.

Dingling Yao, Caroline Muller, and Francesco Locatello. Marrying causal representation learning with dynamical systems for science. *Advances in Neural Information Processing Systems*, 37: 71705–71736, 2024.

Zhe Zhao, Mengshi Qi, and Huadong Ma. Decomposed vector-quantized variational autoencoder for human grasp generation. In *European Conference on Computer Vision*, pp. 447–463. Springer, 2024.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024.

Weiheng Zhong and Hadi Meidani. Pi-vae: Physics-informed variational auto-encoder for stochastic differential equations. *Computer Methods in Applied Mechanics and Engineering*, 403:115664, 2023. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115664. URL https://www.sciencedirect.com/science/article/pii/S0045782522006193.

Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Gaussianworld: Gaussian world model for streaming 3d occupancy prediction. *arXiv preprint arXiv:2412.10373*, 2024.

APPENDIX

A. TRAINING ALGORITHMS

The training procedures for the ablation and proposed models are summarized in Algorithm 2 and Algorithm 1, respectively.

For the PIWM model (Algorithm 1), observations are first encoded into discrete latent representations using a Vision VQ-VAE module. These discrete latents are further mapped into interpretable physical states via a physical encoder $\mathcal{E}_p$. The dynamics model ϕ_θ predicts the future physical state based on the current and next physical states. The predicted physical state is decoded back into the latent space and finally reconstructed into the future observation. The loss is computed similarly, with both reconstruction and weak supervision terms.

For the ablation model (Algorithm 2), consecutive observations are first encoded using the encoder $\mathcal{E}$ into latent variables. The dynamics model ϕ_θ predicts the future latent state by taking as input the latent representations from two consecutive frames. The decoder $\mathcal{D}$ reconstructs the observation at the future time step, and a reconstruction loss is computed. In addition, a weak supervision loss is applied to regularize the encoded latents based on interval labels.

The main difference between the ablation and proposed models lies in the latent structure and dynamics modeling: the ablation model operates directly in the visual latent space, while the proposed model enforces a structured, interpretable physical latent space for dynamics prediction.

We further include additional data-driven ablation variants, where the dynamics predictor ϕ_θ in the ablation model is replaced by a sequential model, such as an LSTM or Transformer network. These models are trained to predict the next latent state given the previous two latent representations. The rest of the architecture (encoder, decoder, and loss computation) remains identical to the basic ablation setup.

The inference procedure is similar across all variants, using a sequence of encoded observations to roll out future predictions recursively over multiple horizons.

B. BACKPROPAGATION THROUGH DYNAMICS

To optimize the parameters θ in the structured dynamics model $\phi(\cdot;\theta)$, we compute gradients of the loss using the chain rule. The loss may depend on the predicted interpretable state $\hat{x}_{t+2}$ (e.g., prediction error or safety violation), or on the sequence $\{x'_t, x'_{t+1}, x'_{t+2}\}$ for temporal consistency. We compute the gradient as:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \hat{x}_{t+2}} \cdot \frac{\partial \hat{x}_{t+2}}{\partial \theta}.$$

Since $\hat{x}_{t+2}$ is computed as $\phi(x'_{t+1}, a_{t+1}; \theta)$, which in turn depends on previous states through recursive application of ϕ, we expand the gradient recursively:

$$\frac{\partial \hat{x}_{t+2}}{\partial \theta} = \frac{\partial \phi(x'_{t+1}, a_{t+1}; \theta)}{\partial \theta} + \frac{\partial \phi(x'_{t+1}, a_{t+1}; \theta)}{\partial x'_{t+1}} \cdot \frac{\partial x'_{t+1}}{\partial \theta}.$$

This recursion can be unrolled through multiple time steps, enabling gradients to propagate through interpretable state sequences and support end-to-end learning of both state representations and dynamics parameters.

C. EXPERIMENT SETUP

We evaluate our approach on three control benchmarks of increasing complexity: CartPole, Lunar Lander, and Donkey Car. These tasks differ in state dimensionality, control difficulty, and underlying dynamics. CartPole requires balancing a pole on a moving cart using discrete left/right forces. The system has a four-dimensional state space (position, velocity, angle, angular velocity) and a binary discrete action space. It includes four dynamics parameters: cart mass, pole mass, pole length, and applied force magnitude. Lunar Lander involves landing a spacecraft on flat terrain using main and side engines. It has an eight-dimensional state space (positions, velocities, angle,

Algorithm 1: Training PIWMs

Input: Training set $\mathcal{S} = \{(y_{1:M}, a_{1:M}, [x^{\text{lo}}_{1:M}, x^{\text{up}}_{1:M}])\}_{i=1}^{N}$, batch size B, weight λ, optimizer `Adam()`

Output: Trained parameters w for $(\mathcal{E}, \mathcal{E}_p, \phi_\theta, \mathcal{D}_p, \mathcal{D})$

Initialize w randomly;
for $i \leftarrow 1$ **to** $|\mathcal{S}|/B$ **do**
  Sample a batch of B sequences from $\mathcal{S}$;
  foreach *sequence* $(y_{1:M}, a_{1:M}, [x^{lo}_{1:M}, x^{up}_{1:M}])$ *in batch* **do**
    for $t \leftarrow 1$ **to** $M-2$ **do**
      $z_t \leftarrow \mathcal{E}(y_t), \quad z_{t+1} \leftarrow \mathcal{E}(y_{t+1})$;
      $z^*_t \leftarrow \text{Quantize}(z_t), \quad z^*_{t+1} \leftarrow \text{Quantize}(z_{t+1})$;
      $\hat{x}'_t \leftarrow \mathcal{E}_p(z^*_t), \quad \hat{x}'_{t+1} \leftarrow \mathcal{E}_p(z^*_{t+1})$;
      $a_t \leftarrow h(y_t)$;
      $\hat{x}_{t+2} \leftarrow \phi_\theta(\hat{x}'_t, \hat{x}'_{t+1}, a_t)$;
      $\hat{z}^*_{t+2} \leftarrow \mathcal{D}_p(\hat{x}_{t+2})$;
      $\hat{y}_{t+2} \leftarrow \mathcal{D}(\mathbf{e}_{\hat{z}^*_{t+2}})$;
      Sample $\tilde{x}_t \sim \mathcal{U}(x^{\text{lo}}_t, x^{\text{up}}_t)$; ; `// Weak supervision sample`
      Sample $\tilde{x}_{t+2} \sim \mathcal{U}(x^{\text{lo}}_{t+2}, x^{\text{up}}_{t+2})$;
      Compute reconstruction loss: $\mathcal{L}_{\text{rec}} = \|y_{t+2} - \hat{y}_{t+2}\|_2^2$;
      Compute state supervision loss: $\mathcal{L}_{\text{state}} = \|\hat{x}'_t - \tilde{x}_t\|_2^2$;
      Compute dynamics loss: $\mathcal{L}_{\text{dyn}} = \|\hat{x}_{t+2} - \tilde{x}_{t+2}\|_2^2$;
      Compute latent consistency loss: $\mathcal{L}_{\text{latent}} = \text{CrossEntropy}(\hat{z}^*_{t+2}, z^*_{t+2})$;
      Total loss: $\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda(\mathcal{L}_{\text{state}} + \mathcal{L}_{\text{latent}} + \mathcal{L}_{\text{dyn}})$;
  Update parameters: $w \leftarrow \texttt{Adam}(\theta, \nabla_w \mathcal{L})$;
return w

angular velocity, contact flags) and a four-dimensional continuous action space for engine thrust. Two key parameters govern the dynamics: main engine power and side engine power. Donkey Car simulates autonomous vehicle control with continuous throttle and steering inputs. We adopt a bicycle dynamics model, where the primary physical parameter to learn is the effective car length. This task presents additional complexity due to nonlinear dynamics and tight coupling between steering and acceleration.

The vision encoder uses two convolutional layers followed by a channel projection. The latent space has 64 dimensions and is quantized using a codebook with 512 entries. The total loss includes reconstruction loss, codebook update loss, and a commitment loss weighted by a factor of 0.25. The interpretable encoder is implemented as a 2-layer Transformer with 4 attention heads and a feedforward dimension of 512. Each codebook index is embedded into a 16-dimensional vector. The encoder output is mean-pooled and passed through a linear layer to regress physical states. The decoder mirrors the encoder and predicts discrete latent indices, optimized with a cross-entropy loss. The total loss includes state regression loss and index reconstruction loss.

D. Interpretable Codebook in VQVAE

For our intrinsic architecture, we aimed to design a VQ-VAE where the discrete codebook itself would be physically interpretable, allowing a single encoder to map directly from images to a structured, meaningful latent space. To this end, we explored several variants.

Our first attempt involved making the physical dimensions of the codebook learnable. In this configuration, a partition of each codebook vector was initialized with values from a discretized grid of physical states, but these values were allowed to be updated via backpropagation during end-to-end training. We hypothesized that the model could learn an optimal embedding for the physical states. However, this approach proved to be unstable. Under the guidance of only weak supervision, the values in these dimensions would often drift significantly from their intended physical semantics, leading to a failure to learn a coherent physical representation.

To enforce a stronger semantic structure and encourage disentanglement, we next designed concept-specific codebooks. This approach assigned separate, independent embedding tables to different physical variables (e.g., one codebook for position, another for angle). These models were trained with modified loss functions that incorporated regularization terms to encourage semantic alignment and penalize mismatches between predicted states and codebook semantics. Despite this stronger structural prior, all variants still suffered from severe codebook utilization collapse, where the model would rely on only a very small subset of the available codebook entries during both training and testing. This resulted in poor diversity in the latent representations and a failure to capture the full range of physical states.

We attribute the failure of these explorations to the lack of a sufficiently strong signal from the weak supervision to guide such a complex and under-constrained optimization problem. Although we adjusted the commitment loss and interpretability regularization weights, the issue persisted.

Given the limitations of these more flexible, learnable designs, we ultimately adopted the simpler and more constrained approach for the intrinsic-discrete model described in Section 3.1, which yielded better and more stable results. In that architecture, we partition each codebook vector $\mathbf{e}_k$ into a physical part $\mathbf{e}_k^p$ and a visual part $\mathbf{e}_k^v$, but the physical part $\mathbf{e}_k^p$ is fixed as a non-learnable constant representing a specific point on the physical state grid. This method, while sacrificing the flexibility of a learnable physical embedding, proved far more robust against codebook collapse and provided the necessary stability for the model to learn effectively.

E. Additional Results

This section provides supplementary results that further validate the conclusions presented in the main paper. We include detailed prediction performance for the CartPole environment, an analysis of the learned physical parameters for CartPole.

Figure 5 displays the 30-step prediction Root Mean Square Error (RMSE) for the CartPole environment. The results are consistent with the findings from the Donkey Car and Lunar Lander environments discussed in the main text. The extrinsic models, particularly our quantized (discrete) variant, consistently achieve lower prediction error across all noise levels (δ) compared to intrinsic models and data-driven baselines like LSTM and Transformer. This reinforces our conclusion that decoupling perception from physical state inference through an extrinsic architecture provides superior long-horizon prediction stability.

To further demonstrate that our model learns a genuinely interpretable representation, we evaluated its ability to recover the true physical parameters of the CartPole simulation. As shown in Figure 6, our PIWM framework successfully identifies the ground-truth values for the pole's mass, the pole's length, the cart's length, and the applied force with high accuracy, especially under low noise conditions (δ=0). While the variance of the learned parameters increases with higher levels of supervision noise, the model's estimates remain centered around the true values, providing strong evidence that the latent space is structured in a physically meaningful way.

Table 2: Controller Performance on Reconstructed Observations Across All Variants and Noise Levels

Case	Input Type	Latent Space	Supervision Noise Level (δ)		
			0%	5%	10%
Donkey Car (Action RMSE ↓)	One-Stage (Intrinsic)	Continuous	0.12 ± 0.04	0.13 ± 0.04	0.15 ± 0.05
		Discrete	0.21 ± 0.15	0.29 ± 0.16	0.32 ± 0.20
	Two-Stage (Extrinsic)	Continuous	0.15 ± 0.05	0.16 ± 0.05	0.22 ± 0.06
		Discrete	0.15 ± 0.04	0.17 ± 0.05	0.19 ± 0.05
Lunar Lander (Action Acc. ↑)	One-Stage (Intrinsic)	Continuous	$93.0\% \pm 1.8\%$	$90.5\% \pm 2.0\%$	$87.1\% \pm 2.2\%$
		Discrete	$85.5\% \pm 2.5\%$	$82.1\% \pm 2.8\%$	$78.3\% \pm 3.1\%$
	Two-Stage (Extrinsic)	Continuous	$86.2\% \pm 2.4\%$	$83.5\% \pm 2.6\%$	$80.0\% \pm 2.9\%$
		Discrete	$91.5\% \pm 2.1\%$	$88.6\% \pm 2.3\%$	$84.5\% \pm 2.5\%$
Cart Pole (Action Acc. ↑)	One-Stage (Intrinsic)	Continuous	$98.0\% \pm 1.0\%$	$96.5\% \pm 1.2\%$	$94.0\% \pm 1.5\%$
		Discrete	$95.0\% \pm 1.6\%$	$91.5\% \pm 2.0\%$	$87.2\% \pm 2.5\%$
	Two-Stage (Extrinsic)	Continuous	$95.5\% \pm 1.5\%$	$92.0\% \pm 1.8\%$	$88.0\% \pm 2.2\%$
		Discrete	$97.2\% \pm 1.1\%$	$95.0\% \pm 1.4\%$	$92.5\% \pm 1.8\%$

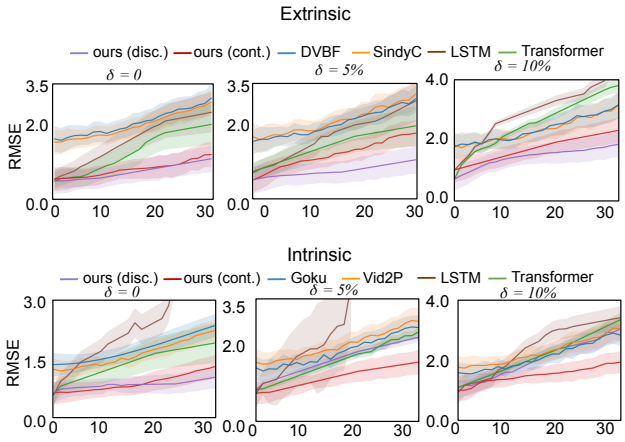

Figure 5: Prediction performance of CartPole. The Root Mean Square Error (RMSE) of our PIWM variants (extrinsic methods, left; intrinsic methods, right) is compared against baselines over a 30-step prediction horizon in the CartPole across varying levels of weak supervision (δ).

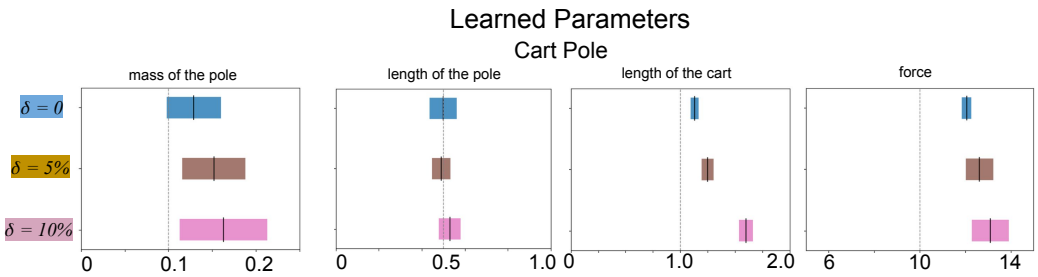

Figure 6: Learned Physical Parameters vs. Ground Truth. The parameters learned by our model (colored bars) are compared against the ground-truth values (dashed lines) under varying noise levels (δ).

F. Dynamics of Cart Pole, Lunar Lander, and Donkey Car

Algorithms 3, 4, and 5 describe the dynamics of the Cart Pole, Lunar Lander, and Donkey Car environments, respectively.

Algorithm 3 models the Cart Pole system, capturing the relationships between the cart's horizontal motion and the pole's angular motion. The dynamics incorporate applied forces and resulting accelerations, governing how the position, velocity, pole angle, and angular velocity evolve over time.

Algorithm 4 presents the dynamics of the Lunar Lander, where discrete actions determine the firing of the main and side engines. The equations govern the lander's position, velocity, orientation, and angular velocity, enabling simulation of its flight behavior under various control inputs.

Algorithm 5 introduces a simplified learnable bicycle model used to approximate the Donkey Car's dynamics. While the true Donkey Car system implemented in Unity involves complex physics and interactions, this model captures essential relationships between the vehicle's position, heading, and speed under steering and acceleration commands, allowing efficient learning and prediction without requiring access to the full simulation engine.

Algorithm 2: Ablation Model: Learning Dynamics from Consecutive Latents to Predict Future States

Input: Training dataset V containing sequences of images and weak state labels $(y_{1:M}, [x^{lo}_{1:M}, x^{up}_{1:M}])$, batch size B, weight λ_1, optimizer `Adam()`

Output: Trained model $(\mathcal{E}, \phi_\theta, \mathcal{D})$ with parameters θ

Initialize parameters θ randomly;

for $i \leftarrow 1$ **to** $len(V)/B$ **do**

  Sample a batch of B sequences from V;

  foreach *sequence in batch* **do**

    for $j \leftarrow 1$ **to** $M-2$ **do**

      $(\mu_{z_j}, \sigma^2_{z_j}) = \mathcal{E}(y_j)$;

      $(\mu_{z_{j+1}}, \sigma^2_{z_{j+1}}) = \mathcal{E}(y_{j+1})$;

      ; `// Encode two consecutive observations`

      $z_j \sim \mathcal{N}(\mu_{z_j}, \sigma^2_{z_j})$;

      $z_{j+1} \sim \mathcal{N}(\mu_{z_{j+1}}, \sigma^2_{z_{j+1}})$;

      ; `// Sample latent variables`

      $\Delta\hat{z}_j = \phi_\theta(z_j, z_{j+1})$;

      ; `// Predict latent dynamics based on` z_j `and` z_{j+1}

      $\hat{z}_{j+2} = z_{j+1} + \Delta\hat{z}_j$;

      ; `// Predict next latent state`

      $\hat{y}_{j+2} = \mathcal{D}(\hat{z}_{j+2})$;

      ; `// Decode predicted latent to reconstruct future observation`

      $\mathcal{L}_0 = \frac{1}{D}\sum_{k=1}^{D} \|\hat{y}_{j+2} - y_{j+2}\|^2$;

      ; `// Reconstruction loss against true` y_{j+2}

      $p \sim \mathcal{N}\left(\frac{x^{up}_j - x^{lo}_j}{2}, \left(\frac{x^{lo}_j + x^{up}_j}{6}\right)^2\right)$;

      $q \sim \mathcal{N}(\mu_{z_j}, \sigma^2_{z_j})$;

      $\mathcal{L}_1 = \sum_{k=1}^{d} \left\| \frac{1}{2}\left(\frac{\sigma_p^2}{\sigma_q^2} + \frac{(\mu_q - \mu_p)^2}{\sigma_q^2} - 1 + \ln\frac{\sigma_q^2}{\sigma_p^2}\right)\right\|^2$;

      ; `// KL divergence with weak supervision`

  Update parameters: $\theta \leftarrow$ `Adam`$(\theta, \nabla_\theta(\mathcal{L}_0 + \lambda_1\mathcal{L}_1))$;

return θ

Algorithm 3: Dynamics of Cart Pole

Input: Current state $z = [x, \dot{x}, \theta, \dot{\theta}]$, action a

Output: Updated state $z_{\text{new}} = [x_{\text{new}}, \dot{x}_{\text{new}}, \theta_{\text{new}}, \dot{\theta}_{\text{new}}]$

// Extract State Variables

$x, \dot{x}, \theta, \dot{\theta} \leftarrow z[:,0], z[:,1], z[:,2], z[:,3]$

// Convert Action to Force

$F \leftarrow \text{force\_mag} \times (2 \cdot a - 1)$

// Compute Trigonometric Values

$\cos(\theta) \leftarrow \text{costheta}, \quad \sin(\theta) \leftarrow \text{sintheta}$

// Compute Intermediate Variable

$\text{temp} \leftarrow \frac{F + m_p \cdot l \cdot \dot{\theta}^2 \cdot \sin(\theta)}{m_p + m_c}$

// Calculate Angular Acceleration

$\ddot{\theta} \leftarrow \frac{g \cdot \sin(\theta) - \cos(\theta) \cdot \text{temp}}{l \cdot \left(\frac{4}{3} - \frac{m_p \cdot \cos^2(\theta)}{m_p + m_c}\right)}$

// Calculate Linear Acceleration

$\ddot{x} \leftarrow \text{temp} - \frac{m_p \cdot l \cdot \ddot{\theta} \cdot \cos(\theta)}{m_p + m_c}$

// Update State Variables

$x_{\text{new}} \leftarrow x + \tau \cdot \dot{x}\ \dot{x}_{\text{new}} \leftarrow \dot{x} + \tau \cdot \ddot{x}\ \theta_{\text{new}} \leftarrow \theta + \tau \cdot \dot{\theta}\ \dot{\theta}_{\text{new}} \leftarrow \dot{\theta} + \tau \cdot \ddot{\theta}$

// Return Updated State

$z_{\text{new}} \leftarrow [x_{\text{new}}, \dot{x}_{\text{new}}, \theta_{\text{new}}, \dot{\theta}_{\text{new}}]$ **return** z_{new};

Algorithm 4: Dynamics of Lunar Lander

Input: Current states $\mathbf{s} = [x, y, \dot{x}, \dot{y}, \theta, \dot{\theta}]$, actions $\mathbf{a}$ (0: do nothing, 1: fire left, 2: fire main, 3: fire right)

Output: Updated states $\mathbf{s}_{\text{new}} = [x_{\text{new}}, y_{\text{new}}, \dot{x}_{\text{new}}, \dot{y}_{\text{new}}, \theta_{\text{new}}, \dot{\theta}_{\text{new}}]$

// Unpack State Variables

$x \leftarrow \mathbf{s}[:,0], \quad y \leftarrow \mathbf{s}[:,1]\ \dot{x} \leftarrow \mathbf{s}[:,2], \quad \dot{y} \leftarrow \mathbf{s}[:,3]\ \theta \leftarrow \mathbf{s}[:,4], \quad \dot{\theta} \leftarrow \mathbf{s}[:,5]$

// Calculate Engine Direction and Dispersion

$\text{tip} \leftarrow [\sin(\theta), \cos(\theta)]\ \text{side} \leftarrow [-\cos(\theta), \sin(\theta)]$

// Process Actions

$\text{fire\_main} \leftarrow (\mathbf{a} == 2)\ \text{fire\_left} \leftarrow (\mathbf{a} == 1)\ \text{fire\_right} \leftarrow (\mathbf{a} == 3)$

// Compute Main Engine Thrust

$m_{\text{power}} \leftarrow \text{fire\_main}\ \dot{x} \leftarrow \dot{x} - \text{tip}[:,0] \cdot \text{main\_power} \cdot m_{\text{power}}/\text{FPS}$

$\dot{y} \leftarrow \dot{y} + \text{tip}[:,1] \cdot \text{main\_power} \cdot m_{\text{power}}/\text{FPS}$

// Compute Side Engine Thrust

$s_{\text{power}} \leftarrow \text{fire\_left} + \text{fire\_right}\ \text{direction} \leftarrow \text{fire\_right} - \text{fire\_left}$

$\dot{x} \leftarrow \dot{x} + \text{side}[:,0] \cdot \text{side\_power} \cdot s_{\text{power}} \cdot \text{direction}/\text{FPS}$

$\dot{\theta} \leftarrow \dot{\theta} + \text{side\_power} \cdot s_{\text{power}} \cdot \text{direction}/\text{FPS}$

// Update Position and Angle

$x \leftarrow x + \dot{x}/\text{FPS}\ y \leftarrow y + \dot{y}/\text{FPS}\ \theta \leftarrow \theta + \dot{\theta}/\text{FPS}$

// Create Updated States

$\mathbf{s}_{\text{new}} \leftarrow [x, y, \dot{x}, \dot{y}, \theta, \dot{\theta}]$ **return** $\mathbf{s}_{\text{new}}$;

Algorithm 5: Dynamics of Bicycle Model

Input: Current state $s = [x, y, \theta, v]$, action $a = [\delta, a_{\text{acc}}]$
Output: Updated state $s_{\text{new}} = [x_{\text{new}}, y_{\text{new}}, \theta_{\text{new}}, v_{\text{new}}]$

// Extract State Variables
$x, y, \theta, v \leftarrow s[:, 0], s[:, 1], s[:, 2], s[:, 3]$
// Extract Action Variables
$\delta, a_{\text{acc}} \leftarrow a[:, 0], a[:, 1]$
// Update State Variables
$x_{\text{new}} \leftarrow x + v \cdot \cos(\theta) \cdot \tau \; y_{\text{new}} \leftarrow y + v \cdot \sin(\theta) \cdot \tau \; \theta_{\text{new}} \leftarrow \theta + \frac{v}{L} \cdot \tan(\delta) \cdot \tau \; v_{\text{new}} \leftarrow v + a_{\text{acc}} \cdot \tau$
// Return Updated State
$s_{\text{new}} \leftarrow [x_{\text{new}}, y_{\text{new}}, \theta_{\text{new}}, v_{\text{new}}]$ **return** s_{new};