JEPA-BISIM: LEARNING ROBUST VISUAL REPRESENTATIONS FOR PLANNING WITH JOINT-EMBEDDING PREDICTIVE WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models learned from high-dimensional visual observations allow agents to make decisions and plan directly in latent space, avoiding pixel-level reconstruction. However, recent latent predictive architectures (JEPAs), including the DINO world model (DINO-WM), display a degradation in test time robustness due to their sensitivity to "slow features". These include visual variations such as background changes and distractors that are irrelevant to the task being solved. We address this limitation by augmenting the predictive objective with a bisimulation encoder that enforces control-relevant state equivalence, mapping states with similar transition dynamics to nearby latent states while limiting contributions from slow features. We evaluate our model on a navigation task (PointMaze) and on a manipulation task (PushT) under different test-time background changes and visual distractors. Across all benchmarks, our model consistently improves robustness to slow features while operating in a reduced latent space, up to $10\times$ smaller than that of DINO-WM. Moreover, our model is agnostic to the choice of pretrained visual encoder and maintains robustness when paired with DINOv2, SimDINOv2, and iBOT features.

1 INTRODUCTION

World models support planning and control by learning latent dynamics from high-dimensional visual observations. An impactful line of recent work centers on Joint Embedding Predictive Architectures (JEPAs), which eschew reconstruction of raw observations in favor of learning representations by predicting future latent embeddings (Grill et al., 2020; Caron et al., 2021; LeCun, 2022; Assran et al., 2023; Sobal et al., 2025; Balestriero & LeCun, 2025). Building on JEPAs, the DINO world model (DINO-WM) (Zhou et al., 2025) learns a latent predictive dynamic by encoding visual observations through a pretrained DINOv2 feature extractor (Oquab et al., 2023). This yields zero-shot planning, with a pretrained, reward-free world model on entirely new tasks.

Despite their empirical success, JEPAs suffer from a fundamental limitation: they pay too much attention to *slowly changing* and *task-irrelevant features* (LeCun, 2022; Sobal et al., 2022). These are aspects of the raw observations that evolve temporally, such as background textures, lighting, camera viewpoints, and distractors. While they frequently co-occur with the related task-irrelevant visual variations (nuisance factors like dynamic distractors, for example), slow features are distinct in that they have no bearing on the control-relevant transition dynamics. But given their disproportional ease of predictability, they also present a degenerate solution to the JEPA objective by *only* encoding these temporally consistent slow features. Consequently, any downstream world models trained with these poor input representations fail to generalize to test-time environments with *identical* dynamics to those in training, and diverging in the slow features present in the observations. Indeed, Sobal et al. (2022) verify this behavior empirically; we also observe this in Figure 2.

Importantly, this limitation *cannot* be fully resolved by larger visual foundation models alone (such as DINOv2 (Oquab et al., 2023), SimDINOv2 (Wu et al., 2025), iBOT (Zhou et al., 2021), masked autoencoders (He et al., 2022), among others) as they also extract control-irrelevant information. Even with invariant perception modules, the learned latent dynamics may still entangle control-relevant states with nuisance visual features, leading to brittle planning under visual distribution shifts at test time. Ideally, a world model should encode only those aspects of the state that are relevant for predicting future states, while being invariant to task-irrelevant visual factors.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Question. Can we learn latent world models that retain the joint-embedding predictive strengths while also being robust to the slow features?

We address this question using bisimulation metrics for learning invariant representations on top of pretrained visual feature extractors (i.e., DINOv2, SimDINOv2, iBOT). These metrics define state equivalences based on immediate rewards and transition dynamics, and have been studied as a principled foundation for state abstraction in Markov Decision Processes (MDPs) (Ferns et al., 2004; Castro et al., 2009), transfer learning (Castro & Precup, 2010), and invariant representation learning (Zhang et al., 2021; Kemertas & Aumentado-Armstrong, 2021; Tkachev & Abate, 2011; Shimizu & Tomizuka, 2025). By construction, encoders trained with a bisimulation regularizer discard nuisance features that do not affect control, making it a natural candidate for addressing the slow-feature limitation of JEPAs. Compared to methods that learn distance-aware representations (Park et al., 2024; Wang et al., 2026), our approach specifically targets the adaptation of rich pretrained visual feature spaces (DINOv2, SimDINOv2, iBOT) for planning, leveraging their semantic richness while suppressing control-irrelevant information.

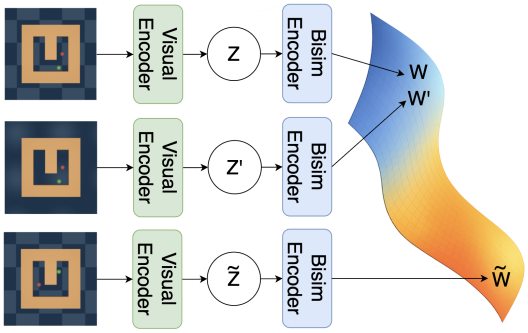


Figure 1: Visually distinct observations that differ only in background (checkerboard and gradient) are first mapped to latent embeddings Z and Z' by a pretrained encoder (initial state in green, goal in red). A bisimulation encoder then projects these into lower-dimensional representations W and W', which are equivalent under on-policy transition dynamics.

Key Point. DINO-WM (Zhou et al., 2025) displays a substantial degradation in performance on relatively simple navigation tasks (e.g., PointMaze (Fu et al., 2020)) when evaluated under background changes and visual distractors at test time (see Figure 2-(Right)). To address this, we propose a *bisimulation encoder* for visual world models that augments latent dynamics with an on-policy bisimulation objective, enforcing invariance in the learned latent space. The bisimulation encoder is trained jointly with the transition model, enforcing states with similar transition behavior to be mapped to nearby latent embeddings while discarding task-irrelevant visual information (see Figure 1). Importantly, we also show that the reward signal is not fundamental to learning invariant world-model representations, thus simplifying training.

Contributions. Our main contributions are as follows:

- We propose a bisimulation encoder for visual world models that enforces control-relevant state equivalence while discarding task-irrelevant visual features from pretrained visual feature extractors (e.g., DINOv2, SimDINOv2, iBOT).
- We integrate the bisimulation objective into a latent predictive world model architecture (see Figure 2-(Left)), showing that invariant representations can be learned jointly with the transition model *without* relying on reward prediction.
- We show that our bisimulation encoder yields robust planning performance under several background changes and moving distractors for navigation and manipulation tasks, with latent embeddings dimensions that are $10\times$ smaller than DINO-WM.

Apart from the works referenced above, we also draw inspiration from the myriad works on Joint Embedding and Predictive Architectures, introduced in LeCun (2022); Sobal et al. (2025), as well as the seminal works of Castro et al. (2009); Castro & Precup (2010); Zhang et al. (2021) on bisimulation metrics and invariant representations. As a result, this paper is also closely related to work at the intersection of model-based control and latent world models (Sun et al., 2024; Lim et al., 2025) Due to limitations in space, we refer the reader to Appendix A.1 for a more detailed discussion

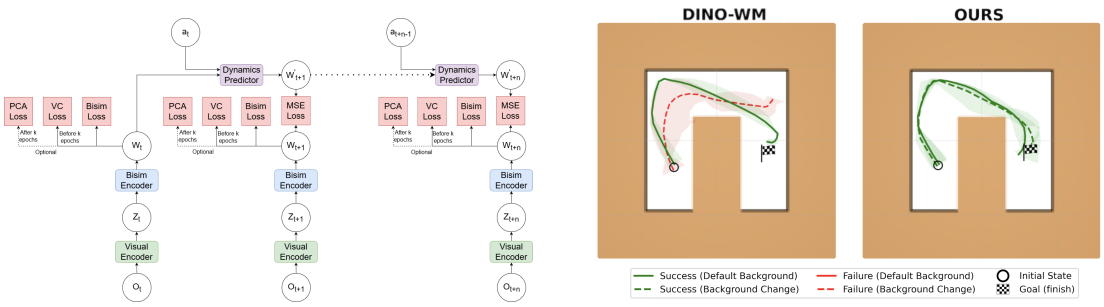


Figure 2: **Left.** Model architecture and training objectives. **Right.** Rollouts for PointMaze navigation under a background change at test time. For clarity, the background change is not depicted in the figure. See Section 4.2 for details on background changes. While DINO-WM fails to reach the goal due to background change our model succeeds.

2 PRELIMINARIES

Throughout, we consider a vision-based control task given by a partially observable MDP (POMDP) that evolves in discrete time $t \in \{0, 1, 2, \dots\}$. An observation is an RGB image $o_t \in \mathcal{O}$, an action is denoted by $a_t \in \mathcal{A}$, and a trajectory is a sequence $(o_t, a_t, o_{t+1}, a_{t+1}, \dots)$. We denote the data distribution induced by a behavior policy π by $\mathcal{D}_\pi$, meaning that $(o_t, a_t, o_{t+1}) \sim \mathcal{D}_\pi$ when rollouts are generated by playing the environment with π. We use $\mathbb{E}_{\mathcal{D}_\pi}[\cdot]$ to denote the expectation with respect to this data distribution. We use $\mathcal{B}$ to denote the replay buffer (distinct from the data distribution $\mathcal{D}_\pi$); the replay buffer stores previously collected transitions and is used for pairwise comparisons in the bisimulation loss (see Appendix A.2).

2.1 JOINT-EMBEDDING PREDICTIVE ARCHITECTURES

JEPAs learn a visual encoder $f_\theta : \mathcal{O} \to \mathbb{R}^{d_z}$ that maps an observation o_t to a latent embedding, i.e., $z_t = f_\theta(o_t) \in \mathbb{R}^{d_z}$, where d_z is the embedding dimension and θ is the vector of encoder parameters. Jointly, a predictor (i.e., latent dynamics model) $\tilde{T}_\phi : \mathbb{R}^{d_z} \times \mathcal{A} \to \mathbb{R}^{d_z}$ is trained to map the current latent state and action to a predicted next latent state $\hat{z}_{t+1} = \tilde{T}_\phi(z_t, a_t)$, with parameters ϕ.

Since representations are defined up to arbitrary invertible transformations, JEPA training includes a target embedding. The target is denoted by $z_{t+1} = f_\theta(o_{t+1})$. The JEPA objective minimizes the prediction loss between $\hat{z}_{t+1}$ and z_{t+1},

$$\mathcal{L}_{\text{pred}}(\theta, \phi) = \mathbb{E}_{\mathcal{D}_\pi}\left[\ell\left(\tilde{T}_\phi(f_\theta(o_t), a_t), f_\theta(o_{t+1})\right)\right], \tag{1}$$

where $\ell : \mathbb{R}^{d_z} \times \mathbb{R}^{d_z} \to \mathbb{R}_+$ is the mean squared error.

We emphasize that DINO-WM (Zhou et al., 2025), follow this principle, with the key distinction that the visual encoder parameters θ are fixed and not learned during training. The motivation for using frozen pretrained encoders rather than learning from scratch is twofold: (i) large-scale pretrained visual features (DINOv2, SimDINOv2, iBOT) provide semantically rich representations that improve sample efficiency for dynamics learning, and (ii) fixing the encoder decouples representation learning from dynamics learning, enabling modular composition with different visual backbones. The bisimulation encoder then adapts these fixed features to be control-relevant, combining the best of both worlds. Therefore, observations are first mapped to latent embeddings using a pretrained visual foundation model (e.g., DINOv2 (Oquab et al., 2023)), and a predictive dynamics model is trained on top of that latent space. Here, we also treat the pretrained visual encoder as fixed and train an extra bisimulation encoder jointly with the dynamics on top of it.

Remark 2.1 (Slow Features)**.** *We emphasize that JEPAs inadvertently focus on the slow features. In particular, when observations contain static backgrounds or persistent distractors that vary slowly relative to task-relevant dynamics, the prediction loss* (1) *can be minimized by encoding such features into the latent embedding $z_t = f_\theta(o_t)$, since they induce small temporal differences $\|z_{t+1} - z_t\|$. As a result, slowly varying nuisance information may dominate the predictive objective, leading to latent embeddings that fail to capture the underlying task structure (Sobal et al., 2022) and thus are not generalizable to background changes at test time (see Figure 2-(Right)). This limitation motivates the use of bisimulation. By defining latent distances in terms of on-policy tran-*

sition behavior, bisimulation induces representations that collapse nuisance visual features while preserving the task-relevant information for planning.

2.2 BISIMULATION METRIC

Let us now (informally) revisit the definition of on-policy bisimulation metrics from Castro (2020). Intuitively, two latent states, in the pretrained visual encoder space, z and z' are "bisimilar" if they induce similar immediate rewards (for optimality equivalence) and similar future state evolution (for dynamics equivalence) under the same actions.

Here, the environment is partially observed and represented through latent embeddings obtained from a visual encoder (e.g., DINOv2, SimDINOv2, iBOT). The goal is to learn a lower-dimensional representation $w = h_\eta(z) \in \mathbb{R}^{d_w}$, with $d_w \ll d_z$, that preserves control-relevant structure while discarding nuisance visual variation. We formalize state equivalence with the *on-policy* bisimulation metric (Castro, 2020).

Definition 2.1 (On-Policy Bisimulation Metric (Castro, 2020))**.** *Given a policy π, a reward function $r(z, a)$, and a transition kernel $P_\pi(\cdot \mid z)$ induced by π, the on-policy bisimulation metric is*

$$d_\pi(z, z') = \underbrace{|r_\pi(z) - r_\pi(z')|}_{\text{reward similarity}} + \gamma \underbrace{W_1\left(P_\pi(\cdot \mid z), P_\pi(\cdot \mid z')\right)}_{\text{dynamics similarity}}, \text{ with } r_\pi(z) = \mathbb{E}_{a\sim\pi(\cdot|z)}[r(z, a)],$$

where $\gamma \in (0, 1)$ is the discount factor and W_1 denotes the 1-Wasserstein distance.

On Dropping the Reward Term. Note that $d_\pi(z, z')$ has two terms: a reward similarity term $|r_\pi(z) - r_\pi(z')|$ and a dynamics similarity term measured via the 1-Wasserstein distance (Givens & Shortt, 1984). The reward term ensures that if two state trajectories are similar, they are not collapsed if they yield different rewards. In particular, such reward discrepancy is crucial when bisimulation is used for policy transfer or value-function approximation (Ferns et al., 2004; Castro et al., 2009; Castro & Precup, 2010). However, for the purpose of learning an invariant representation for JEPAs, we claim that this term is not fundamental. That is because the transition term alone induces a meaningful notion of equivalence: states that exhibit similar evolution under the same action sequences are mapped to nearby latent representations in the bisimulation latent space (see our illustration in Figure 1)

Moreover, dropping the reward term is well aligned with JEPAs, which are trained without reward signal on offline trajectory data and rely on predictive consistency in latent space rather than task-specific reward structure. Incorporating rewards would require task-specific supervision and would break the modularity between representation learning and zero-shot downstream planning – a core design principle of JEPAs (LeCun, 2022). Accordingly, we omit the reward term in our implementation and rely only on approximating the Wasserstein transition similarity to enforce invariance. This design choice is also supported numerically in Appendix A.8, where a reward predictor is included.

2.3 LEARNING INVARIANT REPRESENTATIONS

Now, we discuss the invariant representation learning problem for our joint-embedding predictive world model. Recall that $z_t = f_\theta(o_t) \in \mathbb{R}^{d_z}$ typically contains both control-relevant information and nuisance features. Our goal is to learn an encoder $w_t = h_\eta(z_t) \in \mathbb{R}^{d_w}$, that limits the contribution of the slow features in the transition dynamics.

To do so, we jointly learn the dynamics in the bisimulation space through $T_\phi : \mathbb{R}^{d_w} \times \mathcal{A} \to \mathbb{R}^{d_w}$, such that $\hat{w}_{t+1} = T_\phi(w_t, a_t)$ predicts the next latent state. With the notation set up in Section 2.1, we have the predictive dynamics augmented with the bisimulation encoder, i.e., $T_\phi(h_\eta(f_\theta(o_t)), a_t)$, where again θ is fixed.

To connect invariance to bisimulation, we consider an abstract state space $\mathcal{W} \subseteq \mathbb{R}^{d_w}$ equipped with the metric d_π. We aim to learn h_η and T_ϕ jointly so that the distances in $\mathcal{W}$ approximate an on-policy bisimulation metric, meaning that states that are equivalent for control under rollouts are mapped nearby in such latent space. We do this by optimizing a bisimulation objective that aligns pairwise latent distances with a target quantity that compares on-policy dynamics. Concretely, for a pair of observation, action, and next observation (o_t, a_t, o_{t+1}) and (o'_t, a'_t, o'_{t+1}) drawn from $\mathcal{D}_\pi$, we have the latent states

$$\underbrace{z_t = f_\theta(o_t),\ z'_t = f_\theta(o'_t)}_{\text{Pretrained latents}},\ \underbrace{w_t = h_\eta(z_t),\ w'_t = h_\eta(z'_t)}_{\text{Bisimulation latents}}.$$

The transition dynamics T_ϕ is trained using

$$\mathcal{L}_{\text{dyn}}(\eta, \phi) \triangleq \mathbb{E}_{\mathcal{D}_\pi}\Big[\|h_\eta(f_\theta(o_{t+1})) - T_\phi(h_\eta(f_\theta(o_t)), a_t)\|^2\Big],$$

and we note that, in practice, as we do not have $\mathcal{D}_\pi$, we compute the empirical mean (MSE) instead.

On the other hand, for invariance, our loss encourages the distance $\|w_t - w'_t\|_2$ to match a control-relevant similarity signal measured by the bisimulation metric. We define the on-policy one-step bisimulation target between states as $\Delta_{\text{bisim}}(\eta, \phi) \triangleq \gamma \|T_\phi(w_t, a_t) - T_\phi(w'_t, a'_t)\|_2$, where it compares the predicted next latent states under the learned dynamics. Note that the actions a_t and a'_t are drawn independently from the two separate trajectories sampled from the replay buffer $\mathcal{B}$; they are not required to be the same. This is consistent with the on-policy bisimulation formulation of Castro (2020), where the metric averages over the policy's action distribution. In practice, using independently sampled actions provides a diverse set of comparisons across the replay buffer, which stabilizes training. We fit the bisimulation encoder by minimizing the following loss:

$$\mathcal{L}_{\text{bisim}}(\eta, \phi) \triangleq \mathbb{E}_{\mathcal{D}_\pi}\left[\left(\|w_t - w'_t\|_2 - \Delta_{\text{bisim}}(\eta, \phi)\right)^2\right].$$

Finally, we define the overall training objective as

$$\mathcal{L}_{\text{jepa-bisim}}(\eta, \phi) \triangleq \mathcal{L}_{\text{dyn}}(\eta, \phi) + \lambda_{\text{bisim}} \mathcal{L}_{\text{bisim}}(\eta, \phi), \tag{2}$$

with $\lambda_{\text{bisim}} > 0$ being the regularization weight.

Minimizing the loss formalizes our goal of learning a compact latent space that preserves on-policy predictive structure for planning while suppressing task-irrelevant visual features. We next describe the architecture, loss instantiation, and planning in the learned bisimulation space.

3 Our World Model

We now describe our architecture for learning invariant world models that builds on pretrained visual features (DINOv2, SimDINOv2, and iBOT) and augments them with a bisimulation encoder to suppress task-irrelevant information. The model learns latent transition dynamics and performs planning with CEM. An overview of the pipeline objectives is shown in Figure 2-(Left).

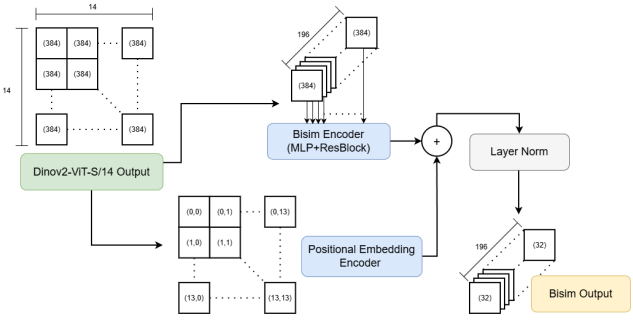


Figure 3: Bisimulation encoder with patch-based design, with 196 of patches and 32 output patch dimension.

3.1 Pretrained Encoders

In our experiments, we consider three self-supervised visual encoders: DINOv2 (Oquab et al., 2023) (with ViT-S/14), SimDINOv2 (Wu et al., 2025) (with ViT-B/16), and iBOT (Zhou et al., 2021) (with ViT-S/16). These are trained on large-scale image datasets using joint embedding or masked prediction objectives and have been shown to produce semantically rich visual features (Jiang et al., 2023). We include SimDINOv2 as a variant of DINOv2 that incorporates coding-rate regularization for improved feature quality, and iBOT as a masked-prediction alternative. These pre-trained encoders allow us to assess whether the bisimulation encoder's benefits are agnostic to the choice of visual backbone. When using DINOv2 features, we refer to our model as DINO-Bisim.

3.2 Bisimulation Encoder

Given pretrained visual embeddings z_t, we learn a bisimulation encoder $h_\eta : \mathbb{R}^{N_p \times d_z} \to \mathbb{R}^{N_p \times d_w}$, $w_t = h_\eta(z_t)$, as discussed in Section 2.2. We note that in contrast to Zhang et al. (2021); Shimizu & Tomizuka (2025), our bisimulation encoder is designed in patches (we refer the reader to Appendix A.2.3 for details). Such encoder consists of a per-patch residual MLP projection head (Linear-ResMLP-Linear) that maps each DINOv2 patch embedding (or raw image patch) into a $d_w = 32$ dimensional latent, followed by learned spatial positional embeddings and LayerNorm.

This encoder is trained to ensure that distances in the w-space reflect on-policy bisimulation distances, as defined in (2). To this end, we *flatten* the spatial patch features prior to computing the previously defined losses and compare states across different trajectories that are stored in a replay

buffer $\mathcal{B}$, following Zhang et al. (2021). However, minimizing the bisimulation loss admits degenerate solutions as mapping all states to a single representation trivially minimizes pairwise distances, and we refer to this as the representation *collapse*. To circumvent such collapse, prior work (Bardes et al., 2021; Zhu et al., 2023) has adopted variance-invariance-covariance regularization (VICReg) to guarantee that each latent dimension keeps sufficient variance over batches of states (i.e., to guarantee that the latents are sufficiently spread). We refer the reader to (Bardes et al., 2021, Section 4.1) for an excellent overview of the approach. Here, invariance is given by the bisimulation loss.

In our setting, however, the standard VICReg (Bardes et al., 2021) *may be* insufficient. This is because, for the pretrained visual encoders considered in this work (i.e., DINOv2, SimDINOv2, and iBOT) a large fraction of the variance is often concentrated in a small number of principal components that correlate with the slow features rather than task-relevant dynamics. Therefore, imposing a uniform variance across all coordinates may preserve nuisance visual features, preventing the bisimulation encoder from collapsing the task-irrelevant information.

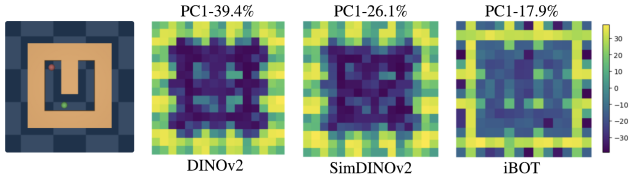


Figure 4: Visualization of the first principal component (PC1) of latent embeddings produced by different visual feature encoders for a PointMaze observation. In all cases, PC1 predominantly encodes background and layout information rather than control-relevant features, motivating our PCA-based VC-Reg in the bisimulation encoder.

Leveraging this insight, we propose a PCA-based variance-covariance regularization (VC-Reg) loss, which adapts VICReg to pretrained visual embeddings by guaranteeing a minimum variance in directions that are not dominated by a large fraction of slow features. Throughout the paper, we use VICReg to refer to the original formulation of Bardes et al. (2021), which enforces variance, invariance, and covariance regularization uniformly across all latent dimensions. We use VC-Reg (or PCA-VC-Reg) to refer to our variant, which applies variance and covariance regularization selectively in the PCA basis, omitting the invariance term (handled by the bisimulation loss instead).

3.3 PCA-based Variance Covariance Regularization

As illustrated in Figure 4, the pretrained visual embeddings from DINOv2, SimDINOv2, and iBOT allocate a large amount of variance to the first principal component (PC1), which are correlated with task-irrelevant features coming from backgrounds and distractors. This concentration of task-irrelevant variance in the leading PC is a common property of self-supervised visual encoders: as noted in the DINOv2 paper (Oquab et al., 2023), background information tends to dominate the first principal component and can be attenuated by thresholding it. While this property may not hold universally for all encoders or environments, it is observed consistently for DINOv2, SimDINOv2, and iBOT in our PointMaze setting (see Figure 4 and Appendix A.3). If we use the standard VICReg loss with uniform variance across the coordinates of the bisimulation latent space, the optimizer may still preserve such irrelevant directions simply to satisfy the variance threshold, conflicting with the goal of learning a representation invariant to those slow features. Therefore, we do VC-Reg in the PCA basis of the current batch (see Appendix A.3 for more details).

We emphasize that if the visual encoder does not isolate control-irrelevant features within a dominant principal component, the PCA-based VC-Reg regularization becomes *optional*. In this case, the bisimulation-based encoder alone can mitigate, though only partially, the influence of task-irrelevant variations in the learned representation.

3.4 Transition Model

We model latent dynamics using a vision transformer that predicts the next state in the bisimulation latent space. Following DINO-WM (Zhou et al., 2025), the vision transformer architecture (Dosovitskiy, 2020) is suitable for processing the underlying patch-based embeddings. Since the bisimulation latent states are already represented as sequences of patch embeddings, we remove the ViT tokenization layer, yielding a transformer architecture operating directly on patch-level bisimulation features without a separate tokenization step (analogous to a "decoder-only" design in LLMs, where the transformer processes tokens causally without a separate encoder stage).

The transition model takes as input a history of past bisimulation latent states and actions across multiple trajectories, $\{(w_{j,t-H:t-1}, a_{j,t-H:t-1})\}_{j=1}^{N_{\text{traj}}}$, where H denotes the context length and N_{traj}

the number of trajectories, and it predicts the next latent state $\hat{w}_t$ (i.e., in the bisimulation space). Note that the subscript j denotes the trajectory embedding and action at time step t. To respect temporal causality, a causal attention mask is implemented (as in Zhou et al. (2025)) ensuring that predictions at time t depend only on past latent states and actions. More precisely, each patch embedding w_t^i attends to the corresponding patch embeddings $\{w_{t-H:t-1}^i\}_{i=1}^{N_p}$ over the context. Predicting all patches jointly yields global spatial structure and temporal dynamics that improves temporal generalization as discussed in Zhou et al. (2025).

By "filtering out" task-irrelevant visual features, the bisimulation encoder enables the transition model to learn dynamics that are invariant to nuisance visual factors (backgrounds, distractors), focusing only on control-relevant state transitions. The learned joint-embedding predictive world model is then used for planning via MPC (Kouvaritakis & Cannon, 2016), with actions selected using the cross-entropy method (CEM).

3.5 Planning

Given an initial observation o_0 and a goal observation o_g, we encode $w_0 = h_\eta(f_\theta(o_0))$ and the goal $w_g = h_\eta(f_\theta(o_g))$. For a fixed goal, we define the latent goal-reaching cost as follows:

$$c(w) \triangleq \|w - w_g\|_2^2. \tag{3}$$

We optimize the action sequence $\boldsymbol{a} = (a_0, \ldots, a_{T-1})$ through the learned latent dynamics:

$$\min_{\boldsymbol{a}} \quad c(\hat{w}_T) \quad \text{subject to} \quad \hat{w}_{t+1} = T_\phi(\hat{w}_t, a_t), \quad t = 0, \ldots, T-1, \quad \hat{w}_0 = w_0. \tag{4}$$

We solve this terminal-cost objective using CEM with a planning horizon of $T = 5$ in a receding-horizon MPC scheme, executing the first action and replanning at each step. Our bisimulation latent space is $10\times$ smaller than DINO-WM's, reducing the computational cost of latent rollouts.

3.6 A Generalization Bound

Before turning to the experiments, we also provide a theoretical justification for learning bisimulation-based representations on top of JEPA latents. Let us first define the T-step on-policy cost-to-go as follows:

$$J_T^\pi(z) \triangleq \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t c\left(h_\eta(z_t)\right) \mid z_0 = z\right], \; z_{t+1} \sim P_\pi(\cdot \mid z_t),$$

where the expectation is taken over trajectories $\{z_t\}_{t=0}^T$ induced by policy π and transition kernel P_π starting from a visual pretrained latent state $z_0 = z$. We defer the proof of Theorem 3.1 to Appendix A.11.

Theorem 3.1 (Reward-free planning generalization bound)**.** *Suppose $h_\eta(\cdot)$ ε-approximately preserves the reward-free bisimulation metric in the sense that*

$$\left|\|h_\eta(z) - h_\eta(z')\|_2 - d_\pi(z, z')\right| \le \varepsilon, \;\; \textit{for any } z, z'.$$

In addition, suppose that $h_\eta(\cdot)$ is uniformly bounded as $\|h_\eta(z)\|_2 \le H_w$ for all z and thus $\|w_g\|_2 \le H_w$. Then for any two pretrained visual latent embeddings z, z',

$$|J_T^\pi(z) - J_T^\pi(z')| \le 8 H_w T d_\pi(z, z'). \tag{5}$$

We also note that the linear dependence on the horizon T contrasts with the $(1-\gamma^T)/(1-\gamma)$ dependence from (Zhang et al., 2021; Shimizu & Tomizuka, 2025). This difference arises from dropping the reward term in the bisimulation metric: while reward-aware bisimulation enables geometric discounting of cost discrepancies through immediate reward alignment, a reward-free metric propagates the discrepancies through the transition dynamics, yielding a cumulative horizon-dependent bound. In this sense, the T factor can be interpreted as the price paid for enforcing control-relevant invariance without task-specific reward supervision.

Key Takeaway. The above result implies that if two latent states in the pretrained visual encoder space differ only due to nuisance visual variations, such as slow features, then their induced planning behavior remains similar. More concretely, let $z = f_\theta(o)$ embed a PointMaze scene with a checkerboard background, and let $z' = f_\theta(o')$ embed the same scene with a gradient background (see Figure 1), since these visual changes do not affect on-policy transition behavior, the bisimulation distance $d_\pi(z, z')$ is small. Consequently, the difference in the induced planning cost is bounded by $\mathcal{O}\left(d_\pi(z, z')\, T\right)$. Therefore, robustness of planning when starting from w or w' in the bisimulation latent space is governed by their distance in that space.

4 EXPERIMENTS

We now evaluate our world modelby comparing it with DINO-WM Zhou et al. (2025) and domain randomization (DR) Tobin et al. (2017) for a navigation task (PointMaze) and a manipulation task (PushT) under different visual conditions, and pretrained visual encoders. Further details on the task setup are provided in Appendix A.4.

Evaluation Metric. We evaluate each condition using 50 independently sampled initial and goal pairs and define success rate (SR) as the fraction of successful episodes. We report the absolute SR under the native background (NC) and the relative degradation under each shifted condition s, defined as $\Delta_s = 100 \times (1 - \mathrm{SR}_s/\mathrm{SR}_{\mathrm{NC}})$. Positive values indicate a percentage decrease relative to the same model's native-background performance, while negative values indicate improvement. We also report the mean relative degradation across the five shifted conditions (SC, C, LC, LCG, and D). Lower degradation is better.

4.1 BASELINE

We compare our model with two baselines: DINO-WM Zhou et al. (2025) and domain DR Tobin et al. (2017). Both baselines are evaluated under the same task, planning horizon, and evaluation metric previously described.

DINO-WM. We use the original DINO-WM architecture as proposed in Zhou et al. (2025). The model operates directly on pretrained DINOv2 patch embeddings with 196 patches, each of dimensionality 384, and learns latent dynamics without explicit invariance regularization. DINO-WM is trained with a batch size of 32.

4.2 BACKGROUND CHANGES

Domain Randomization (DR). We also train DINO-WM with domain randomization Tobin et al. (2017). In particular, we randomize the visual appearance of the task for 40% of the training dataset by applying background color changes and gradient-based background modifications, for the PointMaze task. To avoid introducing spurious temporal artifacts, the randomized background is kept consistent across trajectories. We refer the reader to Appendix A.7.

Our model adopts the same number of visual patches (i.e., 196) as DINO-WM, but introduce a bisimulation encoder that maps each patch to a 32-dimensional latent embedding (see Figure 3). This leads to a latent space that is 10× lower-dimensional than the 384-dimensional patch embeddings used by DINO-WM. Our model is trained with a batch size of 20. We refer the reader to Appendix A.2.4 for details on the model hyperparameters.



Figure 5: Visual distribution shifts used to evaluate robustness on **PointMaze** (top) and **PushT** (bottom). From left to right, the conditions are: **NC**: No Change, **SC**: Slight Change, **C**: Color Gradient, **LC**: Large Color Change, **LCG**: Large Color Gradient, and **D**: moving Distractors.

4.3 RESULTS

We consider different background changes and moving distractors (see Figure 5) that alter the visual appearance of the scene while preserving the underlying transition dynamics. These changes are applied only at test time, unless otherwise stated, and are designed to isolate the effect of slow features on the planning performance. The moving distractors (colored dots) move continuously within the maze but do not collide with or affect the agent.

Table 1: **PointMaze.** Absolute success rate under NC (↑) and relative degradation (%, ↓) under shifts. Negative values indicate improvement. Avg.: mean degradation over the five background shifts.

Model	NC (Abs.)	SC (Rel.%)	C (Rel.%)	LC (Rel. %)	LCG (Rel.%)	D (Rel.%)	Avg (Rel.%)
Comparison with DR							
DINO-WM	0.8	10.0	25.0	30.0	40.0	2.5	21.5
DINO-WM w/DR	0.82	0.0	0.0	17.1	22.0	0.0	7.8
Ours (DINOv2)	0.78	−2.6	2.6	−10.3	0.0	−5.1	**−3.1**
Encoder ablation							
End-to-End	0.68	35.3	−2.9	61.8	47.1	5.9	29.4
SimDINOv2	0.40	5.0	10.0	−5.0	−5.0	10.0	3.0
iBOT	0.72	2.8	−2.8	0.0	0.0	0.0	**0.0**

Table 1 reports the absolute success rate under NC and relative degradation under background changes and a moving distractor for PointMaze. Negative degradation indicates improvement over the same model's native-background performance.

Robustness to the background. DINO-WM degrades when the visual background differs from the one seen during training, with relative degradation ranging from 10.0% under SC to 40.0% under LCG. In contrast, our model largely preserves its native-background performance, with degradation ranging from -10.3% to 2.6% across background changes. These results are consistent with the bisimulation encoder reducing sensitivity to background-dependent features, supporting reliable planning despite visual changes.

Comparison with DR. Domain randomization preserves performance under SC and C, with 0.0% degradation in both conditions. However, degradation increases to 22.0% under LCG, an aggressive background change not covered by the randomized training distribution. Our model exhibits 0.0% degradation under this condition, retaining its native-background performance.

Moving distractor. Under the moving distractor, DINO-WM exhibits a small relative degradation of 2.5%, while domain randomization preserves its native-background performance. Our model achieves -5.1% degradation, corresponding to a small observed improvement over its NC baseline.

Table 2 reports absolute success rates under NC and relative degradation under visual shifts for PushT across different training data sizes.

Table 2: **PushT.** Absolute success rate under NC (↑) and relative degradation (%, ↓) under shifts for different training data sizes. Negative values indicate improvement.

Samples	Model	NC (Abs.)	SC (Rel.%)	C (Rel.%)	LC (Rel. %)	LCG (Rel.%)	D (Rel.%)	Avg (Rel.%)
1000	DINO-WM	0.48	21.0	21.0	71.0	50.0	75.0	47.6
1000	DINO-Bisim	0.36	0.0	0.0	11.0	11.0	17.0	**7.8**
5000	DINO-WM	0.74	8.0	14.0	43.0	24.0	70.0	31.8
5000	DINO-Bisim	0.44	−14.0	9.0	−5.0	14.0	77.0	**16.2**
Full	DINO-WM	0.78	15.0	13.0	36.0	26.0	74.0	32.8
Full	DINO-Bisim	0.44	0.0	18.0	14.0	5.0	41.0	**15.6**

DINO-Bisim achieves lower mean relative degradation across the five shifts than DINO-WM: 7.8%, 16.2%, and 15.6% with 1000 samples, 5000 samples, and the full dataset, respectively, compared to 47.6%, 31.8%, and 32.8%. The difference is particularly pronounced under aggressive background changes: with 1000 samples, DINO-Bisim exhibits 11.0% degradation under both LC and LCG, compared to 71.0% and 50.0% for DINO-WM. Moving distractors remain challenging, although DINO-Bisim shows lower degradation with 1000 samples (17.0% versus 75.0%) and the full dataset (41.0% versus 74.0%). With 5000 samples, however, its degradation is higher (77.0% versus 70.0%). These results indicate better retention of native-background performance on average, although DINO-Bisim's lower NC success rates mean that this does not imply higher absolute success across conditions.

4.4 On Different Pretrained Visual Encoders

Table 1 compares native-background success rates and relative degradation across visual encoders. End-to-end training (i.e., training a visual encoder from scratch) yields 29.4% mean degradation, compared to -3.1% with DINOv2, suggesting that pretrained features support robustness when combined with our bisimulation encoder. SimDINOv2 and iBOT also retain their native-background performance, with mean degradation of 3.0% and 0.0%, respectively. Thus, robustness extends across pretrained encoders, despite differences in absolute success rates.

5 Discussions and Future Work

We show that DINO-WM is not robust under background changes and distractors. By enforcing control-relevant invariance via bisimulation on top of pretrained visual features, we obtain latent embeddings $10\times$ lower-dimensional than that of DINO-WM that are robust for planning under visual shifts, without reward supervision.

Our work has two main limitations: it relies on informative pretrained visual features, as suggested by reduced robustness without pretraining (i.e., End-to-End in Table 1), and its reward-free bisimulation analysis yields a horizon-dependent bound that is looser than its reward-aware counterpart. Future work includes integrating the bisimulation encoder into gradient-based planning (Parthasarathy et al., 2025), adapting representations online to new visual conditions, extending the analysis to reward-aware bisimulation, and evaluating more complex manipulation and long-horizon tasks.

AI USE STATEMENT

We used generative AI tools only to edit and polish the manuscript for clarity, grammar, and readability. We did not use generative AI tools to develop the theoretical results, formulate mathematical claims, construct proofs, design the model architecture, process data, or interpret the results. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content of this work.

REFERENCES

Alessandro Abate. Approximation metrics based on probabilistic bisimulations for general state-space markov processes: a survey. *Electronic Notes in Theoretical Computer Science*, 297:3–25, 2013.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Pablo Castro and Doina Precup. Using bisimulation for policy transfer in mdps. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, pp. 1065–1070, 2010.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10069–10076, 2020.

Pablo Samuel Castro, Prakash Panangaden, and Doina Precup. Equivalence relations in fully and partially observable markov decision processes. In *IJCAI*, volume 9, pp. 1653–1658, 2009.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pp. 162–169, 2004.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Clark R Givens and Rae Michael Shortt. A class of wasserstein metrics for probability distributions. *Michigan Mathematical Journal*, 31(2):231–240, 1984.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

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

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*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Dongsheng Jiang, Yuchen Liu, Songlin Liu, Jin'e Zhao, Hao Zhang, Zhen Gao, Xiaopeng Zhang, Jin Li, and Hongkai Xiong. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023.

Mete Kemertas and Tristan Aumentado-Armstrong. Towards robust bisimulation metric learning. *Advances in Neural Information Processing Systems*, 34:4764–4777, 2021.

Basil Kouvaritakis and Mark Cannon. Model predictive control. *Switzerland: Springer International Publishing*, 38(13-56):7, 2016.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Ian Lenz, Ross A Knepper, and Ashutosh Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*, volume 10, pp. 25. Rome, Italy, 2015.

Jing Yu Lim, Zarif Ikram, Samson Yu, Haozhe Ma, Tze-Yun Leong, and Dianbo Liu. Jedi: Latent end-to-end diffusion mitigates agent-human performance asymmetry in model-based reinforcement learning. *arXiv preprint arXiv:2505.19698*, 2025.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

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

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations. *arXiv preprint arXiv:2402.15567*, 2024.

Arjun Parthasarathy, Nimit Kalra, Rohun Agrawal, Yann LeCun, Oumayma Bounou, Pavel Izmailov, and Micah Goldblum. Closing the train-test gap in world models for gradient-based planning. *arXiv preprint arXiv:2512.09929*, 2025.

Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2004.

Yutaka Shimizu and Masayoshi Tomizuka. Bisimulation metric for model predictive control. In *International Conference on Learning Representations*, 2025.

Vlad Sobal, Jyothir SV, Siddhartha Jalagam, Nicolas Carion, Kyunghyun Cho, and Yann LeCun. Joint embedding predictive architectures focus on slow features. *arXiv preprint arXiv:2211.10831*, 2022.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim GJ Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *arXiv preprint arXiv:2502.14819*, 2025.

Ruixiang Sun, Hongyu Zang, Xin Li, and Riashat Islam. Learning latent dynamic robust representations for world models. *arXiv preprint arXiv:2405.06263*, 2024.

Basile Terver, Randall Balestriero, Megi Dervishi, David Fan, Quentin Garrido, Tushar Nagarajan, Koustuv Sinha, Wancong Zhang, Mike Rabbat, Yann LeCun, et al. A lightweight library for energy-based joint-embedding predictive architectures. *arXiv preprint arXiv:2602.03604*, 2026.

Ilya Tkachev and Alessandro Abate. On infinite-horizon probabilistic properties and stochastic bisimulation functions. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 526–531. IEEE, 2011.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim GJ Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. *arXiv preprint arXiv:2603.12231*, 2026.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 1714–1721. IEEE, 2017.

Ziyang Wu, Jingyuan Zhang, Druv Pai, XuDong Wang, Chandan Singh, Jianwei Yang, Jianfeng Gao, and Yi Ma. Simplifying dino via coding rate regularization. *arXiv preprint arXiv:2502.10385*, 2025.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, 2025.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

Jiachen Zhu, Katrina Evtimova, Yubei Chen, Ravid Shwartz-Ziv, and Yann LeCun. Variance-covariance regularization improves representation learning. *arXiv preprint arXiv:2306.13292*, 2023.