Back to Parsimonious Latents: Learning Task-Centric World Models from Visual Foundations

Minghao Fu[1] **Fan Feng**[1] **Nicklas Hansen**[1] **Biwei Huang**[1]
[1]University of California, San Diego
{m9fu, f2feng, nihansen, bih007}@ucsd.edu

Abstract

World models enable agents to predict future dynamics conditioned on actions, making the choice of latent representation central to planning and control. Such representations are often either learned directly from pixels with limited semantic structure or inherited from frozen visual foundation models with excessive task-irrelevant detail, yielding state spaces that are poorly matched to downstream planning and control. This is especially challenging in reward-free offline settings, where the model must learn from fixed trajectories without reward supervision or online interaction. To address this, we propose TC-WM, a framework for turning foundation-model embeddings into compact, task-sufficient world representations. The key design is to treat the pretrained embedding space as a semantic scaffold rather than as the final state space: TC-WM linearly projects high-dimensional visual embeddings into a compact latent as the dynamic space, aligns a subspace with the agent's physical state via contrastive learning, and reconstructs embeddings to preserve useful visual structure. This combines the generality of foundation features with the controllability of task-centric dynamics. Theoretically, we show that under the stated population-level assumptions, TC-WM suffices to identify the underlying task-centric latent factors up to a simple transformation. Empirically, across nine simulated environments, TC-WM yields its clearest planning gains on contact-rich manipulation while preserving task-relevant visual representation. Fine-tuning Cosmos3-Nano with TC-WM further improves physical consistency on DROID, showing the objective scales to large video models and real-robot data.

Webpage: https://minghaofu.com/tc-wm/

1 Introduction

An agent that can simulate its environment can plan, generalise, and act efficiently from limited experience, a long-standing goal that motivates research on *world models* [Sutton, 1991, Ha and Schmidhuber, 2018, Hafner et al., 2020, Hansen et al., 2024]. Recent progress has crystallised the field into three architectural branches that differ in the level at which dynamics are predicted (Figure 1). ***Generative*** models (a) predict future observations directly in pixel space [Ha and Schmidhuber, 2018], providing rich visualisation but at heavy computational cost. ***Latent*** models (b) instead learn a compact latent representation and predict dynamics within it [Hafner et al., 2021, 2023, 2025, Hansen et al., 2022, 2024, Schrittwieser et al., 2020], with reward signals (*e.g.*, in TD-MPC and MuZero) often shaping the latent into a task-relevant representation. ***Embedding*** models (c), most recently exemplified by DINO-WM and V-JEPA [Assran et al., 2025, Zhou et al., 2025, Baldassarre et al., 2025], skip the representation-learning step by predicting directly in the frozen feature space of a pre-trained foundation model, leveraging its broad semantic priors for zero-shot generalisation. DINO-WM also conditions its predictions on proprioception.

Generally, pretrained visual embeddings are useful not because they form ideal state representations, but because they provide a strong semantic coordinate system from which such states can be

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

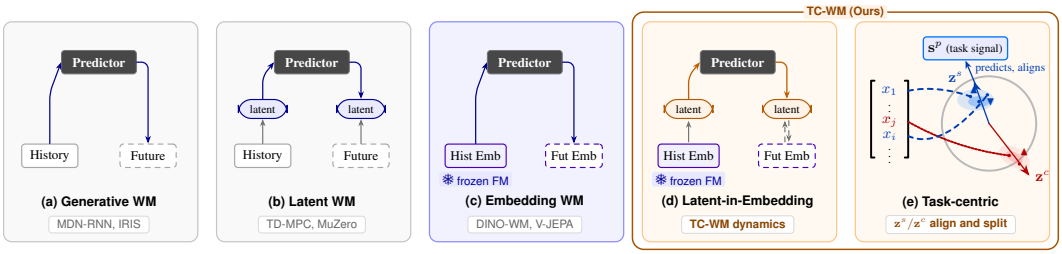


Figure 1: Comparison of world model paradigms (a)–(c) and our task-centric refinement (d)–(e). Detailed related work can be found in Section D.

extracted. This view is aligned with recent representation-alignment results in generative modelling (*e.g.*, REPA [Yu et al., 2025]), which shows that tying generative latents to frozen visual encoders simplifies high-dimensional generation. For world models, we therefore build dynamics *on top of* foundation embeddings rather than from pixels alone, inheriting object- and scene-level structure before learning temporal prediction.

The challenge is that foundation embeddings are optimised for broad semantic coverage rather than for world dynamics or specific planning tasks [Hafner et al., 2020, Hu et al., 2023, Zhang et al., 2025]: they encode textures, lighting, and background details that do not affect control. A robot navigating a room needs spatial layout and movable-object dynamics, not wall colour or sofa texture. When generic embeddings serve directly as world-model latents, the model spends capacity on irrelevant variation, hurting modelling efficiency and control interpretability [Schneider et al., 2024]. This burden is most acute under *high-dimensional action spaces*: `Maze` or `Push-T` use only 2-D actions, but contact-rich `Robomimic` requires a 7-DoF arm with a 43-D proprioceptive state (Table A1), where every irrelevant latent dimension becomes a possible factor of representation-collapse and wasted planning capacity. The task-centric fix, pursued broadly in representation learning [Lesort et al., 2018, Schölkopf et al., 2021, Locatello et al., 2020, Ho et al., 2022] and implemented by reward-guided latents in, *e.g.*, TD-MPC2 and MuZero [Hansen et al., 2024, Schrittwieser et al., 2020], is to retain only the information needed to predict task-relevant futures.

We introduce TC-WM (Figure 1d–e), a task-centric world model designed around this decomposition. Instead of predicting pixels or rolling out the full foundation embedding, TC-WM treats the embedding as a reusable visual scaffold and learns compact latent dynamics above it. A designated subspace aligns with the environment's physical state, while the complementary subspace remains anchored to the pretrained embedding via reconstruction, preserving useful visual structure while filtering nuisance variation. This design yields an ***identifiability*** guarantee of task-relevant representation of world dynamics, validated by our linear probing and ablations.

Once the task-centric representation is ready, it supports *a wide suite of* downstream policy-learning control, including test-time planning with model predictive control [Draeger et al., 1995] for goal-reaching tasks, a latent diffusion planner [Xie et al., 2025] for manipulation tasks, and model-free reinforcement learning [Haarnoja et al., 2018] for locomotion tasks.

Empirically, TC-WM improves both world model quality, reward-free offline planning and control across 9 benchmarks; Figure 2 highlights the `Robomimic` setting with a high-dimensional action space, where the gains are largest, the learned latent linearly recovers physical states, and our design prevents representation collapse compared with models directly building on visual embeddings. We also examine task-centric alignment in Cosmos3-Nano on DROID robot trajectories.

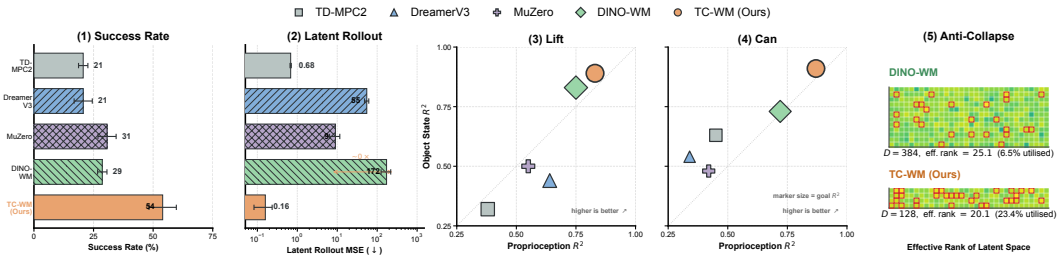


Figure 2: Experiment results on `Robomimic`. **(1)** Success rate. **(2)** Latent-rollout MSE. **(3)–(4)** Linear probes on `Lift` and `Can`. **(5)** Anti-collapse: TC-WM has better utilisation of latent space.

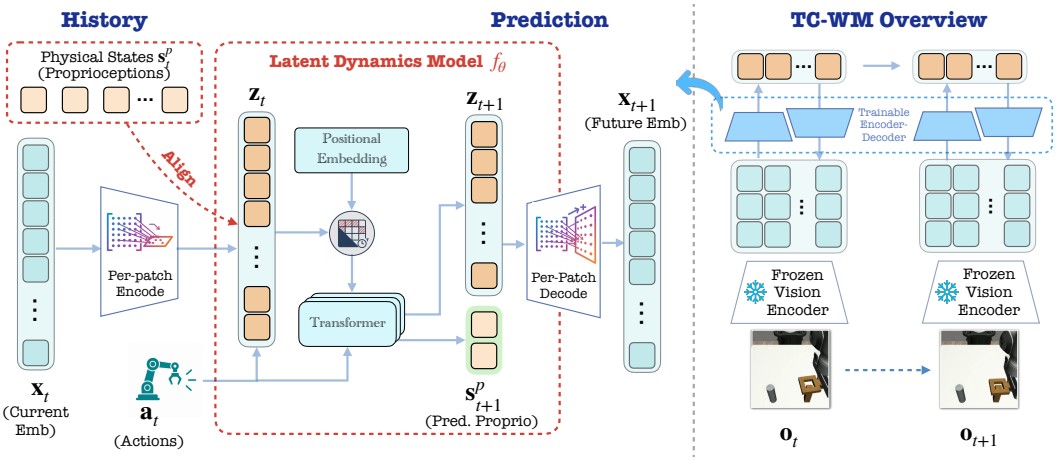


Figure 3: We present **TC-WM**, a method for training a world model by extracting a compact, task-centric latent dynamics from pretrained visual embeddings of image frames using the task-centric side information, *e.g.*, proprioception.

2 Method

2.1 Problem Setup

We formulate the problem as a partially observable Markov decision process (POMDP) [Kaelbling et al., 1998] defined by $(\mathcal{S}, \mathcal{O}, \mathcal{A}, p_{\text{env}})$, where $\mathcal{S}$ denotes the latent environment state space, $\mathcal{O}$ the observation space, and $\mathcal{A}$ a continuous action space. The environment evolves according to latent dynamics $p_{\text{env}}(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$, while observations $\mathbf{o}_t \in \mathcal{O}$ are generated from latent states $\mathbf{s}_t$ through an unknown observation process. We assume access to a low-dimensional *proprioceptive* signal $\mathbf{s}_t^p \in \mathbb{R}^d$, which captures the agent's physical configuration (*e.g.*, joint positions, end-effector pose, gripper state). This signal represents a partial, directly observable projection of the full state $\mathbf{s}_t$.

Given an offline dataset of visual trajectories $\mathcal{D} = \{(\mathbf{o}_t, \mathbf{a}_t, \mathbf{s}_t^p)\}_{t=0}^T$, where visual observations can be further encoded into embeddings $\mathbf{x}_t$, our goal is to learn a world model that infers a latent representation $\mathbf{z}_t$ from observation histories and models its dynamics. We aim for the learned latent sequence $[\mathbf{z}_0, \ldots, \mathbf{z}_T]$ to be *task-centric*, meaning that it preserves information necessary to predict future task-relevant dynamics under actions, which is formally defined below.

Definition 1 (Task-Centric Representation)**.** *A latent representation $\mathbf{z}_t = f(\mathbf{o}_{\leq t}, \mathbf{a}_{<t})$ is* task-centric *with respect to proprioceptive state $\mathbf{s}_t^p$ if it satisfies the following two conditions.*

$$\underbrace{p_{env}(\mathbf{s}^p_{t+H} \mid \mathbf{o}_{\leq t}, \mathbf{a}_{<t}, \mathbf{a}_{t:t+H-1}) = p_{env}(\mathbf{s}^p_{t+H} \mid \mathbf{z}_t, \mathbf{a}_{t:t+H-1})}_{\textbf{\textit{Sufficiency:} for any action sequence replacing history by } \mathbf{z}_t \textit{ leaves the future unchanged}} \qquad \underbrace{p_{env}(\mathbf{s}^p_t \mid \mathbf{z}_t) = p_{env}(\mathbf{s}^p_t \mid \mathbf{o}_{\leq t}, \mathbf{a}_{<t})}_{\textbf{\textit{Task-Aligned:} preserve the current proprioceptive state}}$$

2.2 Task-Centric World Model (TC-WM)

Overview. TC-WM learns a compact rollout state $\mathbf{z}_t$ on top of frozen visual embeddings, using proprioception to identify the dimensions needed for physical dynamics. Figure 3 illustrates the pipeline.

▷ **Joint Embedding.** At each time step, a frozen encoder f^o_{emb}, such as DINOv2 [Oquab et al., 2023], maps the image $\mathbf{o}_t$ to a visual feature, while a trainable linear layer f^p_{emb} maps proprioception $\mathbf{s}_t^p$. Their concatenation $\mathbf{x}_t = [f^o_{\text{emb}}(\mathbf{o}_t), f^p_{\text{emb}}(\mathbf{s}_t^p)]$ forms the input to the latent encoder, which produces the rollout state $\mathbf{z}_t$. Frozen visual features contribute spatially coherent, object-centric cues [Zhou et al., 2025, Baldassarre et al., 2025, Yu et al., 2025], and proprioception supplies the physical coordinate injection, which serves as side information.

▷ **Latent Encoder.** To further distill the visual embeddings, TC-WM treats them as a scaffold rather than directly modeling temporal dynamics within them. Specifically, we introduce an encoder

$q(\mathbf{z}_t \mid \mathbf{x}_t)$ that transforms the joint embedding $\mathbf{x}_t$ into a latent representation $\mathbf{z}_t$, which is further trained to satisfy Definition 1. This projection is the central abstraction step: it retains the semantic directions useful for reconstructing and predicting the scene, but compresses away dimensions that are not useful for future physical state prediction.

▷ **TC Alignment.** Directly aligning the full latent space with $\mathbf{s}^p$ would discard the rich semantic content captured by the visual embedding. We therefore perform a *partial* alignment: we factorise $\mathbf{z}_t = [\mathbf{z}_t^s, \mathbf{z}_t^c]$ and align only the state-relevant block $\mathbf{z}_t^s$ with the proprioceptive state $\mathbf{s}_t^p$ via an InfoNCE-style contrastive objective [Oord et al., 2018], leaving the complementary block $\mathbf{z}_t^c$ free to absorb residual semantic structure (object appearance, background, etc.) so that the embedding-reconstruction loss introduced below can still be satisfied. Alignment concentrates physical information in $\mathbf{z}^s$, while $\mathbf{z}^c$ can retain information useful for dynamics and reconstruction.

$$\mathcal{L}_{\text{align}} = -\log \frac{\exp(\text{sim}(g_\phi(\mathbf{z}_t^s), h_\psi(\mathbf{s}_t^p))/\tau)}{\sum_{\mathbf{s}_t^-} \exp(\text{sim}(g_\phi(\mathbf{z}_t^s), h_\psi(\mathbf{s}_t^{p-}))/\tau)}, \tag{1}$$

where g_ϕ, h_ψ are small projection heads, $\text{sim}(\cdot,\cdot)$ is cosine similarity, τ is the temperature, positive pairs $(\mathbf{z}_t^s, \mathbf{s}_t^p)$ are drawn from the same timestep, and negatives $\mathbf{s}_t^{p-}$ from other samples in the batch. In practice, g_ϕ is implemented as a *linear* map $\mathbb{R}^{d_z} \to \mathbb{R}^{d_s}$ with an ℓ_1 penalty on its weight matrix, so $\mathbf{z}_t^s$ is automatically *selected* by sparsity rather than pre-sliced from $\mathbf{z}_t$; d_s functions only as an upper bound on the width of the task-centric block.

▷ **Latent Dynamics.** We adopt the ViT [Dosovitskiy, 2020] as the transition architecture f_θ to predict $\hat{\mathbf{z}}_{t+1} = f_\theta(\mathbf{z}_{t-H:t}, \mathbf{a}_{t-H:t})$, which models the temporal dependencies of states and actions by minimizing a predictive consistency loss

$$\mathcal{L}_{\text{dyn}}^z = \|\hat{\mathbf{z}}_{t+1} - q(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1})\|_2^2$$

which encourages the transition to capture dynamics in the projected latent space rather than at the raw embedding level.

▷ **TC Dynamics.** Beyond instantaneous alignment, we predict these proprioceptive variables directly from the history to encourage the latent space to capture task-centric dynamics:

$$\mathcal{L}_{\text{dyn}}^s = \|f_\eta^p(\mathbf{z}_{t-H:t}, \mathbf{a}_{t-H:t}) - \mathbf{s}_{t+1}^p\|_2^2.$$

We predict raw (but standardised) proprioception rather than its embedding, which avoids trivial solutions where the state-embedding layer f_{emb}^p collapses to match its own output.

▷ **Embedding Decoder.** A trainable linear decoder f_{dec} reconstructs the visual embedding $\mathbf{x}_t$ from $\mathbf{z}_t$. This term acts as a *regulariser* that anchors the compact latent to the foundation-embedding manifold, keeping the latent space smooth, well-spread across samples, and free of the degenerate solutions that compact latents are otherwise prone to:

$$\hat{\mathbf{x}}_t = f_{\text{dec}}(\mathbf{z}_t), \qquad \mathcal{L}_{\text{rec}} = \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2. \tag{2}$$

Reconstruction provides visual supervision in generative world models [Hafner et al., 2020, 2023, Alonso et al., 2024]. Reconstructing foundation embeddings, such as DINO [Oquab et al., 2023] or V-JEPA [Assran et al., 2023], preserves visual structure while regularizing the compact latent (Section 3.3).

Visual Decoder. For visualization *only*, an auxiliary decoder maps predicted embeddings to images, $\hat{\mathbf{o}}_t = f_{\text{vis}}(\hat{\mathbf{x}}_t)$, with $\mathcal{L}_{\text{img}} = \|\hat{\mathbf{o}}_t - \mathbf{o}_t\|_2^2$ separated from world-model optimization.

The overall training objective combines dynamics prediction and representation alignment:

$$\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{dyn}}^z + \mathcal{L}_{\text{dyn}}^s + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}},$$

where λ_{align} balances alignment and dynamics, and λ_{rec} governs the embedding reconstructions.

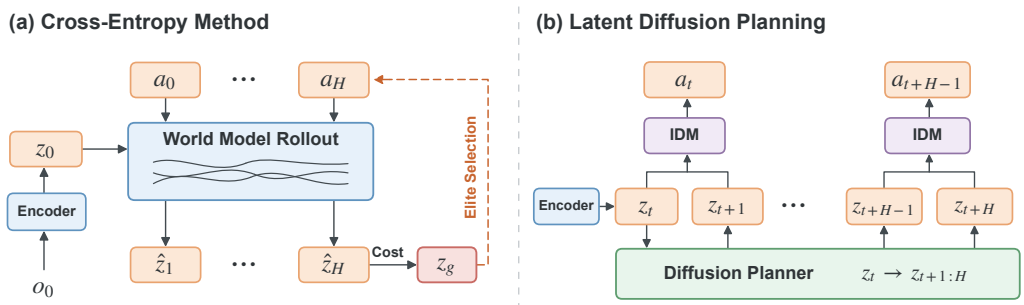


Figure 4: Planning algorithms used in TC-WM. **(a) Cross-Entropy Method (CEM)** for navigation: the world model rolls out K sampled action sequences, scores each rollout against a goal latent $\mathbf{z}_g$ via cost $\mathcal{C}$, and refines the action distribution by elite selection. **(b) LDP planning** for manipulation: a diffusion planner generates future latent trajectories $[\mathbf{z}_{t+1}, \ldots, \mathbf{z}_{t+H}]$, and an inverse dynamics model (IDM) recovers executable actions from consecutive latent pairs.

2.3 Test-Time Planning and Control with TC-WM

To validate whether the task-centric representations can be translated into better downstream planning, we evaluate the learned world model on visual goal-reaching tasks performed entirely in the latent space $\mathcal{Z}$. Given an initial observation $\mathbf{o}_0$ and a goal observation $\mathbf{o}_g$, we encode them as $\mathbf{z}_0 = q(\mathbf{z}_0 \mid \mathbf{x}_0)$ and $\mathbf{z}_g = q(\mathbf{z}_g \mid \mathbf{x}_g)$. Planning is then formulated as minimizing a latent-space objective that measures the distance between the predicted latent state and the goal state:

$$\mathcal{L}_{\text{plan}}(\mathbf{a}_{0:H_p-1}) = \| f_\theta^{H_p}(\mathbf{z}_0, \mathbf{a}_{0:H_p-1}) - \mathbf{z}_g \|_2^2.$$

Here H_p is the number of actions and predicted transitions. The operator $f_\theta^{H_p}$ rolls out the full latent $[\mathbf{z}^s, \mathbf{z}^c]$ autoregressively, with history initialization defined in Appendix A.7. Because a *goal-reaching* task provides a target observation, and hence a goal latent $\mathbf{z}_g$ that defines a planning cost, we adopt model predictive control (MPC) [Draeger et al., 1995] with the cross-entropy method (CEM): its sampling-based search is training-free and stays tractable in these low-dimensional action spaces, iteratively refining action sequences toward $\mathbf{z}_g$ based on predicted rollouts by $\mathcal{L}_{\text{plan}}(\mathbf{a}_{0:H_p-1})$ (Figure 4a). For high-dimensional, contact-rich *manipulation*, the action distribution is multimodal and scales poorly for sampling, so we instead learn latent diffusion planning (LDP) [Xie et al., 2025] over TC-WM's task-centric latents (Figure 4b). LDP trains a diffusion planner for future latent trajectories and an inverse dynamics model (IDM) for executable actions:

$$\mathcal{L}_{\text{LDP}}(\psi) = \mathbb{E}_{\tau,\epsilon}\left[\|\epsilon_\psi(\hat{\mathbf{z}}_{t+1:t+H_p}; \mathbf{z}_t, \tau) - \epsilon\|_2^2\right], \quad \mathcal{L}_{\text{IDM}}(\xi) = \mathbb{E}_{i,\tau,\epsilon}\left[\|\epsilon_\xi(\hat{\mathbf{a}}_{t+i}; \hat{\mathbf{z}}_{t+i}, \hat{\mathbf{z}}_{t+i+1}, \tau) - \epsilon\|_2^2\right].$$

At inference, the planner samples $\hat{\mathbf{z}}_{t+1:t+H_p}$ by denoising conditioned on $\mathbf{z}_t$, and the IDM decodes actions from consecutive latent pairs. Compared with the original VAE-based LDP, replacing the encoder with TC-WM's latent space gives a more structured representation for planning; details are in Appendix A.3. For *locomotion*, where success is long-horizon return rather than reaching a fixed state and no goal latent is available to plan toward, we couple TC-WM with an off-policy actor–critic that directly maximizes return. The resulting latent $\mathbf{z}_t$ obtained by TC-WM is treated as the observation of a Soft Actor–Critic (SAC) agent [Haarnoja et al., 2018], which is then trained from environment interaction with the same protocol used by the baselines. We report the mean episode return across 3 seeds.

Theoretical Rationale. The following population-level theorem formalizes, under the stated assumptions, why a compact latent learned on frozen embeddings, once aligned with physical signals, can recover the task-relevant state needed for prediction and planning. We write $L_{\mathbf{v}|\mathbf{u}}$ for the conditional linear operator induced by $p(\mathbf{v} \mid \mathbf{u})$, where $[L_{\mathbf{v}|\mathbf{u}}\phi](\mathbf{v}) = \int p(\mathbf{v} \mid \mathbf{u})\phi(\mathbf{u})\, d\mathbf{u}$. Injectivity of L means that distinct latent states induce distinguishable futures, i.e., the observed dynamics vary across latent states.

Theorem 1 ***(Task-centric identifiability from visual embeddings)***

Suppose images $\mathbf{o}_t$, frozen embeddings $\mathbf{x}_t$, and latent variables $\mathbf{z}_t = [\mathbf{z}_t^s, \mathbf{z}_t^c]$ follow the process in Section 2.2, and the learned latent world model matches the true joint distribution of adjacent embeddings $\{\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}\}$. Assume:

A1. ***(Contextual dynamics):*** *$L_{\mathbf{x}_{t+1:t+H}|\mathbf{z}_t}$ and $L_{\mathbf{x}_{t-H:t-1}|\mathbf{x}_{t+1:t+H}}$ are injective and bounded.*

A2. ***(Latent–observed variability):*** *For any $\mathbf{z}_t^{(1)} \neq \mathbf{z}_t^{(2)}$, $p(\mathbf{x}_t \mid \mathbf{z}_t^{(1)}) \neq p(\mathbf{x}_t \mid \mathbf{z}_t^{(2)})$.*

A3. ***(Differentiability):*** *There exists a differentiable F such that $F[p_{\mathbf{x}_t|\mathbf{z}_t}(\cdot \mid \mathbf{z}_t)] = h_z(\mathbf{z}_t)$.*

A4. ***(Partial alignment):*** *The alignment objective in Eq. (1) is sufficiently minimized.*

Then the learned $\mathbf{z}_t$ is block-wise identifiable, and the task-centric $\mathbf{z}_t^s$ is affine-identifiable:

$$\underbrace{\hat{\mathbf{z}}_t = h_z(\mathbf{z}_t), \quad h_z \text{ invertible and } C^1}_{\textbf{Latent Space Recovery: the compact latent preserves the true world state up to reparameterization}} \qquad \underbrace{\mathbf{z}_t^s = \mathbf{A}\hat{\mathbf{z}}_t^s + \mathbf{b}}_{\textbf{Task-Centric Recovery: the aligned block recovers physical factors up to an affine map}}$$

Proof sketch and intuition. The observable joint distribution over neighboring embeddings determines an operator family that is a similarity transform of the diagonal operator indexed by $p(\mathbf{x}_t \mid \mathbf{z}_t)$. Injective context operators make this diagonalization unique up to a smooth reparameterization, and the alignment loss selects the task-centric block by tying it to physical variables. Under these assumptions, the result identifies the full latent up to a smooth reparameterization and the aligned block up to an affine map (Appendix B). The affine-recovery argument uses a sharp-temperature population limit. Training uses finite data and $\tau = 0.1$.

3 Experiments

We evaluate TC-WM across a diverse set of offline control and visual planning benchmarks to assess whether the learned representations are *task-centric*, *predictively sufficient*, and effective for downstream decision-making. Our evaluation spans navigation, locomotion, and manipulation tasks across simulated and real-world settings. We focus on three complementary aspects: `(i)` the accuracy and stability of the learned world model, `(ii)` planning performance under both low- and high-dimensional action spaces, and `(iii)` the interpretability and structure of the learned latent representations through targeted ablations and analysis.

3.1 Environments and Tasks

We evaluate TC-WM on 9 offline visual-control environments spanning *navigation*, *locomotion*, and *manipulation* in different metrics.

Datasets. As shown in Figure 5, the benchmark suite includes `Maze` and `Wall` for navigation, `Push-T` and `RoboMimic` Mandlekar et al. [2021] (`Lift`, `Can`, `Square`) for manipulation, and `Reacher`, `Cheetah`, and `Hopper` from DeepMind Control Suite [Tassa et al., 2018] for continuous control. Dataset construction, action/state dimensions, and preprocessing details are provided in Appendix A.1.

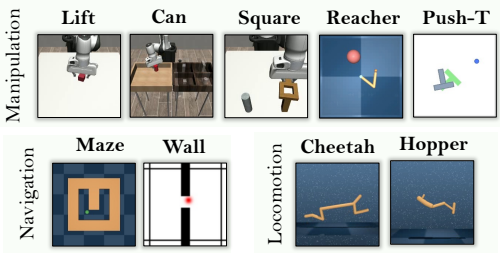


Figure 5: **Environments used in Experiments**.

Evaluation. We evaluate prediction quality using image-space reconstruction, latent rollout error, and SSIM. For goal-reaching tasks, including `Reacher`, we use CEM and report success rate (SR). For `Cheetah` and `Hopper`, we use SAC and report episode return.

Choices of Visual Foundation Models. We use DINOv2 [Oquab et al., 2023] as the default frozen visual encoder, matching DINO-WM for fair comparison. Figure 9 additionally evaluates

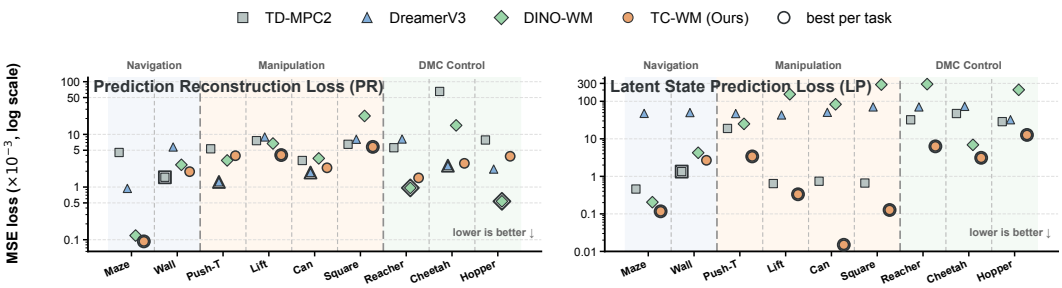


Figure 6: World model prediction accuracy. We report Prediction Reconstruction Loss (PR) and Latent State Prediction Loss (LP) (lower is better) on offline datasets over 300k training steps.

DINOv3 [Siméoni et al., 2025] and Cosmos [Agarwal et al., 2025] as alternative visual foundation models; implementation details are in Appendix A.2.

Baselines. We compare against four representative world-model baselines. **TD-MPC2** [Hansen et al., 2024] learns compact latent dynamics for model-predictive control, adapted here to replay-only offline training. **DreamerV3** [Hafner et al., 2025] learns a recurrent latent generative world model with image reconstruction and imagination. **MuZero** [Schrittwieser et al., 2020], a representative task-oriented latent-dynamics method, is reproduced inside our framework as a representation/dynamics trained on the same offline trajectories (Section A.4). **DINO-WM** [Zhou et al., 2025] predicts future frozen DINO embeddings conditioned on actions and proprioception and is the closest embedding-space baseline. All methods use the same offline trajectories of images, actions, and proprioception.

3.2 Results and Analysis

World Model Prediction. Figure 6 separates image reconstruction from latent rollout accuracy. Image-space error is task-dependent: TC-WM is competitive on `Lift`, `Square`, and `Maze`, while baselines perform better on visually complex tasks like `Push-T`, `Cheetah`, and `Hopper`. In contrast, TC-WM achieves the lowest latent prediction error on nearly all tasks, especially in manipulation settings where accurate dynamics are crucial. This suggests that compressing frozen visual embeddings preserves key structure while yielding a more stable state space.

Open-loop rollouts in Figure 7 provide the qualitative counterpart: TC-WM better preserves contact events, gripper motion, and object placement over long horizons. The main tradeoff is image-space fidelity on visually complex dynamics, where compact projection can discard low-level details.

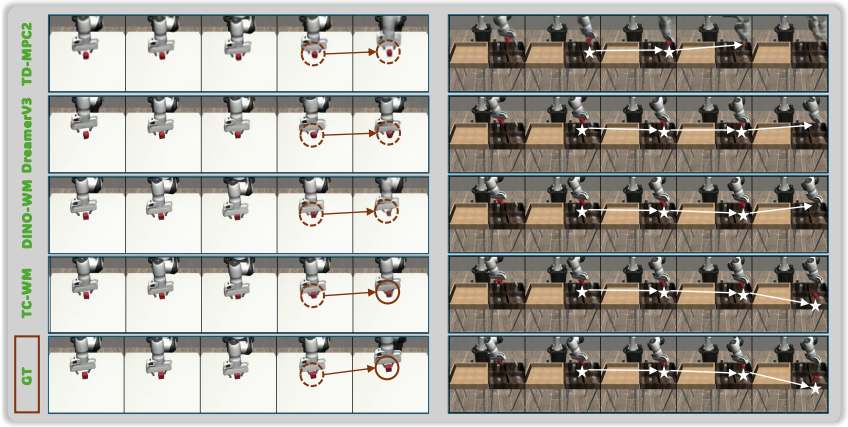


Figure 7: Open-loop rollouts on `RoboMimic`: `Lift` (left half) and `Can` (right half).

Planning Performance. Figure 8 evaluates TC-WM with the controller matched to each task: CEM for goal-reaching, SAC for locomotion, and LDP [Xie et al., 2025] for high-dimensional manipulation. TC-WM is competitive across the goal-reaching and locomotion tasks. The largest planning gains occur on the contact-rich manipulation tasks `Lift`, `Can`, and `Square`.

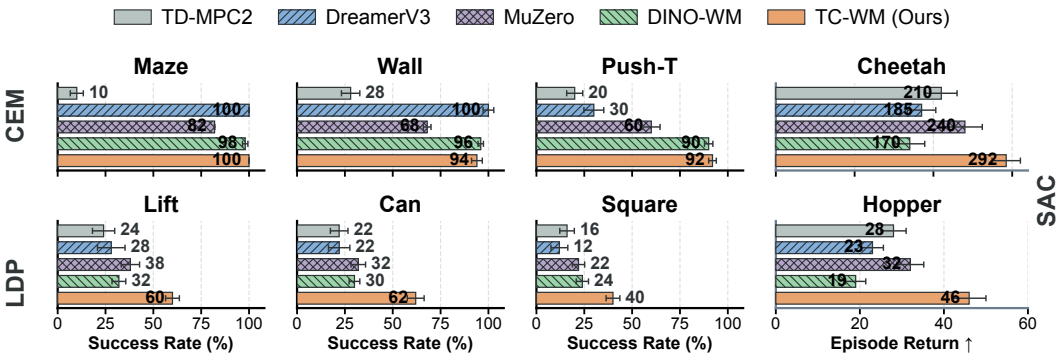


Figure 8: Planning and control performance of TC-WM against TD-MPC2, DreamerV3, MuZero, and DINO-WM, with the controller matched to each task's structure. *(Left, top)* CEM goal-reaching on `Maze`, `Wall`, `Push-T` (success rate, %); *(left, bottom)* LDP manipulation on `Lift`, `Can`, `Square`. *(Right column)* SAC locomotion on `Cheetah` and `Hopper` (episode return; the blue-grey background flags the change of metric). TC-WM matches the strongest baseline on the saturated goal-reaching tasks and is the only method that surpasses DINO-WM on every LDP task.

Architecture and Module Ablations. Figure 9 summarizes architecture-level ablations on `RoboMimic`. Random projection performs poorly, showing that dimensionality reduction alone is not enough. DINOv2 and DINOv3 provide the strongest foundation features, while Cosmos is weaker in this setting. Among projection heads, the linear map gives the best balance between planning success and embedding reconstruction.

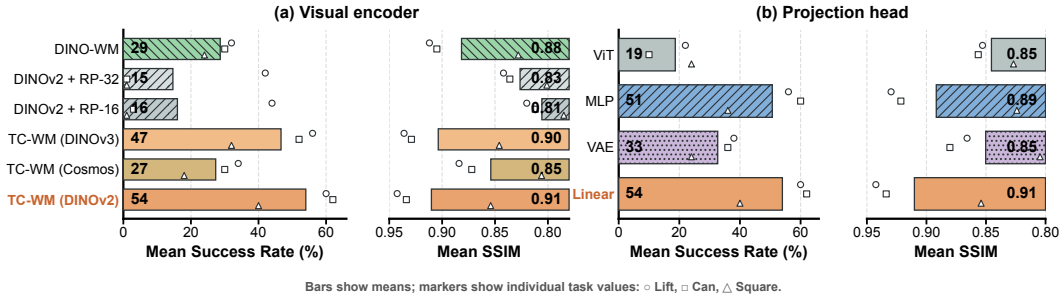


Figure 9: Architecture/module ablations on `RoboMimic`. *(a)* Visual encoder or latent-source choice and *(b)* projection-head ablation, each showing success rate on the left and mirrored SSIM on the right. RP denotes randomly projected DINOv2 embeddings. Bars show means over `Lift`, `Can`, and `Square`; markers show individual task values.

Robustness to unseen visual perturbations. Following the test-time perturbation protocol of DINO-WM [Zhou et al., 2025], we feed each frozen world model inputs corrupted by *Gaussian noise* (σ=0.1) or *color jitter* (per-channel scale $\in [0.8, 1.2]$, shift σ=0.05). Figure 10 shows TC-WM recovers the manipulator and object pose under both perturbations, and its relative SSIM drop is consistently smaller than DINO-WM's, *e.g.*, -2.5% vs. -4.8% under noise on `Lift`.

3.3 Evaluation on Learned Representations

Linear Probing on Latent Representation. To validate that TC-WM learns task-centric representations, we probe the learned latent space using ground-truth *full* states, including proprioceptive signals, object states, and goal-related indicators. We use the same protocol for all methods: train a linear regressor from each method's latent to the target state variables and report R^2 scores. The linear-probing panels in Figure 2 show that TC-WM more accurately recovers these task-centric variables. The learned latent retains linearly accessible proprioceptive, object, and goal information.

Projection Design Rationale. Figure 9 also explains why we use a linear projection. Frozen visual embeddings already contain structured semantic directions, so the projection mainly needs to select and align a compact task-centric subspace rather than relearn visual abstraction. More complex

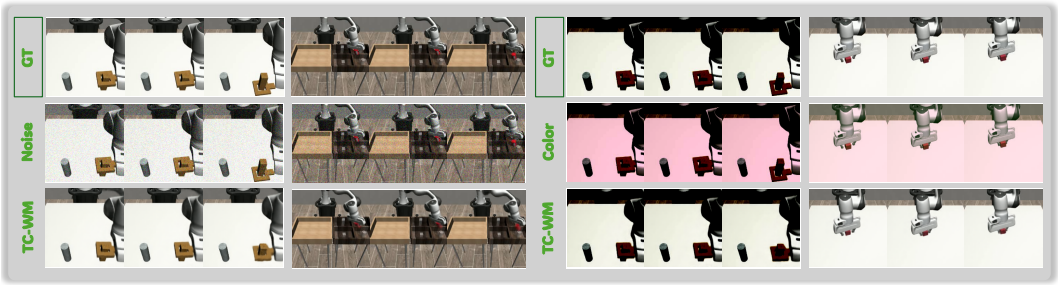


Figure 10: Rollouts under unseen visual perturbations. *Left:* Gaussian noise on `Square` and `Can`. *Right:* color jitter on `Can` and `Lift`.

projectors can distort this geometry or introduce optimization and bottleneck effects. This empirical trend is consistent with Theorem 1, which shows that a linear transformation is sufficient to recover the task-centric subspace up to an affine mapping.

Ablation on Loss Components. Figure 11 isolates the role of each loss on `Lift`. Removing embedding reconstruction causes the largest collapse, reducing both planning success and SSIM, which confirms that $\mathcal{L}_{\text{rec}}$ anchors the compact latent to the foundation embedding space. Removing proprioceptive supervision mostly preserves SSIM but lowers SR, showing that alignment shapes the latent for control rather than visual fidelity. The $\mathbf{z}^s/\mathbf{z}^c$ split dimension has only a modest effect, suggesting that performance is not sensitive to this hyperparameter.

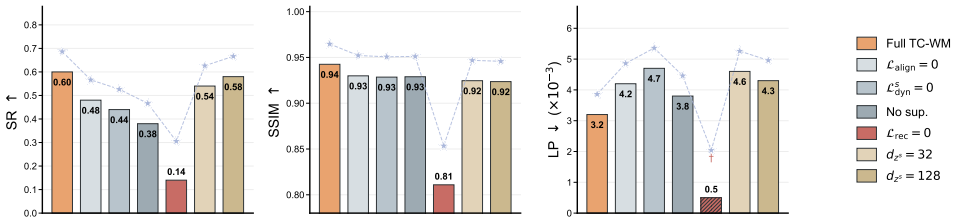


Figure 11: Ablation on loss components and latent split dimension (`Lift`, 100 epochs). $\mathcal{L}_{\text{rec}}$=0 collapses both planning (SR) and visual fidelity (SSIM); proprioceptive supervision ($\mathcal{L}_{\text{align}}$, $\mathcal{L}^s_{\text{dyn}}$) is essential for planning; $\mathbf{z}^s$ split dimension has modest effect. † Without embedding reconstruction, the latent collapses and trivially predicts itself, so LP is artificially low.

3.4 Task-Centric Alignment in a Video Foundation Model

To extend task-centric alignment beyond the DINO-based model, we adapt Cosmos3-Nano [NVIDIA, 2026] on DROID robot trajectories [Khazatsky et al., 2024]. We apply the same partial physical alignment used in TC-WM: a sparse head aligns a selected hidden-state component with synchronized proprioception while retaining the backbone's action-conditioned video-prediction objective. Figure 12 compares the resulting TC-Cosmos3 with Cosmos3-Nano under matched initial frames and recorded actions. In the illustrated manipulation sequences, the rollouts track gripper–object configurations across the wrist and exterior views as the scene evolves. These examples show how partial alignment can be used with a large video foundation model and real-robot data.

4 Conclusion

In this paper, we propose TC-WM, a task-sufficient world modeling framework that abstracts visual foundation embeddings into compact, task-centric latent spaces for accurate world modeling, planning, and control. By learning dynamics in this latent space rather than directly in high-dimensional embeddings, TC-WM improves predictive accuracy and planning performance in reward-free offline settings. Our results highlight the importance of projecting visual foundation embeddings into a task-centric subspace for world modeling and effective control, rather than using them verbatim.

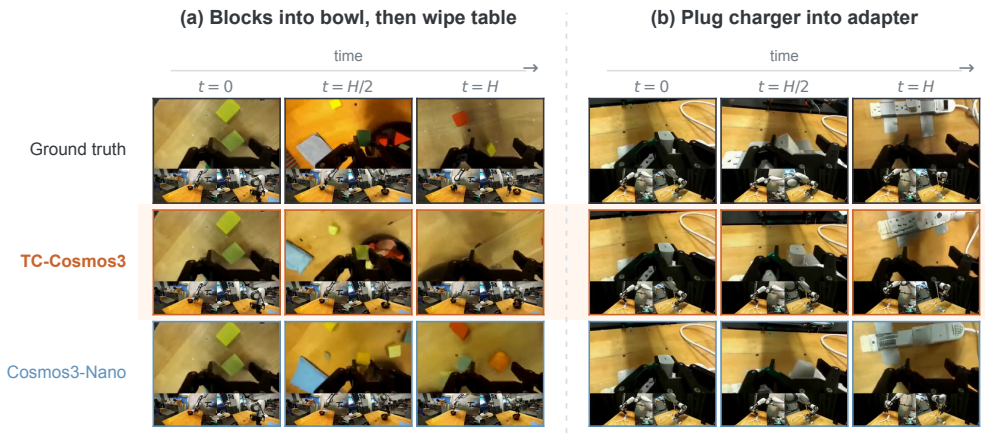


Figure 12: Qualitative action-conditioned rollouts on DROID robot trajectories. Rows show ground truth, TC-Cosmos3 (Cosmos3-Nano with task-centric partial alignment), and the original Cosmos3-Nano. Columns show the common initial frame, midpoint, and final frame; both models receive the same recorded actions. Each image combines a wrist view (top) and two exterior views (bottom).

Limitations and Future Work. TC-WM requires a training-time signal that exposes physical information (here proprioception s_t^p); when unavailable, distilled physical proxies or multi-view supervision are natural substitutes. Two follow-up directions we find promising are using TC-WM as a lightweight downstream-alignment module on top of a larger pretrained generative world model, and training a generative dynamics model directly on the foundation embedding space to combine pixel-free supervision with control-centric structure. Details are in Section E.

Acknowledgments and Disclosures

The authors thank the anonymous reviewers for their helpful comments and suggestions during the reviewing process. The authors acknowledge support from NSF DMS-2428058. The authors declare no competing interests.

References

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

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*, pages 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Alessandro Astolfi. Model reduction by moment matching for linear and nonlinear systems. *IEEE Transactions on Automatic Control*, 55(10):2321–2336, 2010.

Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: Dino as a foundation for video world models. *arXiv preprint arXiv:2507.19468*, 2025.

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

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

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

Andreas Draeger, Sebastian Engell, and Horst Ranke. Model predictive control using neural networks. *IEEE Control Systems Magazine*, 15(5):61–66, 1995.

Fan Feng, Selena Ge, Minghao Fu, Zijian Li, Yujia Zheng, Zeyu Tang, Yingyao Hu, Biwei Huang, and Kun Zhang. Ada-diffuser: Latent-aware adaptive diffusion for decision-making. In *The Fourteenth International Conference on Learning Representations*, 2026.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2786–2793. IEEE, 2017.

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.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*, pages 2424–2433. PMLR, 2019.

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):440, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

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

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

N Hansen, X Wang, and H Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning, PMLR*, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Learning massively multitask world models for continuous control. *arXiv preprint arXiv:2511.19584*, 2025.

Mark K Ho, David Abel, Carlos G Correa, Michael L Littman, Jonathan D Cohen, and Thomas L Griffiths. People construct simplified mental representations to plan. *Nature*, 606(7912):129–136, 2022.

Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, et al. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pages 2207–2217. PMLR, 2020.

Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Ankur Gupta, Silvio Savarese, and Li Fei-Fei. RoboTurk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 610–629, 2018.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

Francesca Mastrogiuseppe and Srdjan Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609–623, 2018.

Aran Nayebi, Rishi Rajalingham, Mehrdad Jazayeri, and Guangyu Robert Yang. Neural foundations of mental simulation: Future prediction of latent representations on dynamic scenes. *Advances in Neural Information Processing Systems*, 36:70548–70561, 2023.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

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

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.

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

Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2050–2053, 2018.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

Rishi Rajalingham, Aida Piccato, and Mehrdad Jazayeri. Recurrent neural networks with explicit representation of dynamic latent variables can mimic behavioral patterns in a physical inference task. *Nature Communications*, 13(1):5865, 2022.

Geoffrey Roeder, Luke Metz, and Durk Kingma. On linear identifiability of learned representations. In *International Conference on Machine Learning*, pages 9030–9039. PMLR, 2021.

Moritz Schneider, Robert Krug, Narunas Vaskevicius, Luigi Palmieri, and Joschka Boedecker. The surprising ineffectiveness of pre-trained visual representations for model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 37:32916–32946, 2024.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

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

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

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990.

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

Jinzhou Tang, Fan Feng, Minghao Fu, Wenjun Lin, Biwei Huang, and Keze Wang. Dreamsac: Learning hamiltonian world models via symmetry exploration. *arXiv preprint arXiv:2603.07545*, 2026.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.

Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.

Yilin Wu, Anqi Li, Tucker Hermans, Fabio Ramos, Andrea Bajcsy, and Claudia P'erez-D'Arpino. Do what you say: Steering vision-language-action models via runtime reasoning-action alignment verification. *arXiv preprint arXiv:2510.16281*, 2025.

Amber Xie, Oleh Rybkin, Dorsa Sadigh, and Chelsea Finn. Latent diffusion planning for imitation learning. *arXiv preprint arXiv:2504.16925*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M Patel, Paul Pu Liang, et al. World-in-world: World models in a closed-loop world. *arXiv preprint arXiv:2510.18135*, 2025.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=D5RNACOZEI.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International conference on machine learning*, pages 12979–12990. PMLR, 2021.

Appendix of "Back to Parsimonious Latents: Learning Task-Centric World Models from Visual Foundations"

A Experiment Details **16**

A.1 Environments and Dataset Collection 16

A.2 Details of Selected Visual Foundation Models 16

A.3 Pipeline of Planning on Manipulation Tasks 17

A.4 Implementation Details of Baselines 18

A.5 Training and Downstream Configurations 19

A.6 Compute Resources 19

A.7 Additional Controls and Robustness 19

B Theoretical Rationale: When and How can We Learn Task-centric Representation from Visual Foundations? **20**

B.1 Illustration and Analysis 21

B.2 Proof of Theorem A1 24

C More Visualizations and Analysis **29**

D Extended Related Work **31**

E Limitations and Future Work **33**

A Experiment Details

A.1 Environments and Dataset Collection

We evaluate across nine environments. Below we describe the data source for each group, organized by collection method.

DINO-WM released data (`Maze`, `Wall`, `Push-T`). We directly adopt the released datasets from DINO-WM [Zhou et al., 2025]. `Maze` follows the PointMaze environment of D4RL [Fu et al., 2020], where a force-actuated 2-DoF ball navigates to a goal; the dataset contains 2000 random trajectories. `Wall` is a 2D room-to-room navigation task with 1920 fixed-wall and 10240 randomized-wall trajectories. `Push-T`, adapted from Chi et al. [2025], uses 18.5k expert demonstrations and 20k randomized trajectories for a block-pushing task.

RoboMimic official demonstrations (`Lift`, `Can`, `Square`). We use the official demonstration datasets released by the RoboMimic benchmark [Mandlekar et al., 2021]. All three tasks are simulated in `robosuite` [Zhu et al., 2020] with MuJoCo, using a 7-DoF Franka Panda robot. Demonstrations are collected via the RoboTurk teleoperation platform [Mandlekar et al., 2018], where human operators control the robot arm through a smartphone interface. We use the Proficient-Human (PH) demonstrations, which are collected by a single skilled operator: 200 trajectories each for `Lift` (`robosuite Lift`), `Can` (`PickPlaceCan`), and `Square` (`NutAssemblySquare`). Image observations are rendered from two simulated cameras (a front-facing `agentview` and a wrist-mounted `robot0_eye_in_hand`).

TD-MPC2 rollouts (`Reacher`, `Cheetah`, `Hopper`). We generate offline trajectories by rolling out pretrained TD-MPC2 checkpoints from Hansen et al. [2025] in the DeepMind Control Suite. For each task we collect trajectories over multiple rounds, with actions produced by the TD-MPC2 policy conditioned on the current observation. We record rendered frames and low-dimensional state observations at each step. To ensure demonstration quality we apply a return-based filter: episodes are accepted only if their total return exceeds $0.75 \times$ median (or $1.25 \times$ median when negative).

Table A1: Action and proprioception dimensionalities of all nine evaluation environments, with the controller matched to each task's structure: CEM for goal-reaching (navigation, pushing, target reaching), SAC for locomotion, and LDP for high-dimensional, contact-rich `Robomimic` manipulation. The `Robomimic` tasks (`Lift`, `Can`, `Square`) have markedly higher-dimensional action and proprioceptive spaces, the regime where task-centric latents matter most.

Category	Dataset	Action dim	Proprio dim	Action type	Planner
Navigation	`Maze`	2	4	continuous force (F_x, F_y)	CEM
	`Wall`	2	4	continuous velocity (v_x, v_y)	CEM
Pushing	`Push-T`	2	5	continuous 2D target (t_x, t_y)	CEM
Manipulation	`Lift`	**7**	**43**	6 DoF joint velocity + gripper	LDP
	`Can`	**7**	**43**	6 DoF joint velocity + gripper	LDP
	`Square`	**7**	**43**	6 DoF joint velocity + gripper	LDP
DMC control	`Reacher`	2	6	continuous joint torque	CEM
	`Cheetah`	6	17	continuous joint torque	SAC
	`Hopper`	4	15	continuous joint torque	SAC

A.2 Details of Selected Visual Foundation Models

DINOv2. We use the TorchHub implementation from `facebookresearch/dinov2` with backbone `dinov2_vits14`. Given an input image tensor $x \in \mathbb{R}^{B\times 3\times H\times W}$, we extract patch-level features as the representation. This yields a sequence of normalized patch tokens of shape (B, N, D), where $D =$ `num_features` (384 for ViT-S/14) and $N = (H/\texttt{patch_size}) \cdot (W/\texttt{patch_size})$ with `patch_size` $= 14$, matching the configuration used by DINO-WM [Zhou et al., 2025].

DINOv3. We use the version `facebook/dinov3-vitb16-pretrain-lvd1689m`. The encoder takes RGB images $x \in \mathbb{R}^{B\times 3\times H\times W}$ as input, which are normalized using the model-provided

mean and standard deviation when available (otherwise ImageNet statistics). DINOv3 employs a ViT-B/16 backbone with patch size 16×16 and hidden dimension $D = 768$. The raw output has shape $(B, 1+R+N, D)$, where $N = (H/16)\cdot(W/16)$ denotes the number of patch tokens and R the number of register tokens. Following standard practice, we discard the CLS token and any register tokens, retaining only the patch tokens of the shape $\mathbb{R}^{B\times N\times 768}$ as the final visual representation.

Cosmos CI Tokenizer. For Cosmos, we use the `NVIDIA/Cosmos-0.1-Tokenizer-CI8x8` tokenizer encoder loaded from a TorchScript checkpoint. Images are resized to 112×112 and encoded in FP16. The encoder produces latents of shape (B, C, H', W') (with $C = 16$ in our setting), which we reshape into tokens $(B, H'W', C)$. With patch size 8 and image size 112, this yields a 14×14 token grid (*i.e.*, 196 tokens), aligning with the token-count budget used by the rest of the architecture.

A.3 Pipeline of Planning on Manipulation Tasks

In this subsection, we describe the latent-space planning and control procedures. We use CEM [Draeger et al., 1995] for goal reaching, LDP [Xie et al., 2025] for high-dimensional contact-rich manipulation, and SAC [Haarnoja et al., 2018] for locomotion. The distinction is intrinsic to the objective: CEM minimizes the distance to a *goal latent* $\mathbf{z}_g$ (below), so it applies wherever a target state is specified; locomotion provides no such goal latent and instead maximizes return, which we optimize with SAC.

On Goal-Reaching Tasks. For goal-reaching tasks (navigation, planar pushing, and target reaching) we plan directly with CEM, treating the encoded goal latent as the target state.

(1) **Latent Encoding.** The current input uses the joint image–proprioception embedding $\mathbf{x}_0 = [f^o_{\text{emb}}(\mathbf{o}_0), f^p_{\text{emb}}(\mathbf{s}^p_0)]$. For an RGB-only goal, the standardized proprioception slot is set to zero, giving $\mathbf{x}_g = [f^o_{\text{emb}}(\mathbf{o}_g), f^p_{\text{emb}}(0)]$. We encode both inputs as

$$\mathbf{z}_0 = q(\mathbf{z}_0 \mid \mathbf{x}_0), \quad \mathbf{z}_g = q(\mathbf{z}_g \mid \mathbf{x}_g).$$

(2) **Planning Objective.** The objective is to minimize the distance between the predicted final latent $\hat{\mathbf{z}}_{H_p}$ and the goal latent $\mathbf{z}_g$:

$$\mathcal{C} = \|\hat{\mathbf{z}}_{H_p} - \mathbf{z}_g\|^2, \quad \text{where} \quad \hat{\mathbf{z}}_{H_p} = f_\theta^{H_p}(\mathbf{z}_0, \mathbf{a}_{0:H_p-1}).$$

(3) **Action Sampling.** At each planning iteration, CEM samples a population of N action sequences $\{\mathbf{a}^i_{0:H_p-1}\}_{i=1}^N$ from a Gaussian distribution. The initial mean and covariance are initialized as $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$.

(4) **Trajectory Rollout.** For each sampled sequence, the learned world model predicts the latent rollout:

$$\hat{\mathbf{z}}_t = f^t_\theta(\mathbf{z}_0, \mathbf{a}_{0:t-1}), \quad t = 1, \ldots, H_p,$$

and the trajectory cost $\mathcal{C}^i$ is computed according to the objective above.

(5) **Elite Selection and Update.** The top K trajectories with the lowest costs are selected as elites, and the sampling distribution is updated by:

$$\boldsymbol{\mu} \leftarrow \text{mean}(\mathbf{a}^{(1:K)}), \quad \boldsymbol{\Sigma} \leftarrow \text{cov}(\mathbf{a}^{(1:K)}).$$

(6) **Iteration.** A new population of N trajectories is resampled from the updated Gaussian, and the process is repeated for a fixed number of iterations T_{cem}.

(7) **Execution.** After optimization, the first H_a actions $\mathbf{a}_{0:H_a-1}$ are executed in the environment. At the next timestep, a new observation $\mathbf{x}_{H_a}$ is encoded and the planning process repeats, forming a closed-loop receding-horizon controller.

On Manipulation Tasks. For high-dimensional, contact-rich actions we couple TC-WM with the Latent Diffusion Planning (LDP) framework [Xie et al., 2025].

(1) **Input Encoding.** We use the same current and goal input convention as CEM, including zero filling the standardized proprioception slot of an RGB-only goal:

$$\mathbf{z}_0 = q(\mathbf{z}_0 \mid \mathbf{x}_0), \quad \mathbf{z}_g = q(\mathbf{z}_g \mid \mathbf{x}_g).$$

(2) **Planning Objective.** The planner forecasts a sequence of future latent states $[\hat{\mathbf{z}}_{k+1}, \ldots, \hat{\mathbf{z}}_{k+H_p}]$ conditioned on $\mathbf{z}_k$, optimized by the diffusion loss:

$$\mathcal{L}_{\text{planner}}(\psi) = \mathbb{E}_{t,\epsilon}\big[\|\epsilon_\psi(\hat{\mathbf{z}}_{k+1}, \ldots, \hat{\mathbf{z}}_{k+H_p}; \mathbf{z}_k, t) - \epsilon\|^2\big].$$

(3) **Inverse Dynamics Model.** The IDM reconstructs actions between latent states via:

$$\mathcal{L}_{\text{IDM}}(\xi) = \mathbb{E}_{t,\epsilon}\big[\|\epsilon_\xi(\hat{\mathbf{a}}_k; \mathbf{z}_k, \mathbf{z}_{k+1}, t) - \epsilon\|^2\big].$$

(4) **Latent Trajectory Sampling.** At inference, the planner samples $\hat{\mathbf{z}}_{k+1}, \ldots, \hat{\mathbf{z}}_{k+H_p} \sim \mathcal{N}(0, I)$ and iteratively denoises via DDPM updates using ϵ_ψ.

(5) **Action Reconstruction.** For each pair $(\hat{\mathbf{z}}_{k+i}, \hat{\mathbf{z}}_{k+i+1})$, actions $\hat{\mathbf{a}}_{k+i}$ are generated through diffusion updates with ϵ_ξ.

(6) **Execution and Replanning.** The first H_a actions $\hat{\mathbf{a}}_k, \ldots, \hat{\mathbf{a}}_{k+H_a-1}$ are executed, and the process repeats with new latent observation $\mathbf{z}_{k+H_a}$, forming a closed-loop latent MPC scheme.

On Locomotion Tasks. Locomotion (`Cheetah`, `Hopper`) is scored by long-horizon return rather than by reaching a fixed goal latent, so distance-to-goal planning does not apply. We instead treat TC-WM's latent $\mathbf{z}_t$ as the observation of a model-free Soft Actor–Critic (SAC) agent [Haarnoja et al., 2018], trained from environment interaction under the same protocol as the baselines, and report the mean episode return over 3 seeds.

A.4 Implementation Details of Baselines

Offline Version of DreamerV3 We implement an offline variant of DreamerV3 [Hafner et al., 2023] that trains exclusively from a fixed replay buffer without any environment interaction. The offline pipeline reads precomputed trajectories (images $\mathbf{o}_t$, actions $\mathbf{a}_t$, and optional proprio/state and rewards), builds the DreamerV3 replay, and runs the standard DreamerV3 world-model training loop on randomly sampled batches. Evaluation is performed on held-out trajectories by reconstructing and open-loop predicting image sequences: the model encodes a context window of observed frames (posterior) and imagines the future (prior), after which the decoder produces reconstructions/predictions that are compared to ground truth. We keep the original DreamerV3 architecture and losses, and only adapt the data loading, replay ingestion, and offline evaluation logic to accommodate reward-free datasets and dataset-specific observation formats.

Offline Version of TD-MPC2 We implement TD-MPC2-Offline by adding a replay-only training pipeline to the official TD-MPC2 codebase [Hansen et al., 2024]. The offline trainer removes all environment interaction and learns solely from a fixed dataset. `Robomimic` demonstrations are loaded from preconverted episode files (RGB video, actions, rewards, terminals), and a fixed replay buffer is built once at startup. Training samples random sequences from this buffer, matching the original TD-MPC2 update equations; no online data is added. For evaluation, we introduce an offline validation split and compute losses on held-out sequences without parameter updates. To align with visual world-model diagnostics, we add a reconstruction decoder and rollout visualization: the decoder is trained with reconstruction loss (with backpropagating into the latent), and rollouts are generated by iterating the latent predictor and decoding predicted latents. Rollout visualization uses the same format as our method. We also extend rollout logic to respect frame skipping by applying the full action sequence within each skip window (rather than subsampling actions), ensuring motion fidelity in long-horizon rollouts.

Offline Version of MuZero We adapt MuZero [Schrittwieser et al., 2020] to the offline setting by keeping its three-network decomposition—representation h, dynamics g, and prediction f—and supervising the joint model on fixed trajectories instead of MCTS-driven self-play. In our implementation, h is the frozen DINOv2 encoder followed by the same compact linear projector as TC-WM, g is a Transformer next-latent predictor that consumes $(\mathbf{z}_t, \mathbf{a}_t)$, and f collects the per-step heads over the latent. As our datasets carry no dense reward, f reduces to a policy head f^π_ξ trained by behaviour cloning on the recorded action. Reward and value heads are disabled on datasets without reward labels. The training loss is $\mathcal{L}_{\text{cons}} + 0.25\,\|f^\pi_\xi(\mathbf{z}_t) - \mathbf{a}_t\|_2^2$, where $\mathcal{L}_{\text{cons}}$ matches the unrolled latent to the encoded next-step latent with a stop-gradient on the target, as in the original MuZero. With reward labels the standard $\mathcal{L}_{\text{cons}} + \mathcal{L}_r + 0.25\,\mathcal{L}_v + 0.25\,\mathcal{L}_\pi$ objective is recovered by simply enabling the reward and Monte-Carlo value heads.

A.5 Training and Downstream Configurations

The public implementation is available at https://github.com/MinghaoFu/TC-WM, with environment setup and training, planning, and rollout instructions. Table A2 summarizes the training and downstream settings.

Table A2: Training settings and downstream configurations.

Component	Configuration
Frozen encoder	DINOv2 ViT-S/14, 224×224 RGB, 384-dimensional patch tokens
Temporal context	3 context steps and 1 target step
Compact latent	$d_z = 256$, $d_s = 64$, $d_c = 192$
Transition model	6-layer ViT, 16 attention heads, MLP width 2048, dropout 0.1
Loss weights	$\lambda^z_{\text{dyn}} = \lambda^s_{\text{dyn}} = \lambda_{\text{align}} = \lambda_{\text{rec}} = 1$
Alignment	Temperature $\tau = 0.1$, ℓ_1 coefficient 10^{-4}
Optimizer	AdamW, learning rate 5×10^{-4} for projector and dynamics, 3×10^{-4} for decoder and proprioception heads, weight decay 10^{-4}
Training	Batch size 32, 300k updates, 3 random seeds
CEM	Horizon 5, 300 candidates, 30 elites, 10 updates for `Maze`, `Wall`, and `Reacher`, 30 for `Push-T`, 5 actions executed before replanning
LDP and IDM	Horizon 16, 50 denoising steps, 1 action executed before replanning
SAC	Two-layer actor and critics with 256 units per layer, batch size 256, learning rate 3×10^{-4}, $\gamma = 0.99$, target-update coefficient 0.005, automatic entropy tuning

A.6 Compute Resources

All training and evaluation runs use a single NVIDIA H100 80 GB GPU per experiment. A typical TC-WM run on `Lift` / `Can` / `Square` converges in roughly 8 hours over 100 epochs at peak memory $\approx$ 35 GB; `Push-T` / `Wall` / `Maze` converge in 1–2 hours; the DMC tasks (`Reacher`, `Cheetah`, `Hopper`) take 4–6 hours each. Each baseline (TD-MPC2, DreamerV3, MuZero, DINO-WM) uses the same single-H100 budget per environment, with comparable wall-clock times. Aggregating training, evaluation, and the linear-probe / unseen-perturbation studies, the experiments reported in the paper consume roughly 1,200 H100 GPU-hours.

A.7 Additional Controls and Robustness

Robustness to imperfect proprioception. On `Lift`, we perturb only the standardized training-time alignment target, leaving encoder inputs and evaluation trajectories clean. Gaussian noise uses $\tilde{\mathbf{s}}^p_t = \bar{\mathbf{s}}^p_t + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ and $\bar{\mathbf{s}}^p_t$ denotes standardized proprioception. We also mask 25% of coordinates, delay the target by one step, or shuffle targets within each minibatch. Table A4 reports means over three seeds. Moderate noise and missing coordinates have a smaller effect than temporal mismatch or shuffled supervision.

Matched supervision and three-task ablations. We equip DINO-WM with the same raw-proprioception prediction head and loss weight as TC-WM, and repeat the supervision and alignment ablations on all three manipulation tasks. Table A5 reports mean success over three seeds. The matched head improves DINO-WM by 4 to 10 percentage points, while removing InfoNCE from TC-WM costs 12, 16, and 12 points on `Lift`, `Can`, and `Square`, respectively.

Held-out object and goal probes. We probe $\mathbf{z}^s$ and $\mathbf{z}^c$ separately for object and goal variables that are never used as training inputs or targets. Five-fold ridge probes report mean $R^2 \pm$ standard error over three seeds in Table A6. The aligned block performs better for each variable group despite having one-third the width of the complementary block.

Goal-reaching on `Reacher`. Table A7 complements the eight-task planning comparison in the main paper with CEM evaluation on `Reacher`. Both methods achieve high goal-reaching success.

Table A3: Physical interpretation of state observations and their partitioning into three task-centric variable groups for the `Robomimic` datasets, based on their roles in the robot trajectory. Gray entries are *not* used during training; they serve only as ground-truth references for planning.

Dataset	Proprio	Object	Goal
`Lift`	`robot0_joint_pos`, `robot0_joint_pos_cos`, `robot0_eef_pos`, `robot0_eef_quat_site`, `robot0_gripper_qvel`	`robot0_joint_pos_sin`, `robot0_joint_vel`, `robot0_eef_quat`, `robot0_gripper_qpos`,	`object_pos`, `object_quat`	`grip_obj_rel_vec`
`Can`	`robot0_joint_pos`, `robot0_joint_pos_cos`, `robot0_eef_pos`, `robot0_eef_quat_site`, `robot0_gripper_qvel`	`robot0_joint_pos_sin`, `robot0_joint_vel`, `robot0_eef_quat`, `robot0_gripper_qpos`,	`object_pos`, `object_quat`	`target_pos`, `target_quat`
`Square`	`robot0_joint_pos`, `robot0_joint_pos_cos`, `robot0_eef_pos`, `robot0_eef_quat_site`, `robot0_gripper_qvel`	`robot0_joint_pos_sin`, `robot0_joint_vel`, `robot0_eef_quat`, `robot0_gripper_qpos`,	`object_pos`, `object_quat`	`target_pos`, `target_quat`

Table A4: Training-time proprioceptive perturbations on `Lift`.

Alignment target	SR (%) ↑	Probe R^2 ↑
Clean	60	0.59
Gaussian noise, $\sigma = 0.05$	59	0.57
Gaussian noise, $\sigma = 0.10$	56	0.53
25% coordinates masked	55	0.51
One-step delay	49	0.43
Within-batch shuffle	31	0.09

Complementary-latent dimension. We vary d_c on `Lift` while retaining the physically aligned block. Table A8 reports success rate and planning time relative to $d_c = 0$. Including the complementary latent raises success from 43% to 60%, with a 19% planning-time increase at the default width.

Latent rollout convention. Let H_c denote the transition context length and H_p the planning horizon. Starting from $\hat{\mathbf{z}}_0 = \mathbf{z}_0 = [\mathbf{z}_0^s, \mathbf{z}_0^c]$, let $\mathcal{H}_k$ contain the latest H_c full-latent/action pairs ending at $(\hat{\mathbf{z}}_k, \mathbf{a}_k)$, with missing initial pairs padded by $(\mathbf{z}_0, 0)$. We define

$$\hat{\mathbf{z}}_{k+1} = f_\theta(\mathcal{H}_k), \quad k = 0, \ldots, H_p - 1, \qquad f_\theta^{H_p}(\mathbf{z}_0, \mathbf{a}_{0:H_p-1}) = \hat{\mathbf{z}}_{H_p}.$$

The context is updated with each predicted full latent and candidate action. Thus H_p actions produce H_p transitions.

B Theoretical Rationale: When and How can We Learn Task-centric Representation from Visual Foundations?

This appendix expands the theoretical statement of Theorem 1. Section B.1 discusses the motivation, the underlying hierarchical generative model, the assumptions, and an empirical sanity check on `Lift`. Section B.2 contains the full proof.

Table A5: Matched-proprioception controls and alignment ablations. Success rate (%).

Variant	`Lift`	`Can`	`Square`
DINO-WM	32	30	24
DINO-WM + raw-proprioception head	42	38	28
TC-WM without proprioceptive supervision	38	34	24
TC-WM without InfoNCE alignment	48	46	28
Full TC-WM	**60**	**62**	**40**

Table A6: Separate latent-block probes on held-out object and goal variables.

	Object R^2		Goal R^2	
Task	$\mathbf{z}^s$	$\mathbf{z}^c$	$\mathbf{z}^s$	$\mathbf{z}^c$
`Lift`	0.72 ± 0.02	0.55 ± 0.03	0.67 ± 0.02	0.49 ± 0.04
`Can`	0.69 ± 0.03	0.52 ± 0.02	0.63 ± 0.03	0.45 ± 0.04
`Square`	0.56 ± 0.04	0.40 ± 0.05	0.51 ± 0.03	0.36 ± 0.05

Table A7: CEM success rate on `Reacher`.

Method	SR (%) ↑
DINO-WM	92.0
TC-WM	93.4 ± 1.6

B.1 Illustration and Analysis

Understanding when and how task-centric representation can be learned from visual foundation models is critical for modern world-model training. We study this problem through *identifiability*, the asymptotic guarantee of *recovering* a semantically meaningful latent space. The discussion is organized around three previously underexplored questions: `(1)` how imposing a latent Markov structure on top of visual foundations helps guide world representations; `(2)` why a simple linear projection is sufficient to transform rich visual features into task-relevant world representations; and `(3)` empirically validating these claims through ablation studies and linear probing experiments.

We first establish identifiability of task-centric representation in a general framework.

Definition 2 (Recoverability of Task-Centric Representation)**.** *The estimated $\hat{\mathbf{z}}_t^s$ is a nontrivial transformation of the true $\mathbf{z}_t^s$ in the defined process of Sec. 2.2 for each frame.*

We next show that, under mild regularity assumptions, our formulation described in Sec. 2.2 guarantees such recoverability.

A View from Hierarchical Model. We analyze the recovery of task-centric latent factors in the hierarchy shown in Figure A1. Our architecture forms a three-level hierarchy: raw observations $\mathbf{o}_t$ are mapped to foundation embeddings $\mathbf{x}_t$ by a frozen encoder, which are then projected into a compact latent $\mathbf{z}_t = [\mathbf{z}_t^s, \mathbf{z}_t^c]$. The theorem below guarantees that this hierarchical process preserves the task-relevant factors.

Table A8: Complementary-latent dimension ablation on `Lift`.

d_c	SR (%) ↑	Relative planning time ↓
0	43	1.00×
64	52	1.04×
128	57	1.11×
192 (default)	60	1.19×

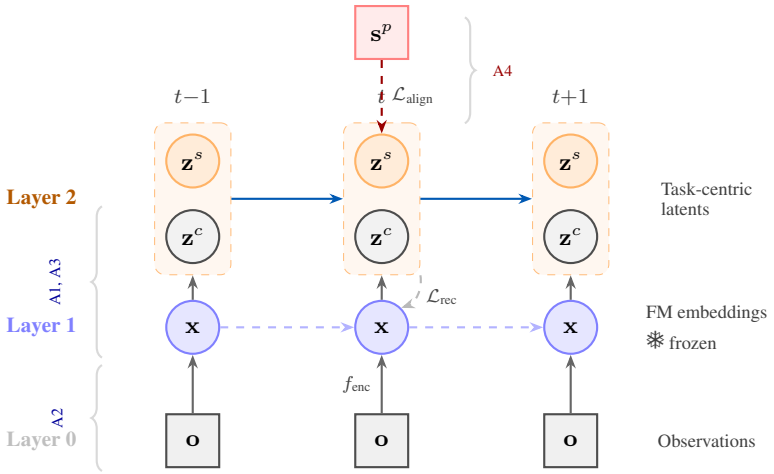


Figure A1: Graphical model of TC-WM's hierarchical structure. Circles denote latent variables; squares denote observed variables. Observations $\mathbf{o}_t$ are mapped to frozen FM embeddings $\mathbf{x}_t$ (Layer 1), then projected to a compact latent $\mathbf{z}_t = [\mathbf{z}_t^s, \mathbf{z}_t^c]$ (Layer 2, dashed orange box). The unified temporal arrows act on the joint latent $\mathbf{z}_t$. $\mathcal{L}_{\text{align}}$ aligns $\mathbf{z}_t^s$ with proprioception $\mathbf{s}_t^p$; $\mathcal{L}_{\text{rec}}$ (curved) regularises by reconstructing the embedding from $\mathbf{z}_t$. Assumptions A1–A4 of Theorem A1 are annotated.

We first formalize this property through the concept of *recoverability*, which ensures that the learned latents are meaningful transformations of the true task-relevant variables.

Theorem A1 ***(Task-centric identifiability from visual embeddings)***

Suppose images $\mathbf{o}_t$, frozen embeddings $\mathbf{x}_t$, and latent variables $\mathbf{z}_t = [\mathbf{z}_t^s, \mathbf{z}_t^c]$ follow the process in Section 2.2, and the learned latent world model matches the true joint distribution of adjacent embeddings $\{\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}\}$. Assume:

A1. ***(Contextual dynamics):*** *$L_{\mathbf{x}_{t+1:t+H}|\mathbf{z}_t}$ and $L_{\mathbf{x}_{t-H:t-1}|\mathbf{x}_{t+1:t+H}}$ are injective and bounded.*

A2. ***(Latent–observed variability):*** *For any $\mathbf{z}_t^{(1)} \neq \mathbf{z}_t^{(2)}$, $p(\mathbf{x}_t \mid \mathbf{z}_t^{(1)}) \neq p(\mathbf{x}_t \mid \mathbf{z}_t^{(2)})$.*

A3. ***(Differentiability):*** *There exists a differentiable F such that $F[p_{\mathbf{x}_t|\mathbf{z}_t}(\cdot \mid \mathbf{z}_t)] = h_z(\mathbf{z}_t)$.*

A4. ***(Partial alignment):*** *The alignment objective in Eq. (1) is sufficiently minimized.*

Then the learned $\mathbf{z}_t$ is block-wise identifiable, and the task-centric $\mathbf{z}_t^s$ is affine-identifiable:

$$\underbrace{\hat{\mathbf{z}}_t = h_z(\mathbf{z}_t), \quad h_z \text{ invertible and } C^1}_{\textbf{Latent Space Recovery: the compact latent preserves the true world state up to reparameterization}} \qquad \underbrace{\mathbf{z}_t^s = \mathbf{A}\hat{\mathbf{z}}_t^s + \mathbf{b}}_{\textbf{Task-centric recovery: the aligned block recovers physical factors up to an affine map}}$$

Takeaway

The theorem characterizes latent recovery from temporal embedding statistics and physical alignment. Under its population-level assumptions, the full latent is identifiable up to a smooth reparameterization and the aligned task-centric block up to an affine map. The result connects our approach to a broader family of "dynamics on pretrained encoders," including recent generative and latent diffusion frameworks such as VideoLDM [Blattmann et al., 2023], DiT [Peebles and Xie, 2023], RAW [Zheng et al., 2025], and REPA [Yu et al., 2025].

Proof Sketch. Our argument unfolds in three steps. (1) Under nonparametric assumptions, embeddings $\mathbf{x}_t$ are invertible transformations of images $\mathbf{o}_t$ (*e.g.*, DINO or β-VAE), thus preserving the environment's essential factors. (2) For the embedding-level latent $\mathbf{z}_t$, we construct an integral operator over $2H+1$ adjacent observations and show, via Markov conditional independence and a spectral decomposition argument, that any model matching the true joint distribution of $\{\mathbf{x}_{t-H:t+H}\}$ recovers $\mathbf{z}_t$ up to an invertible reparametrisation h_z. (3) For the task-centric subspace $\mathbf{z}_t^s$, the InfoNCE alignment with the proprioception $\mathbf{s}_t^p$ collapses h_z on the $\mathbf{z}^s$-block to a map that depends only on $\mathbf{z}_t^s$, and an exponential-family/dot-product argument then forces it to be affine. The full proof is given below.

Empirical Diagnostics of Assumptions on `Lift`. We complement the theory with finite-sample diagnostics related to A1–A4 on a fully trained TC-WM checkpoint (`Lift`, $d_z = 256$, $d_s = 64$). All probes operate on the patch-mean-pooled latent $\mathbf{z}_t$ and the mean-pooled foundation embedding $\mathbf{x}_t$; all randomness uses fixed seeds. These probes characterize the learned geometry rather than establish the population-level assumptions.

A1 (Contextual Dynamics). A1 demands that $L_{\mathbf{x}_{t+1:t+H}|\mathbf{z}_t}$ be both bounded and injective. We probe the closely related forward sensitivity $\|\Delta\mathbf{x}\|/\|\Delta\mathbf{z}\|$ on 4096 random pairs of (latent, embedding) co-drawn from validation trajectories. The ratio is tightly concentrated, with median 14.37, 5th percentile 11.16, and 95th percentile 19.26 (Figure A2); these sampled ratios characterize forward sensitivity rather than establish injectivity or boundedness of the conditional operator.

A2 (Latent–Observed Variability). A2 demands that distinct latents yield distinct conditional embeddings, $\mathbf{z}_t^{(1)} \neq \mathbf{z}_t^{(2)} \Rightarrow p(\mathbf{x}_t \mid \mathbf{z}_t^{(1)}) \neq p(\mathbf{x}_t \mid \mathbf{z}_t^{(2)})$. As a population-level surrogate we measure how strongly latent distance predicts embedding distance: Figure A2 plots $\|\mathbf{z}_i - \mathbf{z}_j\|$ against $\|\mathbf{x}_i - \mathbf{x}_j\|$ over 2037 random validation pairs. The two distance series are nearly comonotone, with Spearman $\rho = 0.910$ and Pearson $\rho = 0.911$; in particular, no points cluster near the $\Delta\mathbf{x} \approx 0$ axis at non-zero $\Delta\mathbf{z}$, showing no such collapse among the sampled pairs.

A3 (Differentiability). The projector and embedding decoder are smooth C^1 maps. Differentiability of the conditional-density functional in A3 remains a population-level assumption.

A4 (Partial Alignment). A4 demands that the InfoNCE alignment loss be sufficiently minimized. We linearly probe both subspaces against the proprioceptive state $\mathbf{s}_t^p$ (Figure A2). Alignment raises

the proprioceptive probe R^2 of the 64-dimensional task-centric block from 0.534 ± 0.027 to 0.592 ± 0.007. The 192-dimensional complementary block achieves 0.680 ± 0.008 and changes little with alignment.

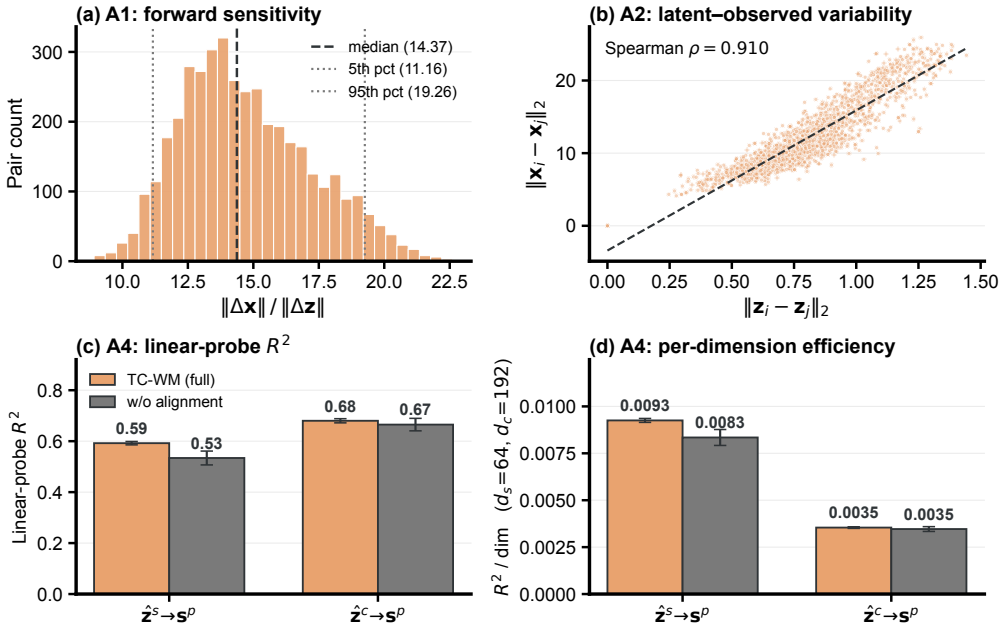


Figure A2: Empirical diagnostics related to assumptions A1, A2, and A4 on `Lift`. **(a)** A1 (Contextual Dynamics): distribution of the forward-sensitivity ratio $\|\Delta\mathbf{x}\|/\|\Delta\mathbf{z}\|$ over 4096 validation pairs, the ratio is bounded above and away from zero, characterizing sensitivity on the sampled pairs. **(b)** A2 (Latent–Observed Variability): scatter of $\|\mathbf{z}_i - \mathbf{z}_j\|$ against $\|\mathbf{x}_i - \mathbf{x}_j\|$ for 2037 validation pairs (Spearman $\rho = 0.910$), latent and embedding distances co-vary on the sampled pairs. **(c)** A4 (Partial Alignment): linear-probe R^2 from $\hat{\mathbf{z}}_t^s$ and $\hat{\mathbf{z}}_t^c$ to $\mathbf{s}_t^p$ (5-fold CV, ridge α=1); the InfoNCE alignment lifts $R^2(\hat{\mathbf{z}}^s \to \mathbf{s}^p)$ from 0.534 to 0.592 while leaving $\hat{\mathbf{z}}^c$ essentially unchanged. **(d)** The same probe scores divided by subspace dimension (d_s=64 or d_c=192).

B.2 Proof of Theorem A1

Notation and setup. We restate the data-generating process from Section 2.2 in a form convenient for the proof. Observations $\mathbf{o}_t \in \mathcal{O}$ are mapped by a frozen visual foundation encoder f_{enc} to embeddings $\mathbf{x}_t = f_{\text{enc}}(\mathbf{o}_t) \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$, and the embedding is generated from a compact latent $\mathbf{z}_t \in \mathcal{Z} \subseteq \mathbb{R}^{d_z}$ through a (possibly noisy) mixing

$$\mathbf{x}_t = g(\mathbf{z}_t, \boldsymbol{\epsilon}_t^0), \qquad \boldsymbol{\epsilon}_t^0 \perp \boldsymbol{\epsilon}_{t'}^0 \text{ for } t \neq t', \qquad \boldsymbol{\epsilon}_t^0 \perp \mathbf{z}_{1:T}, \tag{A1}$$

with the noise sequence $\{\boldsymbol{\epsilon}_t^0\}$ exogenous and i.i.d. The latent splits as $\mathbf{z}_t = [\mathbf{z}_t^s, \mathbf{z}_t^c]$ with $\mathbf{z}_t^s \in \mathcal{Z}^s \subseteq \mathbb{R}^{d_s}$ and $\mathbf{z}_t^c \in \mathcal{Z}^c \subseteq \mathbb{R}^{d_z - d_s}$, and the latent process $\{\mathbf{z}_t\}$ is first-order Markov,

$$p(\mathbf{z}_{t+1} \mid \mathbf{z}_{1:t}) = p(\mathbf{z}_{t+1} \mid \mathbf{z}_t). \tag{A2}$$

The proprioception used by the InfoNCE alignment of Eq. (1) is a deterministic function of the task-centric block,

$$\mathbf{s}_t^p = m(\mathbf{z}_t^s), \qquad m : \mathcal{Z}^s \to \mathcal{S}^p \text{ is a } C^1\text{-diffeomorphism onto its image.} \tag{A3}$$

For any two random variables $\boldsymbol{u} \in \mathcal{U}$ and $\boldsymbol{v} \in \mathcal{V}$ with conditional density $p_{\boldsymbol{v}|\boldsymbol{u}}$, the induced *linear (integral) operator* is

$$L_{\boldsymbol{v}|\boldsymbol{u}} : \mathcal{F}(\mathcal{U}) \longrightarrow \mathcal{F}(\mathcal{V}), \qquad [L_{\boldsymbol{v}|\boldsymbol{u}}\phi](\boldsymbol{v}) := \int_{\mathcal{U}} p(\boldsymbol{v} \mid \boldsymbol{u})\, \phi(\boldsymbol{u})\, \mathrm{d}\boldsymbol{u},$$

where $\mathcal{F}(\cdot)$ denotes a suitable space of L^1-densities, and is bounded whenever the conditional density is essentially bounded. We additionally use the multiplication operator $\Delta_{\mathbf{x}_t} : \mathcal{F}(\mathcal{Z}) \to \mathcal{F}(\mathcal{Z})$

defined by $[\Delta_{\mathbf{x}_t}\phi](\mathbf{z}_t) := p(\mathbf{x}_t \mid \mathbf{z}_t)\,\phi(\mathbf{z}_t)$, which acts as a (functional-valued) diagonal scaling indexed by $\mathbf{z}_t$ and parametrised by $\mathbf{x}_t$.

We use $\hat{\cdot}$ to denote estimated quantities: $\hat{\mathbf{z}}_t \in \hat{\mathcal{Z}}$, $\hat{g}$, $\hat{p}$, etc. The hypothesis of Theorem A1 asserts that the estimated and true joint distributions of $\{\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}\}$ coincide.

Conditional independence structure. The combination of (A1), the i.i.d. noise assumption, and the Markov property (A2) implies the two relations

$$\mathbf{x}_{t-H:t-1} \perp \mathbf{x}_{t+1:t+H} \mid \mathbf{z}_t, \qquad \mathbf{x}_t \perp \mathbf{x}_{t-H:t-1}, \mathbf{x}_{t+1:t+H} \mid \mathbf{z}_t, \tag{CI}$$

which we use throughout. The first relation says that, conditioned on the present latent $\mathbf{z}_t$, past and future embeddings are independent; this is the Markov screening property combined with the fact that each $\mathbf{x}_\tau$ is a noisy function of $\mathbf{z}_\tau$ alone. The second says that the present embedding $\mathbf{x}_t$ depends on the past and future only through $\mathbf{z}_t$, which follows directly from (A1): $\mathbf{x}_t$ is a function of $(\mathbf{z}_t, \boldsymbol{\epsilon}_t^0)$, and $\boldsymbol{\epsilon}_t^0$ is independent of every other quantity.

We first establish block-wise identifiability of $\mathbf{z}_t$ from the matched joint distribution of $2H+1$ adjacent embeddings, and then refine this to the affine recovery on the task-centric subspace using the InfoNCE alignment objective.

Fix a time index t. The goal is to show that any estimated model whose joint distribution over $\{\mathbf{x}_{t-H}, \dots, \mathbf{x}_{t+H}\}$ matches the true one must agree with the true generative model up to an invertible reparametrisation of the latent at t. By the first relation in (CI), the joint density of past and future windows admits the latent-mediated decomposition

$$p(\mathbf{x}_{t-H:t-1}, \mathbf{x}_{t+1:t+H}) = \int_{\mathcal{Z}} p(\mathbf{x}_{t-H:t-1} \mid \mathbf{z}_t)\, p(\mathbf{x}_{t+1:t+H} \mid \mathbf{z}_t)\, p(\mathbf{z}_t)\, \mathrm{d}\mathbf{z}_t. \tag{A4}$$

Dividing by the marginal $p(\mathbf{x}_{t+1:t+H})$ and recognising $p(\mathbf{z}_t \mid \mathbf{x}_{t+1:t+H}) = p(\mathbf{x}_{t+1:t+H} \mid \mathbf{z}_t)\, p(\mathbf{z}_t)/p(\mathbf{x}_{t+1:t+H})$ on the right, this rewrites in operator form as

$$L_{\mathbf{x}_{t-H:t-1} \mid \mathbf{x}_{t+1:t+H}} = L_{\mathbf{x}_{t-H:t-1} \mid \mathbf{z}_t} \circ L_{\mathbf{z}_t \mid \mathbf{x}_{t+1:t+H}}. \tag{A5}$$

Each composing operator is bounded by A1. To see that $L_{\mathbf{z}_t|\mathbf{x}_{t+1:t+H}}$ is well defined and injective, observe that Bayes' rule yields $L_{\mathbf{z}_t|\mathbf{x}_{t+1:t+H}} = D_{p(\mathbf{z}_t)} \circ L^*_{\mathbf{x}_{t+1:t+H}|\mathbf{z}_t} \circ D^{-1}_{p(\mathbf{x}_{t+1:t+H})}$, where D_q denotes the multiplication operator by density q and L^* is the formal adjoint; injectivity of $L_{\mathbf{x}_{t+1:t+H}|\mathbf{z}_t}$ from A1 transfers, via the adjoint and the strict positivity of the marginals on the relevant supports, to injectivity of the posterior operator on the range that arises in (A5).[1] Combined with the injectivity of the cross-operator $L_{\mathbf{x}_{t-H:t-1}|\mathbf{x}_{t+1:t+H}}$ from A1, equation (A5) admits a unique left-inverse on the relevant function space; we will use this to invert the composition in Step 3.

We now bring the present embedding $\mathbf{x}_t$ into the picture. By the second relation in (CI), the four-way joint factorises with $\mathbf{x}_t$ entering only through $p(\mathbf{x}_t \mid \mathbf{z}_t)$:

$$p(\mathbf{x}_{t-H:t-1}, \mathbf{x}_t, \mathbf{x}_{t+1:t+H}) = \int_{\mathcal{Z}} p(\mathbf{x}_{t-H:t-1} \mid \mathbf{z}_t)\, p(\mathbf{x}_t \mid \mathbf{z}_t)\, p(\mathbf{x}_{t+1:t+H} \mid \mathbf{z}_t)\, p(\mathbf{z}_t)\, \mathrm{d}\mathbf{z}_t.$$

Dividing by the marginal $p(\mathbf{x}_{t+1:t+H})$ and treating $\mathbf{x}_t$ as a parameter (so that $p(\mathbf{x}_t \mid \mathbf{z}_t)$ becomes a $\mathbf{z}_t$-indexed scalar), this rewrites as the operator identity

$$L_{\mathbf{x}_{t-H:t-1}, \mathbf{x}_t \mid \mathbf{x}_{t+1:t+H}} = L_{\mathbf{x}_{t-H:t-1}|\mathbf{z}_t} \circ \Delta_{\mathbf{x}_t} \circ L_{\mathbf{z}_t \mid \mathbf{x}_{t+1:t+H}}. \tag{A6}$$

Equations (A5) and (A6) together yield

$$\underbrace{L_{\mathbf{x}_{t-H:t-1}, \mathbf{x}_t \mid \mathbf{x}_{t+1:t+H}} \circ L^{-1}_{\mathbf{x}_{t-H:t-1} \mid \mathbf{x}_{t+1:t+H}}}_{=: T_{\mathbf{x}_t} \text{ (observable)}} = L_{\mathbf{x}_{t-H:t-1}|\mathbf{z}_t} \circ \Delta_{\mathbf{x}_t} \circ L^{-1}_{\mathbf{x}_{t-H:t-1}|\mathbf{z}_t}. \tag{A7}$$

Two facts about (A7) are central. First, the right-hand side is a similarity transform: $T_{\mathbf{x}_t}$ is $\Delta_{\mathbf{x}_t}$ written in the basis given by $L_{\mathbf{x}_{t-H:t-1}|\mathbf{z}_t}$. Since $\Delta_{\mathbf{x}_t}$ is the diagonal multiplication operator with eigenvalue $p(\mathbf{x}_t \mid \mathbf{z}_t)$ at index $\mathbf{z}_t$, the family $\{T_{\mathbf{x}_t}\}_{\mathbf{x}_t \in \mathcal{X}}$ is simultaneously diagonalised by the same

[1] The strict positivity of the marginals on supports is implicit in the assumption that the data process is non-degenerate, as is standard in identifiability arguments for nonlinear ICA. The integral operators are taken on densities supported within $\mathcal{X}$ and $\mathcal{Z}$.

eigenfunction system, namely the columns of $L_{\mathbf{x}_{t-H:t-1}|\mathbf{z}_t}$, with eigenvalues $\{p(\mathbf{x}_t \mid \mathbf{z}_t)\}_{\mathbf{z}_t \in \mathcal{Z}}$. Second, $T_{\mathbf{x}_t}$ is fully determined by the joint distribution of $\{\mathbf{x}_{t-H:t-1}, \mathbf{x}_t, \mathbf{x}_{t+1:t+H}\}$, a quantity matched by the estimated model under the hypothesis of the theorem. Any estimated decomposition

$$T_{\mathbf{x}_t} \;=\; \hat{L}_{\mathbf{x}_{t-H:t-1}|\hat{\mathbf{z}}_t} \circ \hat{\Delta}_{\mathbf{x}_t} \circ \hat{L}^{-1}_{\mathbf{x}_{t-H:t-1}|\hat{\mathbf{z}}_t}$$

must therefore agree with the true one as an operator equality.

It remains to show that the simultaneous diagonalisation of the family $\{T_{\mathbf{x}_t}\}_{\mathbf{x}_t \in \mathcal{X}}$ uniquely determines the eigenvalue assignment $\mathbf{z}_t \mapsto p(\mathbf{x}_t \mid \mathbf{z}_t)$ and the eigenfunction system $\mathbf{z}_t \mapsto p(\mathbf{x}_{t-H:t-1} \mid \mathbf{z}_t)$, up to a single relabelling of the index set $\mathcal{Z}$.

We first use A2 to argue distinct eigenvalues. Suppose, towards a contradiction, that for some pair $\mathbf{z}_t^{(1)} \neq \mathbf{z}_t^{(2)}$ in $\mathcal{Z}$ the maps $\mathbf{x}_t \mapsto p(\mathbf{x}_t \mid \mathbf{z}_t^{(1)})$ and $\mathbf{x}_t \mapsto p(\mathbf{x}_t \mid \mathbf{z}_t^{(2)})$ coincide as functions on $\mathcal{X}$. This is exactly the negation of A2, and is excluded; consequently for almost every $\mathbf{x}_t \in \mathcal{X}$ the eigenvalue assignment $\mathbf{z}_t \mapsto p(\mathbf{x}_t \mid \mathbf{z}_t)$ is injective on $\mathcal{Z}$.

Next we leverage the simultaneous diagonalisation. Fix $\mathbf{x}_t \in \mathcal{X}$ and let $\lambda(\mathbf{z}_t) := p(\mathbf{x}_t \mid \mathbf{z}_t)$. The eigenvalues of $T_{\mathbf{x}_t}$ are $\{\lambda(\mathbf{z}_t)\}_{\mathbf{z}_t}$ with associated eigenfunctions $\phi_{\mathbf{z}_t}(\cdot) := p(\mathbf{x}_{t-H:t-1} \mid \mathbf{z}_t)(\cdot)$. By A1 the eigenfunction system $\{\phi_{\mathbf{z}_t}\}_{\mathbf{z}_t}$ is linearly independent; otherwise two distinct $\mathbf{z}_t$ would induce identical conditionals $p(\mathbf{x}_{t-H:t-1} \mid \mathbf{z}_t)$, contradicting the injectivity of $L_{\mathbf{x}_{t-H:t-1}|\mathbf{z}_t}$.[2] The standard spectral theorem for compact operators with simple spectrum then guarantees that the eigenpairs (λ, ϕ) are unique up to a permutation of the index set: any second simultaneous diagonalisation $\hat{\lambda}(\hat{\mathbf{z}}_t)$, $\hat{\phi}_{\hat{\mathbf{z}}_t}$ must satisfy $\hat{\lambda}(\hat{\mathbf{z}}_t) = \lambda(\mathbf{z}_t)$ and $\hat{\phi}_{\hat{\mathbf{z}}_t} = \phi_{\mathbf{z}_t}$ whenever the indices are matched by some bijection $\pi : \mathcal{Z} \to \hat{\mathcal{Z}}$ with $\hat{\mathbf{z}}_t = \pi(\mathbf{z}_t)$.

To make this last step concrete: if $T_{\mathbf{x}_t}$ has the two diagonalisations

$$T_{\mathbf{x}_t} \;=\; \int \lambda(\mathbf{z}_t)\, \phi_{\mathbf{z}_t} \otimes \psi_{\mathbf{z}_t}\, \mathrm{d}\mathbf{z}_t \;=\; \int \hat{\lambda}(\hat{\mathbf{z}}_t)\, \hat{\phi}_{\hat{\mathbf{z}}_t} \otimes \hat{\psi}_{\hat{\mathbf{z}}_t}\, \mathrm{d}\hat{\mathbf{z}}_t,$$

where $\{\psi_{\mathbf{z}_t}\}$ are the dual basis vectors making the spectral resolution diagonal, then evaluating both sides as operators on $\phi_{\mathbf{z}_t^*}$ for any fixed $\mathbf{z}_t^*$ gives $\lambda(\mathbf{z}_t^*)\phi_{\mathbf{z}_t^*} = \int \hat{\lambda}(\hat{\mathbf{z}}_t)\, \langle \psi_{\mathbf{z}_t}, \phi_{\mathbf{z}_t^*} \rangle\, \hat{\phi}_{\hat{\mathbf{z}}_t}\, \mathrm{d}\hat{\mathbf{z}}_t$. Linear independence of the $\hat{\phi}$'s, combined with distinctness of the $\hat{\lambda}$'s away from a measure-zero set, forces this expansion to be supported at a single $\hat{\mathbf{z}}_t = \pi(\mathbf{z}_t^*)$. Repeating the argument across $\mathbf{z}_t^*$ yields the bijection π and the matching identities

$$p(\mathbf{x}_t \mid \mathbf{z}_t) \;=\; \hat{p}(\mathbf{x}_t \mid \pi(\mathbf{z}_t)), \qquad p(\mathbf{x}_{t-H:t-1} \mid \mathbf{z}_t) \;=\; \hat{p}(\mathbf{x}_{t-H:t-1} \mid \pi(\mathbf{z}_t)). \tag{A8}$$

The bijection π obtained above is purely set-theoretic; assumption A3 promotes it to a differentiable map. By A3 there is a functional F on conditional densities such that

$$F\big[\, p_{\mathbf{x}_t|\mathbf{z}_t}(\cdot \mid \mathbf{z}_t) \,\big] \;=\; h_z(\mathbf{z}_t) \quad \text{for all } \mathbf{z}_t \in \mathcal{Z},$$

where $h_z : \mathcal{Z} \to \mathbb{R}^{d_z}$ is differentiable. Since $p(\mathbf{x}_t \mid \cdot)$ separates points of $\mathcal{Z}$ by A2, h_z is in fact a C^1-injection on $\mathcal{Z}$, and hence a diffeomorphism onto its image whenever its Jacobian is non-degenerate (which is the regular case considered here). Applying F to the matched conditional from (A8),

$$h_z(\mathbf{z}_t) \;=\; F\big[p(\mathbf{x}_t \mid \mathbf{z}_t)\big] \;=\; F\big[\hat{p}(\mathbf{x}_t \mid \pi(\mathbf{z}_t))\big] \;=\; \hat{h}_z(\pi(\mathbf{z}_t)),$$

where $\hat{h}_z$ is the analogous functional applied to the estimated conditional, and is a differentiable map by the same argument applied to $\hat{p}$. Composing,

$$\pi(\mathbf{z}_t) \;=\; \hat{h}_z^{-1}\big(h_z(\mathbf{z}_t)\big).$$

The right-hand side is a composition of two C^1-diffeomorphisms onto their images, hence is itself a C^1-diffeomorphism. Identifying π with the diffeomorphism $h_z := \hat{h}_z^{-1} \circ h_z$ (overloading notation by absorbing the inversion into the definition, as is standard), we obtain

$$\hat{\mathbf{z}}_t \;=\; h_z(\mathbf{z}_t),$$

[2] Injectivity of $L_{\mathbf{x}_{t-H:t-1}|\mathbf{z}_t}$ on densities is the dual statement: distinct eigenfunctions $p(\mathbf{x}_{t-H:t-1} \mid \cdot)$ must produce distinct images.

with $h_z : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$ invertible and differentiable. This proves conclusion (i).

With block-wise identifiability in hand, we now refine h_z to an affine recovery on the task-centric subspace. Write $\hat{\mathbf{z}}_t = h_z(\mathbf{z}_t^s, \mathbf{z}_t^c)$ broken into its first d_s and last $d_z - d_s$ coordinates as $\hat{\mathbf{z}}_t = (\hat{\mathbf{z}}_t^s, \hat{\mathbf{z}}_t^c)$, and define

$$h_s \,:\, \mathcal{Z}^s \times \mathcal{Z}^c \;\longrightarrow\; \hat{\mathcal{Z}}^s, \qquad h_s(\mathbf{z}^s, \mathbf{z}^c) \;:=\; \pi_s\big(h_z(\mathbf{z}^s, \mathbf{z}^c)\big),$$

where π_s denotes projection onto the first d_s coordinates. *A priori*, h_s may depend non-trivially on both arguments. We will show that A4 forces h_s to be independent of $\mathbf{z}^c$, and that the residual map $\mathbf{z}^s \mapsto h_s(\mathbf{z}^s)$ is affine. We begin by deriving the population limit of the InfoNCE objective in our setting, working from Eq. (1). Write the per-anchor loss as

$$\ell_M(\hat{\mathbf{z}}^s, \mathbf{s}^p) \;:=\; -\log \frac{e^{\langle \hat{\mathbf{z}}^s,\, m'(\mathbf{s}^p)\rangle/\tau}}{e^{\langle \hat{\mathbf{z}}^s,\, m'(\mathbf{s}^p)\rangle/\tau} + \sum_{j=1}^{M} e^{\langle \hat{\mathbf{z}}^s,\, m'(\mathbf{s}_j^{p,-})\rangle/\tau}},$$

where $\mathbf{s}_j^{p,-}$ are i.i.d. negatives drawn from the marginal $p(\mathbf{s}^p)$ and $(\hat{\mathbf{z}}^s, \mathbf{s}^p) \sim p_{\text{pos}}$ is a positive pair from the joint distribution induced by the model. Splitting the logarithm and using the law of large numbers,

$$\tfrac{1}{M}\sum_{j=1}^{M} e^{\langle \hat{\mathbf{z}}^s,\, m'(\mathbf{s}_j^{p,-})\rangle/\tau} \;\xrightarrow[M\to\infty]{\text{a.s.}}\; \mathbb{E}_{\mathbf{s}^{p,-}\sim p(\mathbf{s}^p)}\big[\, e^{\langle \hat{\mathbf{z}}^s,\, m'(\mathbf{s}^{p,-})\rangle/\tau}\,\big] \;=:\; Z(\hat{\mathbf{z}}^s),$$

so that, up to additive $\log M$,

$$\mathbb{E}\,\ell_M(\hat{\mathbf{z}}^s, \mathbf{s}^p) \;\longrightarrow\; -\,\mathbb{E}_{(\hat{\mathbf{z}}^s,\mathbf{s}^p)\sim p_{\text{pos}}}\big[\,\langle \hat{\mathbf{z}}^s, m'(\mathbf{s}^p)\rangle/\tau\,\big] \;+\; \mathbb{E}_{\hat{\mathbf{z}}^s\sim p(\hat{\mathbf{z}}^s)}\big[\log Z(\hat{\mathbf{z}}^s)\,\big] \;+\; \log M. \tag{A9}$$

Define the model conditional

$$q(\mathbf{s}^p \mid \hat{\mathbf{z}}^s) \;:=\; \frac{e^{\langle \hat{\mathbf{z}}^s, m'(\mathbf{s}^p)\rangle/\tau}}{Z(\hat{\mathbf{z}}^s)}\, p(\mathbf{s}^p),$$

so that $-\log q(\mathbf{s}^p \mid \hat{\mathbf{z}}^s) = -\langle \hat{\mathbf{z}}^s, m'(\mathbf{s}^p)\rangle/\tau + \log Z(\hat{\mathbf{z}}^s) - \log p(\mathbf{s}^p)$. Substituting into (A9) and recognising the cross-entropy structure,

$$\mathbb{E}\,\ell_M \;\longrightarrow\; \mathbb{E}_{\hat{\mathbf{z}}^s\sim p(\hat{\mathbf{z}}^s)}\,\mathbb{E}_{\mathbf{s}^p\sim p(\mathbf{s}^p\mid\hat{\mathbf{z}}^s)}\big[-\log q(\mathbf{s}^p \mid \hat{\mathbf{z}}^s)\big] \;-\; \mathbb{E}\big[\log p(\mathbf{s}^p)\big] \;+\; \log M. \tag{A10}$$

The last two terms are constants independent of the encoder, so InfoNCE minimisation is, in the $M \to \infty$ limit, equivalent to minimising the cross-entropy

$$H\big(p(\mathbf{s}^p \mid \hat{\mathbf{z}}^s),\; q(\mathbf{s}^p \mid \hat{\mathbf{z}}^s)\big) \;=\; \mathbb{E}_{\mathbf{s}^p}\big[-\log q(\mathbf{s}^p \mid \hat{\mathbf{z}}^s)\big] \tag{A11}$$

in expectation over $\hat{\mathbf{z}}^s \sim p(\hat{\mathbf{z}}^s)$. The minimum of cross-entropy is attained, for each $\hat{\mathbf{z}}^s$, when $q(\cdot \mid \hat{\mathbf{z}}^s) = p(\cdot \mid \hat{\mathbf{z}}^s)$ pointwise.

We now show that this alignment forces h_s to be independent of $\mathbf{z}^c$. Substituting the explicit form of q into the optimum condition $q = p$,

$$p(\mathbf{s}^p \mid \hat{\mathbf{z}}^s) \;=\; \frac{e^{\langle \hat{\mathbf{z}}^s,\, m'(\mathbf{s}^p)\rangle/\tau}}{Z(\hat{\mathbf{z}}^s)}\, p(\mathbf{s}^p) \quad \text{for almost every } (\hat{\mathbf{z}}^s, \mathbf{s}^p).$$

This is a strictly positive density in $\mathbf{s}^p$ for each $\hat{\mathbf{z}}^s$. Now we contrast this with the actual joint distribution of $(\hat{\mathbf{z}}_t^s, \mathbf{s}_t^p)$ in our setting. We have $\hat{\mathbf{z}}_t^s = h_s(\mathbf{z}_t^s, \mathbf{z}_t^c)$ and $\mathbf{s}_t^p = m(\mathbf{z}_t^s)$, and the latents $(\mathbf{z}_t^s, \mathbf{z}_t^c)$ are jointly distributed with marginal $p(\mathbf{z}^s, \mathbf{z}^c)$. Conditioning on $\hat{\mathbf{z}}_t^s = \mathbf{a}$,

$$\begin{aligned} p(\mathbf{s}_t^p \mid \hat{\mathbf{z}}_t^s = \mathbf{a}) \;&=\; \int p(\mathbf{s}_t^p \mid \mathbf{z}^s, \mathbf{z}^c)\, p(\mathbf{z}^s, \mathbf{z}^c \mid \hat{\mathbf{z}}_t^s = \mathbf{a})\, \mathrm{d}\mathbf{z}^s\, \mathrm{d}\mathbf{z}^c \\ &=\; \int \delta\big(\mathbf{s}_t^p - m(\mathbf{z}^s)\big)\, p(\mathbf{z}^s, \mathbf{z}^c \mid \hat{\mathbf{z}}_t^s = \mathbf{a})\, \mathrm{d}\mathbf{z}^s\, \mathrm{d}\mathbf{z}^c \\ &=\; \int_{\{(\mathbf{z}^s,\mathbf{z}^c):\, h_s(\mathbf{z}^s,\mathbf{z}^c)=\mathbf{a}\}} \delta\big(\mathbf{s}_t^p - m(\mathbf{z}^s)\big)\, \mathrm{d}\mu_{\mathbf{a}}(\mathbf{z}^s, \mathbf{z}^c), \end{aligned} \tag{A12}$$

where $\mu_{\mathbf{a}}$ is the disintegration of $p(\mathbf{z}^s, \mathbf{z}^c)$ along $h_s = \mathbf{a}$. Compare (A12) with the smooth strictly positive density obtained at the InfoNCE optimum: equality of the two requires the right-hand side of (A12) to be a non-singular density in $\mathbf{s}^p$. The integral of a Dirac in $\mathbf{s}^p$ is a non-singular density

only if either (a) the level set $\{h_s = \mathbf{a}\}$ contains exactly one $\mathbf{z}^s$-value, or (b) m is constant along that level set. The latter is excluded because m is a C^1-diffeomorphism on $\mathcal{Z}^s$. Hence the level set $\{(\mathbf{z}^s, \mathbf{z}^c) : h_s(\mathbf{z}^s, \mathbf{z}^c) = \mathbf{a}\}$ is contained in $\{\mathbf{z}^s_*(\mathbf{a})\} \times \mathcal{Z}^c$ for some unique $\mathbf{z}^s_*(\mathbf{a}) \in \mathcal{Z}^s$, which means that

$$h_s(\mathbf{z}^s_t, \mathbf{z}^c_{t,1}) \;=\; h_s(\mathbf{z}^s_t, \mathbf{z}^c_{t,2}) \qquad \text{for all } \mathbf{z}^c_{t,1}, \mathbf{z}^c_{t,2} \in \mathcal{Z}^c.$$

We may therefore drop the $\mathbf{z}^c$-argument and write $h_s(\mathbf{z}^s_t, \mathbf{z}^c_t) = \tilde{h}(\mathbf{z}^s_t)$ for some map $\tilde{h} : \mathcal{Z}^s \to \hat{\mathcal{Z}}^s$. Bijectivity of $\tilde{h}$ is inherited from injectivity of h_z (proved in Part (i)) restricted to its first block, so $\tilde{h}$ is a bijection from $\mathcal{Z}^s$ onto $\hat{\mathcal{Z}}^s$.

Substituting $\hat{\mathbf{z}}^s = \tilde{h}(\mathbf{z}^s)$ and $\mathbf{s}^p = m(\mathbf{z}^s)$ into the density-matching condition above,

$$p(\mathbf{s}^p \mid \tilde{h}(\mathbf{z}^s)) \;=\; \frac{e^{\langle \tilde{h}(\mathbf{z}^s),\, m'(\mathbf{s}^p)\rangle/\tau}}{Z(\tilde{h}(\mathbf{z}^s))}\, p(\mathbf{s}^p), \tag{A13}$$

which holds for almost every $\mathbf{z}^s \in \mathcal{Z}^s$ and $\mathbf{s}^p \in \mathcal{S}^p$. Since by Step 2 the conditional $p(\mathbf{s}^p_t \mid \hat{\mathbf{z}}^s_t = \tilde{h}(\mathbf{z}^s))$ is supported at a single point $\mathbf{s}^p = m(\mathbf{z}^s)$, the population identity (A13) should be interpreted at the natural-parameter level of the underlying model: the InfoNCE-trained encoder selects $\tilde{h}$ so that the implicit model $q(\,\cdot \mid \hat{\mathbf{z}}^s)$ approaches the data conditional in the sharp-temperature limit $\tau \to 0$. In this limit, the geometric content of the matching is captured by the equality of *log-likelihood ratios*

$$\log \frac{q(\mathbf{s}^{p,1} \mid \hat{\mathbf{z}}^s)}{q(\mathbf{s}^{p,2} \mid \hat{\mathbf{z}}^s)} \;=\; \log \frac{p(\mathbf{s}^{p,1} \mid \hat{\mathbf{z}}^s)}{p(\mathbf{s}^{p,2} \mid \hat{\mathbf{z}}^s)} \quad \text{for all } \mathbf{s}^{p,1}, \mathbf{s}^{p,2} \in \mathcal{S}^p.$$

Substituting the form of q from Step 1 and writing $\mathbf{s}^{p,k} = m(\mathbf{z}^{s,k})$ for $k = 1, 2$,

$$\tfrac{1}{\tau}\big\langle \tilde{h}(\mathbf{z}^s),\, m'(m(\mathbf{z}^{s,1})) - m'(m(\mathbf{z}^{s,2}))\big\rangle \;=\; \log \frac{p(\mathbf{s}^{p,1} \mid \hat{\mathbf{z}}^s)}{p(\mathbf{s}^{p,2} \mid \hat{\mathbf{z}}^s)}. \tag{A14}$$

The right-hand side is a function of $\mathbf{z}^s$ (through $\hat{\mathbf{z}}^s = \tilde{h}(\mathbf{z}^s)$) and the pair $(\mathbf{z}^{s,1}, \mathbf{z}^{s,2})$. Symmetry of the data process with respect to anchor and positive, which are both sampled from the same per-time distribution, forces this function to be symmetric: swapping the roles of anchor and positive negates (A14) on both sides. Setting $\mathbf{z}^{s,1} = \mathbf{z}^s$ and $\mathbf{z}^{s,2}$ arbitrary, the right-hand side measures how strongly $\hat{\mathbf{z}}^s$ discriminates a positive at $\mathbf{z}^s$ from one at $\mathbf{z}^{s,2}$. The standard outcome of such symmetry analyses (see, *e.g.*, the dot-product preservation argument that underlies isometric latent recovery) is that the bilinear form $(\mathbf{z}^{s,1}, \mathbf{z}^{s,2}) \mapsto \langle \tilde{h}(\mathbf{z}^{s,1}), \tilde{h}(\mathbf{z}^{s,2})\rangle$ is determined, up to additive and multiplicative constants, by the bilinear form $(\mathbf{z}^{s,1}, \mathbf{z}^{s,2}) \mapsto \langle \mathbf{z}^{s,1}, \mathbf{z}^{s,2}\rangle$ on $\mathcal{Z}^s$. To make this concrete: consider the second-order Taylor expansion of both sides of (A14) around $\mathbf{z}^{s,1} = \mathbf{z}^{s,2}$. The left-hand side is bilinear in $(\mathbf{z}^{s,1} - \mathbf{z}^{s,2})$ and $\hat{\mathbf{z}}^s$ (via m' and the linearity of the inner product), while the right-hand side, by Taylor expansion of the data conditional, is a quadratic form in $(\mathbf{z}^{s,1} - \mathbf{z}^{s,2})$ with coefficient determined by the negative log-density curvature. Identifying the two quadratic forms gives, after re-arrangement,

$$\big\langle \tilde{h}(\mathbf{z}^{s,1}),\, \tilde{h}(\mathbf{z}^{s,2})\big\rangle \;=\; c\,\big\langle \mathbf{z}^{s,1},\, \mathbf{z}^{s,2}\big\rangle \;+\; d \qquad \forall\, \mathbf{z}^{s,1}, \mathbf{z}^{s,2} \in \mathcal{Z}^s, \tag{A15}$$

for constants $c > 0$ and $d \in \mathbb{R}$ that depend on τ, the prior $p(\mathbf{z}^s)$ and the curvature of $\log p$.

It remains to show that any C^1-map $\tilde{h} : \mathcal{Z}^s \to \hat{\mathcal{Z}}^s$ on a compact convex body $\mathcal{Z}^s \subseteq \mathbb{R}^{d_s}$ satisfying (A15) must be affine. Subtracting (A15) evaluated at $(\mathbf{z}^{s,1}, \mathbf{z}^{s,2})$ and at $(\mathbf{z}^{s,1}, \mathbf{z}^{s,3})$ gives

$$\big\langle \tilde{h}(\mathbf{z}^{s,1}),\, \tilde{h}(\mathbf{z}^{s,2}) - \tilde{h}(\mathbf{z}^{s,3})\big\rangle \;=\; c\,\big\langle \mathbf{z}^{s,1},\, \mathbf{z}^{s,2} - \mathbf{z}^{s,3}\big\rangle \qquad \forall\, \mathbf{z}^{s,1}, \mathbf{z}^{s,2}, \mathbf{z}^{s,3} \in \mathcal{Z}^s.$$

This identity says the linear functional $\mathbf{z}^{s,1} \mapsto \big\langle \tilde{h}(\mathbf{z}^{s,1}),\, \tilde{h}(\mathbf{z}^{s,2}) - \tilde{h}(\mathbf{z}^{s,3})\big\rangle$ on $\mathcal{Z}^s$ equals the linear functional $\mathbf{z}^{s,1} \mapsto c\,\langle \mathbf{z}^{s,1},\, \mathbf{z}^{s,2} - \mathbf{z}^{s,3}\rangle$. Since $\mathcal{Z}^s$ has non-empty interior, two linear functionals that agree on $\mathcal{Z}^s$ agree globally; matching them yields the existence of a linear operator $\mathbf{A} : \mathbb{R}^{d_s} \to \mathbb{R}^{d_s}$ such that, for any $\mathbf{z}^{s,2}, \mathbf{z}^{s,3} \in \mathcal{Z}^s$,

$$\tilde{h}(\mathbf{z}^{s,2}) - \tilde{h}(\mathbf{z}^{s,3}) \;=\; \mathbf{A}\,(\mathbf{z}^{s,2} - \mathbf{z}^{s,3}).$$

Fixing a reference point $\mathbf{z}^s_0 \in \mathcal{Z}^s$ and defining $\mathbf{b} := \tilde{h}(\mathbf{z}^s_0) - \mathbf{A}\mathbf{z}^s_0$, this rearranges into

$$\tilde{h}(\mathbf{z}^s) \;=\; \mathbf{A}\,\mathbf{z}^s + \mathbf{b} \qquad \forall\, \mathbf{z}^s \in \mathcal{Z}^s.$$

Plugging this back into (A15) forces $\mathbf{A}^\top \mathbf{A} = c\,\mathbf{I}$, so $\mathbf{A}/\sqrt{c}$ is orthogonal; in particular $\mathbf{A}$ is invertible. Combining with $\hat{\mathbf{z}}_t^s = \tilde{h}(\mathbf{z}_t^s)$,

$$\hat{\mathbf{z}}_t^s \;=\; \mathbf{A}\,\mathbf{z}_t^s \;+\; \mathbf{b},$$

and inverting,

$$\mathbf{z}_t^s \;=\; \mathbf{A}^{-1}\big(\hat{\mathbf{z}}_t^s - \mathbf{b}\big) \;=:\; \mathbf{A}'\,\hat{\mathbf{z}}_t^s \;+\; \mathbf{b}',$$

which is conclusion (ii). □

C More Visualizations and Analysis

Open-loop rollouts on `Square`. The `Lift` and `Can` rollouts appear in the main paper (Figure 7); the corresponding `Square` panel is shown in Figure A3. `Square` requires fitting a square peg through a tightly toleranced opening, giving a substantially longer effective horizon than `Lift` or `Can`. TC-WM preserves the contact event and final peg placement across the horizon, while the baselines blur or drift earlier.



Figure A3: Open-loop rollouts on `Square`: TC-WM vs. TD-MPC2, DreamerV3, DINO-WM. Rows show different methods with ground truth at the bottom.

Projection-architecture rollouts. Figure A4 is the qualitative counterpart of the projection-head ablation in Figure 9: the linear projection used by TC-WM yields the most stable long-horizon rollout, while MLP, VAE, and ViT projectors blur the cube or distort the gripper geometry, consistent with the SR/SSIM ranking in the bar chart.

Training dynamics across ablations and baselines. Figure A5 compares the training dynamics of TC-WM against its seven objective ablations and the closest baseline DINO-WM on `Lift`, evaluated on four uniform task-quality metrics: PSNR and SSIM on the predicted future frame, latent next-step error in the projected latent space, and the image-reconstruction MSE on the predicted frame. Removing the embedding-reconstruction term ($\mathcal{L}_{\text{rec}}$, "w/o Embed recon") is the only ablation that visibly degrades prediction quality on all four metrics, validating its role as a regulariser; the remaining ablations converge to TC-WM's level of prediction fidelity.

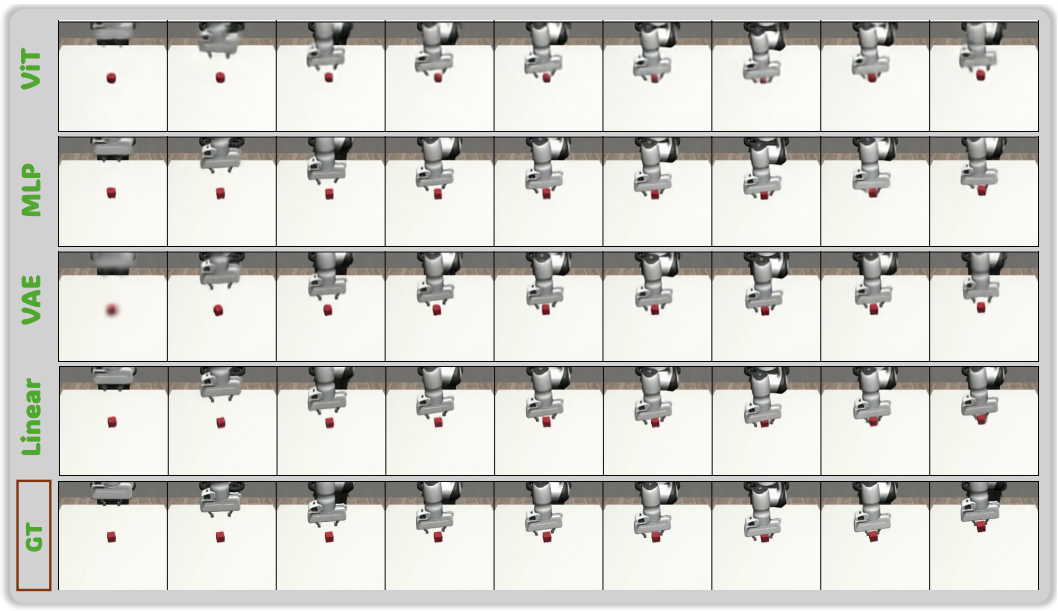


Figure A4: Open-loop rollouts on `Lift` under different projection architectures (linear vs. MLP / VAE / ViT). The linear projection of TC-WM preserves geometric structure best.

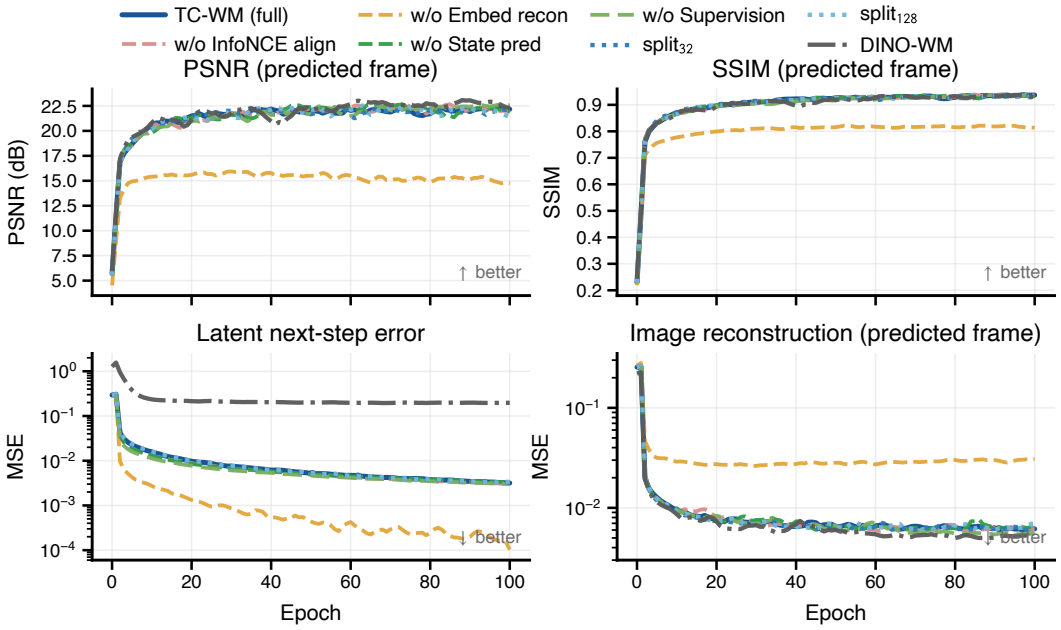
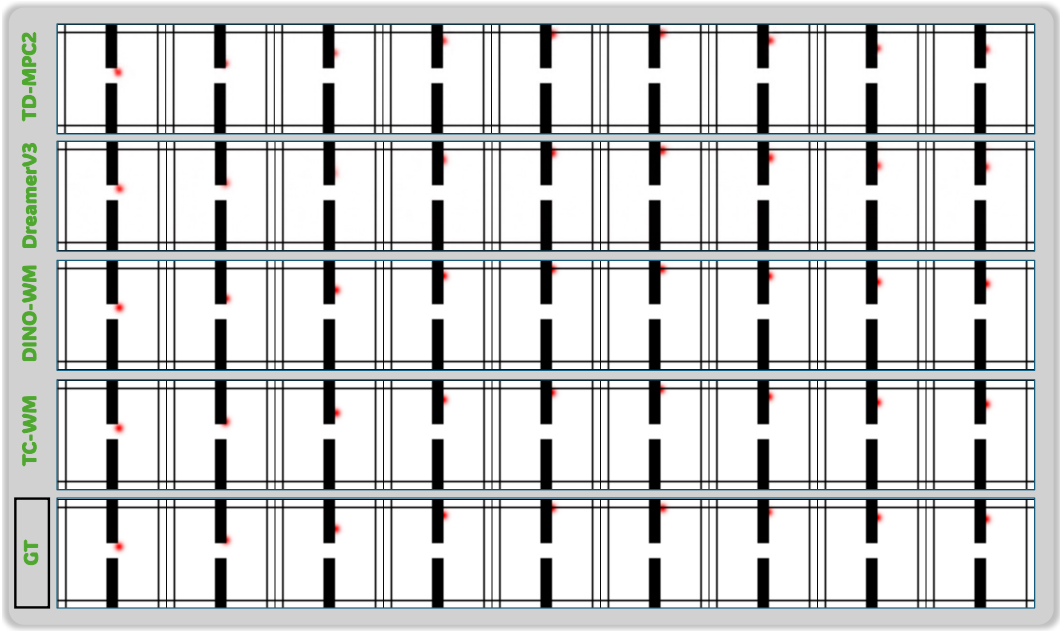
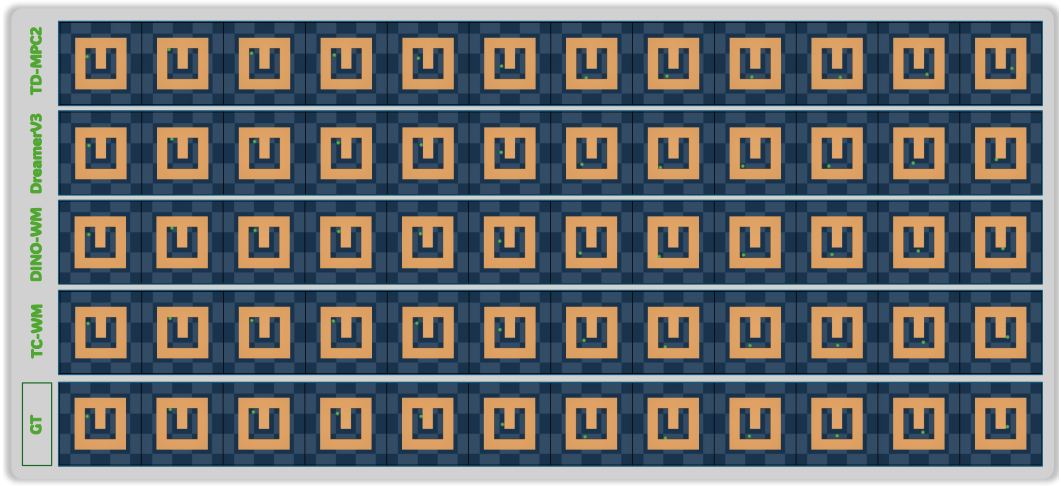


Figure A5: Training dynamics on `Lift` across TC-WM (full), seven ablations, and DINO-WM. Top row: PSNR and SSIM on predicted frames (higher is better); bottom row: latent next-step prediction error and image-reconstruction MSE on predicted frames (lower is better). Removing the embedding reconstruction term ($\mathcal{L}_{\text{rec}}$, "w/o Embed recon") is the only ablation that visibly degrades all four metrics.

Open-loop rollouts on `Wall` and `Maze` tasks. We compare offline world models by visualizing open-loop rollouts on the `Wall` and `Maze` environments (Figure A6). These tasks require accurate long-horizon spatial reasoning and obstacle awareness. Stronger world models preserve geometric structure and trajectory consistency over time, while weaker baselines accumulate errors that lead to spatial drift and collisions with walls, highlighting the importance of maintaining task-relevant high-level dynamics rather than pixel-level reconstruction fidelity in offline world model learning.



(a) `Wall`



(b) `Maze`

Figure A6: Open-loop rollouts on `Wall` and `Maze` tasks for offline world model comparison.

Open-loop rollouts of DMC at different horizons. We visualize open-loop rollouts of our method on `Reacher`, `Cheetah`, and `Hopper` across increasing rollout horizons (Figure A7). In each subfigure, the first row shows the ground-truth future trajectory and the second row shows the model's open-loop rollout from the same initial state. Across all three environments, our method closely matches the ground-truth dynamics at short horizons and remains stable as the rollout horizon increases, preserving task-relevant structure such as joint configurations in `Reacher`, locomotion phase consistency in `Cheetah`, and balance and contact dynamics in `Hopper`, which demonstrates that our world model captures long-horizon dynamics beyond pixel-level reconstruction.

D Extended Related Work

Model-based RL & World Models. Learning dynamics models is a longstanding theme in control and planning, underpinning model-based reinforcement learning and optimal control [Sutton, 1990, Astolfi, 2010, Williams et al., 2018]. By explicitly modeling environment dynamics, model-

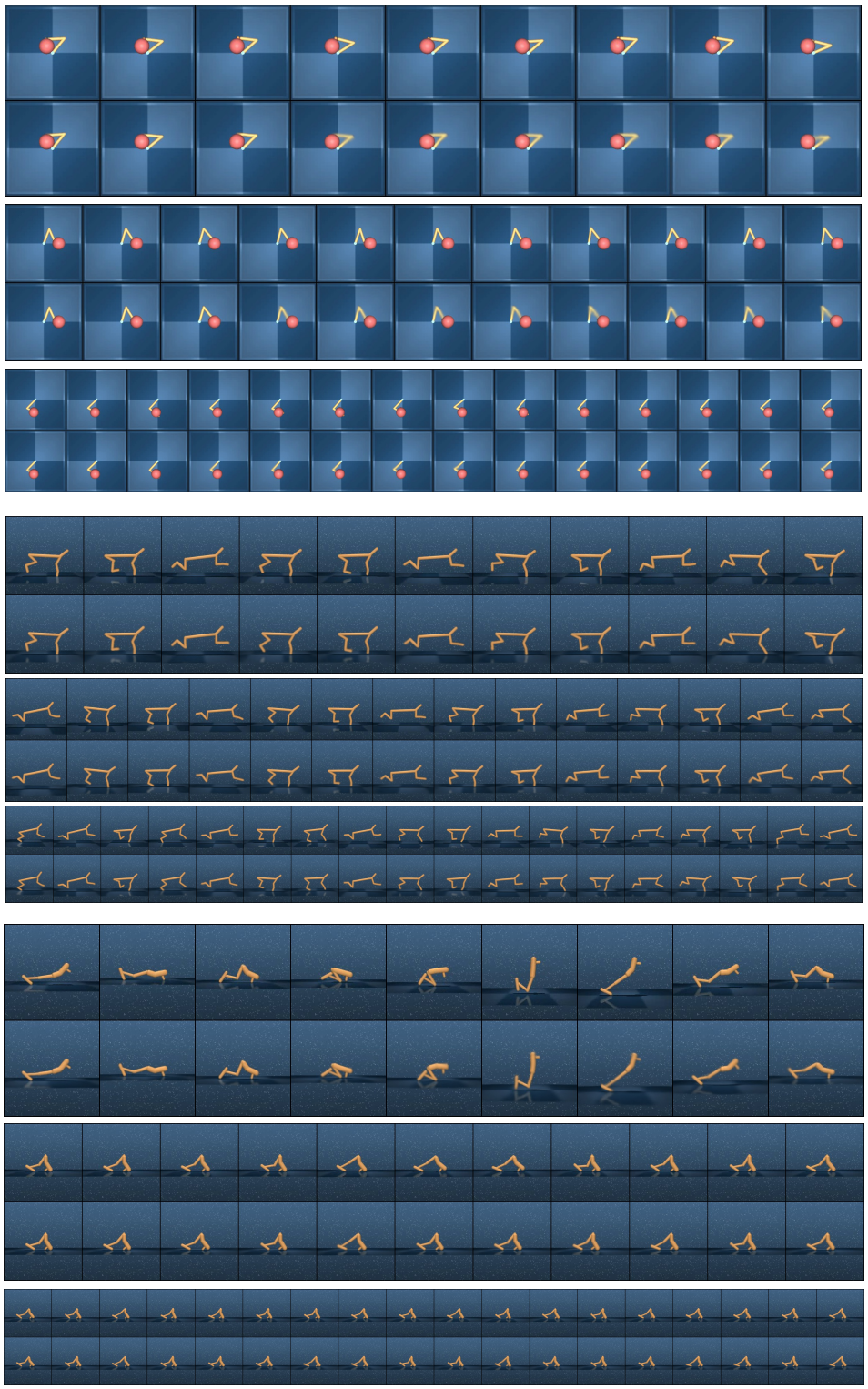


Figure A7: Open-loop rollouts of our method at different horizons on `Reacher`, `Cheetah`, and `Hopper`.

based approaches support planning and imagination, and have been shown to improve sample efficiency and generalization across embodied learning settings, including online reinforcement learning [Sekar et al., 2020], exploration [Pathak et al., 2017], planning [Watter et al., 2015, Finn and

Levine, 2017], and imitation learning [Pathak et al., 2018]. Recent works such as Dreamer [Hafner et al., 2021, 2023, 2025] and TD-MPC [Hansen et al., 2022, 2024] model compact dynamics from high-dimensional observations to build world models for MBRL. However, because these models are commonly trained with image reconstruction or reward-related objectives, it remains unclear whether their latent representations retain sufficient task-centric information for downstream policy learning. More recently, visual foundation models have been adopted as frozen encoders to provide semantically rich representations for dynamics prediction. Approaches such as DINO-WM [Zhou et al., 2025] and DINO-world [Baldassarre et al., 2025] learn world models directly in embedding spaces and demonstrate cross-domain generalization and zero-shot planning; a complementary line imposes physics-based inductive structure on the latent through Hamiltonian dynamics and symmetry priors [Tang et al., 2026], while a parallel line uses latent-aware diffusion for decision-making [Feng et al., 2026]. However, visual embeddings optimized for general semantic understanding typically do not *necessarily* provide the structured, task-centric information required for physical simulation and planning.

Task-Centric Representation Learning. Prior work from neuroscience, cognition, and modeling highlights the importance of structured latent representation for planning [Ho et al., 2022], physical inference [Rajalingham et al., 2022], and predictions consistent with neural and dynamical constraints [Mastrogiuseppe and Ostojic, 2018, Nayebi et al., 2023]. In representation learning for control, this objective is commonly formalized as learning *task-centric statistics*: minimal representations that only retain all information necessary for predicting future task-relevant states or rewards [Lesort et al., 2018, Greff et al., 2019, Schölkopf et al., 2021, Locatello et al., 2020]. Notably, TD-MPC2 [Hansen et al., 2024] explicitly frames its latent dynamics as a task-oriented representation optimized jointly with an actor–critic objective, while MuZero [Schrittwieser et al., 2020] learns value-equivalent representations that preserve only reward-relevant information. These methods learn task-sufficient representations end-to-end from reward signals, whereas TC-WM achieves task-centric structure through alignment with proprioceptive observations in a reward-free, offline setting. Motivated by this perspective, we aim to learn a latent space *on top of* visual embeddings, which are semantically powerful but contain substantial task-irrelevant information, and therefore require a *post-processing stage* for world models.

Representation Alignment. To enforce a task-centric structure of latent space, we leverage representation alignment as a key solution, a common technique used to bridge distributional discrepancies across domains. Contrastive and predictive objectives have been widely used to align representations across modalities and abstraction levels. Contrastive learning, such as InfoNCE [Oord et al., 2018] and its extensions (SimCLR [Chen et al., 2020], BYOL [Grill et al., 2020], VICReg [Bardes et al., 2021]), encourage invariant yet discriminative representations for cross-modal alignment [Radford et al., 2021], while predictive objectives enforce consistency across time or modalities [Liu et al., 2023]. In the context of world models, alignment objectives help ensure consistency between encoded observations and predicted future states [Bardes et al., 2024, Assran et al., 2025], and have also been explored for aligning information across modalities [Wu et al., 2025]. Recently, REPA [Yu et al., 2025] demonstrates that aligning intermediate representations of diffusion transformers with pretrained visual encoders yields dramatic training speedups, validating the broader principle that frozen foundation features can serve as useful semantic scaffolds. TC-WM transfers this principle from visual generation to physical prediction and planning: instead of aligning a generative model to a visual encoder only for optimization, we learn a compact dynamics state above the encoder and use proprioceptive alignment to select the portion of that state that is controllable and task-relevant. Related identifiability results study auxiliary-variable models [Khemakhem et al., 2020], contrastive representations [Zimmermann et al., 2021], and bilinear representations [Roeder et al., 2021]. Our analysis combines compact-latent recovery from temporal statistics with affine recovery of the physically aligned block under the stated assumptions.

E Limitations and Future Work

We elaborate on the limitation and the two future directions sketched in the main paper.

Limitation: training-time alignment signal. TC-WM's identifiability guarantee assumes access to a signal that exposes the controllable physical state of the system. We use proprioception, but the

same alignment loss is compatible with any low-dimensional task-centric proxy, *e.g.*, 2D/3D pose from a tracking model, depth from a monocular estimator, or cross-view consistency in a multi-camera setup. Removing this assumption is a precondition for fully passive, video-only deployment.

Future Direction A: TC-WM as a downstream alignment module. The same recipe used for foundation *visual* encoders extends to foundation *world* models. Given a large, pretrained generative or video world model, freezing the backbone and training only the TC-WM head (linear projection, alignment, embedding reconstruction) on a target-domain dataset would specialize the backbone's general-purpose dynamics for a specific embodiment, in the spirit of LoRA-style adaptation.

Future Direction B: generative world models on foundation embeddings. Existing world-model families occupy distinct trade-offs: pixel-space generative models pay a high reconstruction cost; free-latent models suffer from *latent dynamics collapse*, the predictor learns easy directions instead of physical ones; embedding-space JEPA-style models avoid both but predict in a space optimized for general semantics rather than control. A natural next step is to train a *generative* dynamics model whose target lives *on the foundation embedding manifold*, *e.g.*, a diffusion or flow-matching head over embedding-space trajectories. TC-WM-style projections at the output of such a generator would reintroduce the control-centric prior, combining the strengths of all three families.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the method, theoretical guarantee, and empirical scope across nine offline control benchmarks. The conclusion explicitly states the main limitation, namely reliance on physical alignment signals during training.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The conclusion includes a limitations paragraph discussing the need for physical alignment signals and possible extensions that distill such signals from other foundation representations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: The theory section states the assumptions used by the identifiability result and the appendix provides the proof and empirical checks of the assumptions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The main experiments section specifies the datasets, evaluation protocol, and baseline configurations; the appendix details dataset construction (Section A.1), visual-foundation-model variants (Section A.2), the CEM/SAC/LDP planning and control pipelines (Section A.3), the offline adaptations of every baseline (Section A.4), and the compute setup (Section A.6).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The code is publicly available at https://github.com/MinghaoFu/TC-WM, with environment setup and training, planning, and rollout instructions. Dataset sources, model settings, downstream configurations, and baseline adaptations are documented in Sections A.1 and A.3 to A.5.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The main experiments describe the evaluation protocol, and the appendix provides dataset construction, visual encoder details, planning procedures, and baseline implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: The main planning charts (Figure 8) report standard deviations over three seeds, and the held-out object/goal probes (Table A6) report standard errors over three seeds. The additional matched-supervision controls (Table A5) report three-seed means.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Section A.6 reports the hardware (NVIDIA H100 80 GB), per-experiment memory and wall-clock cost, and the cumulative compute used across the project, including preliminary runs that did not appear in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes].

   Justification: The work uses established simulation benchmarks and released datasets, and does not involve new human-subject experiments, sensitive personal data, or deployment in safety-critical settings.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A].

Justification: This is foundational research on offline world models using existing simulated-control and robot datasets. We do not identify a direct deployment pathway or specific societal impact beyond general progress in machine learning.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The paper does not release high-risk pretrained generative models, scraped web datasets, or assets with an apparent dual-use misuse risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: All datasets, simulators, foundation encoders, and baseline codebases used in this work are cited at first use. Each external asset is used under its original license (Apache 2.0, MIT, or its respective public-asset license), consistent with the corresponding terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes].

Justification: The public code repository includes a README, environment dependencies, training and evaluation entry points, and configuration files, and is released under the MIT license. Existing datasets and foundation models retain their original licenses.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not conduct new crowdsourcing or human-subject experiments. It uses existing benchmark data and simulation environments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: No new human-subject data collection or intervention is performed in this work.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: LLMs are not a component of the proposed method, experiments, or theoretical contribution.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.