# SPECTRAL-TARGET PHYSICAL LATENT STRUCTURING FOR JEPA-STYLE WORLD MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Latent world models have become increasingly popular as a method to predict and plan in latent space rather than pixel space. Recent architectures, such as LeWorldModel (LeWM), jointly train the encoder and predictor using regularization techniques like SIGReg to prevent representation collapse. Even with such regularization preventing representation collapse, we identify a new world model failure mode of *physical representation laziness*, particularly noted in highly dynamic environments. For these lazy cases, the learned latent states do not collapse but nonetheless fail to represent key physical properties, causing ubiquitous downstream planning failure. To resolve this issue, we propose training-time auxiliary supervision with a lightweight "Fourier auxiliary head", which enforces physically-informed structuring of the latent space with no additional inference-time cost and can be generalized to any environment. Experimentally, we show that the auxiliary head substantially improves planning success rates in dynamic environments where the baseline LeWM exhibits physical representation laziness. It also leads to modest improvements in other environments, even when the baseline does not exhibit physical representation laziness. We further observe superior planning performance being accompanied by higher latent space correlations with key physical properties, indicating both the ability of our method to physically structure latent states and the potential planning-side benefit to the learned representation being physically structured. Finally, we ablate across various training data amounts, showing that in low-data regimes, auxiliary supervision is particularly impactful in increasing success rate. These findings support the use of our Fourier auxiliary head method to improve both overall success rate and data efficiency, while avoiding representation laziness in latent world models.

## 1 INTRODUCTION

A world model for planning should preserve the parts of a scene that matter for future actions. It does not need to reproduce every pixel, but it must represent important objects and predict how their states will change. Joint-Embedding Predictive Architectures (JEPAs) follow this idea by predicting future observations in a learned latent space rather than reconstructing them in pixel space (Assran et al., 2023b). This makes the prediction more efficient and allows the model to ignore visual details that are not relevant for the underlying task.

Recent work has extended JEPAs to action-conditioned world modeling and planning. For example, LeWorldModel (LeWM) (Maes et al., 2026) jointly trains an encoder and a predictor from raw pixels using only a latent prediction loss and Sketched Isotropic Gaussian Regularization (SIGReg) (Balestriero & LeCun, 2025). The prediction loss trains the predictor to estimate the latent representation of a future observation, while SIGReg prevents the representation from collapse to a single vector by requiring the latent distribution to match an isotropic Gaussian.

However, preventing collapse does not guarantee that the encoder preserves the physical information necessary for planning (Joseph et al., 2026). There is an important difference between a *collapsed* encoder and a *lazy* encoder. A collapsed encoder produces nearly the same representation for every observation. A lazy encoder produces different representations, and therefore satisfies SIGReg, but captures only the minimum physical information needed to reduce the latent prediction loss. It



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

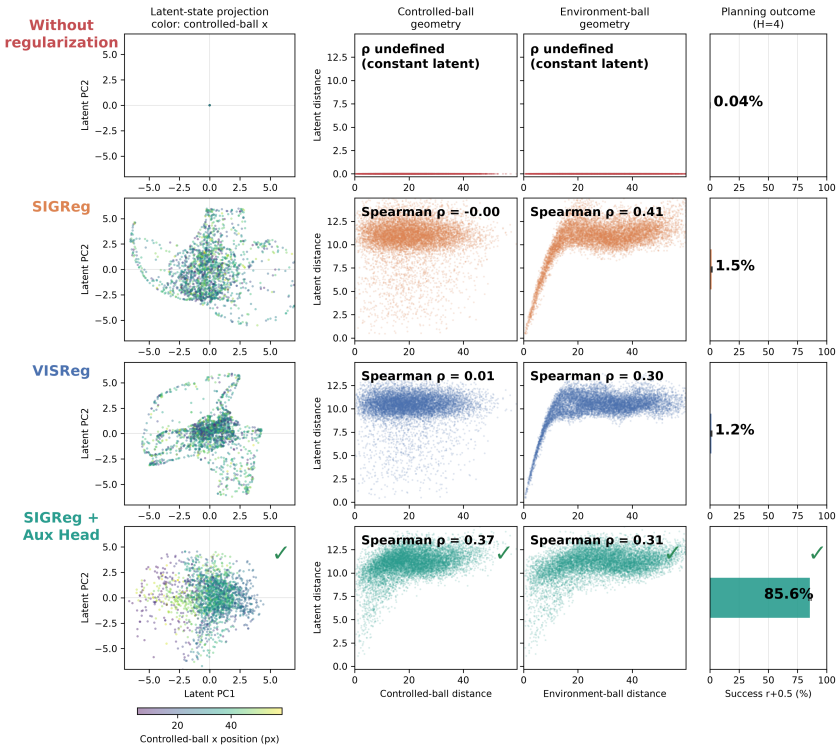


Figure 1: LeWM configurations for two-ball environment task across latent state projection (left), latent state vs position correlation (middle), and planning outcome (right). LeWM without regularization (top) leads to a collapsed representation which impedes any physical correlation or planning success. LeWM using SIGReg (second row) and VISReg (third row) both prevent representation collapse, but learn a lazy representation that only attunes to the environment ball instead of the control ball, hindering planning success. LeWM using SIGReg combined with an auxiliary head (bottom) maintains an uncollapsed representation and correlation to the control and environment ball, enabling downstream planning success.

may ignore objects or geometric relations that are important for the downstream planning task. We demonstrate such a failure directly through a case study on bouncing-ball problems in Section 3.

To address this problem, we introduce a simple auxiliary head that physically grounds the learned latent representation. During training, we assume access to the positions or bounding boxes of the objects that matter for the planning task. These annotations may come directly from a simulator or dataset, or from a lightweight object detector or tracker. The auxiliary head is a small multilayer perceptron that is trained to recover spectral features of these quantities from the encoded latent state. The same supervision can also be applied to predicted latent states, encouraging the predictor to preserve the relevant physical information over time.

This Fourier auxiliary head provides a direct learning signal that is missing from the standard JEPA objective. SIGReg asks whether the latent distribution is sufficiently diverse; the auxiliary objective asks whether this diversity contains the physical information required by the task. The two objectives therefore address different failure modes. SIGReg prevents all observations from becoming identical in latent space, while the auxiliary head prevents physically important observations from being treated as effectively identical. We provide a further theoretical discussion on the Fourier auxiliary head performance with both SIGReg and VISReg (Wu et al., 2026), a popular newer alternative to SIGReg, in the Appendix (A.2).

The proposed head is lightweight and is needed primarily during training. After training, it can be removed without changing the encoder–predictor architecture. When the planning task contains explicit rules, however, the learned head can also be retained and applied to imagined latent states.

For example, predicted object positions can be used to penalize collisions, boundary violations, or other geometric constraints during planning. Thus, the auxiliary head can improve planning in two ways: indirectly, by improving the encoder and predictor, and directly, by translating clear physical rules into planning costs. In this paper, we analyze the indirect mechanism, but briefly examine direct planning improvement in the Appendix (A.3).

Our experiments show that physical grounding through a simple auxiliary head improves the representation learned by the encoder, reduces latent prediction error, and increases downstream planning success. These improvements are particularly useful in data-limited regimes, where the standard prediction objective may not provide enough evidence for the model to discover task-relevant physical structure on its own. The auxiliary head therefore has practical value: it uses inexpensive geometric supervision to reduce the amount of interaction data needed to learn a useful world model. Overall, our results support a sharp conclusion: avoiding representation collapse is necessary, but not sufficient. A JEPA must also be guided to preserve the parts of the physical state that matter for downstream decisions.

Our main contributions are:

- We identify *physical representation laziness*, a failure mode in which SIGReg prevents global collapse, but the encoder still omits physical information required for prediction and planning.
- We introduce a lightweight Fourier auxiliary head that grounds encoded and predicted latent states using the positions or bounding boxes of task-relevant objects. Our permutation-aware formulation also handles visually indistinguishable objects without relying on arbitrary object identities or orderings.
- We show that auxiliary physical supervision improves the encoder, the latent predictor, and downstream planning success, with particularly useful benefits when training data are limited. When explicit physical constraints are available, the learned head can additionally support rule-aware planning.

## 2 Related Work

**World models.** World models learn predictive dynamics for control and planning. Early approaches learn compact latent dynamics while using pixel reconstruction or prediction to train the representation (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; Micheli et al., 2023). DreamerV3 follows this reconstruction-based approach and scales it across diverse domains (Hafner et al., 2023). In contrast, TD-MPC2 learns a decoder-free latent world model using task-specific consistency, reward, and value objectives (Hansen et al., 2024). DINO-WM also avoids pixel reconstruction, but models dynamics over features produced by a fixed pretrained DINOv2 encoder (Zhou et al., 2025).

**Joint-embedding predictive models.** Joint-embedding predictive architectures learn by predicting target representations rather than reconstructing pixels. I-JEPA and V-JEPA demonstrate the effectiveness of this principle for images and videos (Assran et al., 2023a; Bardes et al., 2024). In world models, PLDM trains latent dynamics using a multi-term variance–invariance–covariance objective (Sobal et al., 2025). LeWM instead jointly trains the encoder and predictor from pixels using a latent prediction objective and SIGReg to prevent representation collapse (Maes et al., 2026). Very recently, there is a proposal to prevent the collapse called the Variance-Invariance-Sketching Regularization (VISReg) instead of SIGReg Wu et al. (2026). Our work builds on this end-to-end setting but studies a different failure mode: the representation can remain non-collapsed while omitting physical variables required for planning.

**Extensions of LeWM.** Recent work improves LeWM along several complementary directions. Fast-LeWM predicts action prefixes to accelerate long-horizon planning, Hi-LeWM introduces hierarchical latent subgoals, and AdaJEPA adapts the model online during planning (Gao & Xu, 2026; Caselli et al., 2026; Wang et al., 2026). Other studies refine latent regularization through temporally centered SIGReg or quantile-based distribution matching (Liu et al., 2026; Yu et al., 2026). Related to Yu et al. (2025) which explains a theoretical motivation to auxiliary head training on LeWM, RC-

aux and Temporal-Distance JEPA further introduce auxiliary objectives that make latent distances more useful for planning (Li et al., 2026; Bai & Xiong, 2026).

Most closely related to our work, depth regularization supplies geometric supervision during training, while PSG-JEPA grounds representations using robot proprioception and joint-state changes (Khan, 2026; Yan et al., 2026). We study task-relevant object grounding: a lightweight auxiliary head supervises the positions or bounding boxes of important objects in both encoded and predicted states. We further consider limited-data training, scenes containing visually indistinguishable objects, and planning tasks with explicit physical rules. This provides a simple way to reduce physical representation laziness while improving prediction and downstream planning.

## 3 Beyond Representation Collapse: A Case Study of Encoder Laziness in JEPA World Models

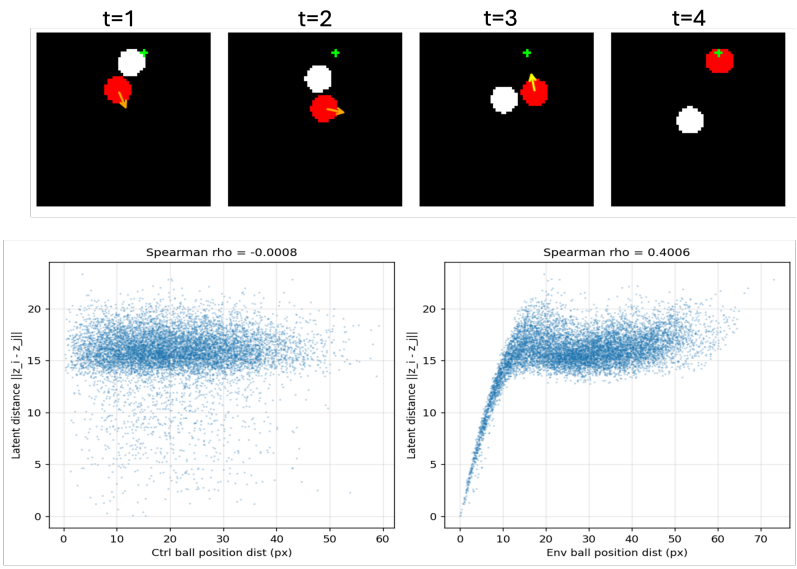


Figure 2: **Encoder laziness despite SIGReg.** Top: example frames from the two-ball task. The orange arrows indicate actions applied to the controlled ball. Bottom: latent distance versus the position difference of each ball. The latent space captures the environmental-ball position ($\rho_{\text{env}} = 0.4006$) but not the controlled-ball position ($\rho_{\text{ctrl}} = -0.0008$).

We consider a simple two-ball environment shown in Figure 2. The white environmental ball has an initial velocity, moves autonomously, and bounces off the walls. The red controlled ball moves only in response to external actions. The task is to move the controlled ball from its initial position to the green target without colliding with the environmental ball. One expects the world model to plan the actions needed to complete the task given the initial frame and final frame. Successful planning therefore requires the model to capture positions of both balls.

We train LeWM using its latent prediction loss and SIGReg. To measure which physical information is preserved, we sample pairs of observations $(o_i, o_j)$, encode them as $z_i$ and $z_j$, and compute their latent distance

$$d_z(i, j) = \|z_i - z_j\|_2.$$

For each ball $b \in \{\text{ctrl}, \text{env}\}$, we compute its position difference

$$d_b(i, j) = \|p_i^b - p_j^b\|_2$$

and report the Spearman correlation between $d_z$ and $d_b$. The nonzero and broadly distributed latent distances confirm that representation collapse does not occur. However, the near-zero correlation for the controlled ball shows that its position is not preserved in the latent geometry. The encoder is therefore non-collapsed but physically lazy: it captures the position of only one ball, although planning requires both.

This example shows that SIGReg prevents global collapse but does not determine which physical variables the encoder should preserve. Consequently, the learned latent space does not provide a reliable planning objective. Representative planning failures are included in the supplementary material. This limitation motivates the physical grounding through lightweight auxiliary head, which will be discussed in details in the following.

# 4 METHODS

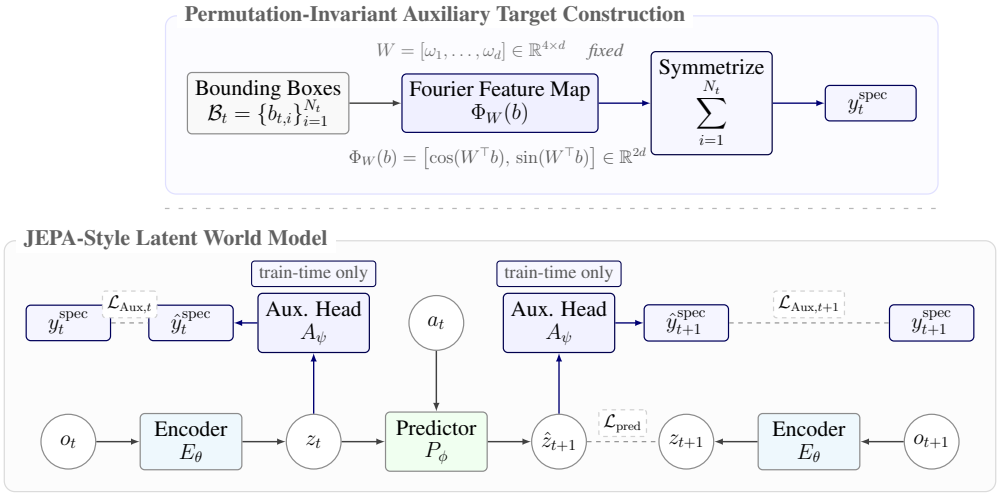


Figure 3: **Overview of the proposed Fourier-symmetrized auxiliary supervision framework.** Bounding-box states for exchangeable objects are mapped through a fixed Fourier feature map and symmetrized by summation to form the spectral target $y_t^{\text{spec}}$. During training, auxiliary heads encourage both encoded and predicted latent states to preserve this geometric information. The auxiliary heads are discarded at inference time.

## 4.1 JEPA-STYLE WORLD MODELS

JEPA-style world models, such as LeWM, are built by jointly training the entire architecture with a focus on latent representation over pixel reconstruction. They consist of an encoder ($E_\theta$) that maps a given frame observation $\sigma_t$ to latent $z_t$, and a predictor ($P_\phi$) that predicts the following latent $\hat{z}_{t+1}$ from $z_t$ conditioned by action $a_t$, i.e.,

$$z_t = E_\theta(o_t), \quad \hat{z}_{t+1} = P_\phi(z_t, a_t).$$

Naturally, the model should be trained by a prediction loss which is the mean-squared error (MSE) of the predicted and encoded latent:

$$\mathcal{L}_{\text{pred}} \coloneqq ||\hat{z}_{t+1} - z_{t+1}||_2^2. \tag{1}$$

This loss alone, however, leads to a lazy optimization of the latent known as representation collapse, where the encoder learns to map all observations into the same latent to minimize the prediction loss. To prevent this representation collapse, LeWM introduces the Sketched-Isotropic-Gaussian Regularizer (SIGReg) Balestriero & LeCun (2025); Maes et al. (2026). It projects embeddings into $M$ random unit-norm directions and optimizes the univariate Epps-Pulley test statistics (Epps & Pulley, 1983), effectively enforcing a Gaussian-distributed latent embedding. The complete training loss, with hyperparameter $\lambda_S$ is thus as follows:

$$\mathcal{L}_{\text{LeWM}} \coloneqq \mathcal{L}_{\text{pred}} + \lambda_S \text{SIGReg}(Z) \tag{2}$$

More recently, VISReg (Wu et al., 2026) extends sketching-based methods to be flexible to distribution shape, and can been used a drop-in replacement for SIGReg. Although both SIGReg and VISReg help prevent representation collapse, they do not guarantee that the learned latent representation preserves all the key information required for planning. To address this limitation, we introduce an auxiliary prediction head that infers key geometric properties of predefined, task-relevant objects directly from the latent state. We choose to use SIGReg for our method as opposed to VISReg, for reasons empirically and theoretically examined in Appendix A.2.

### 4.2 Fourier-Symmetrized Bounding Box Auxiliary Head

We use the bounding boxes of objects to represent structural information. This serves as a sparse and visually obtainable representation of object geometry, entirely in pixel space, which also analogously preserves depth information in 3D environments. Bounding boxes can also be calculated using foundation object-detection models in order to construct a ground-truth target from real visual data; as a result our method imposes little-to-no additional manual annotation or supervision footprint.

We thus define the geometric representation of the environment at time $t$ as the set of bounding-box states $\mathcal{B}_t = \{b_{t,i}\}_{i=1}^{N_t^{\text{id}}}$, constructed from the bounding boxes $N_t^{\text{id}}$ individual objects. For the $i$-th object at time $t$, each state vector $\mathbf{b}_{t,i}$ is defined as:

$$\mathbf{b}_{t,i} = [x_{t,i}, y_{t,i}, w_{t,i}, h_{t,i}] \in \mathbb{R}^4$$

where $(x_{t,i}, y_{t,i})$ denotes the box center, and $(w_{t,i}, h_{t,i})$ its width and height respectively, in pixel coordinates. When all objects are visually distinct, this state vector can in theory simply be aggregated across objects via concatenation and used as the auxiliary target. However, when there are visually similar or identical objects, the objective becomes poorly defined since arbitrarily-assigned object identity affects the aggregate state vector. To address this, we instead symmetrically aggregate objects via a Fourier sum to construct a permutation-invariant spectral representation of the bounding boxes. We first define the distribution of bounding box spatial states using point measures, for the bounding boxes of a set of visually identical objects at time $t$, i.e., $\mathcal{B}_t^{\text{id}} = \{\mathbf{b}_{t,1}, ..., \mathbf{b}_{t,N_t^{\text{id}}}\}$:

$$\mu_t(\mathbf{b}) = \sum_{i=1}^{N_t^{\text{id}}} \delta(\mathbf{b} - \mathbf{b}_{t,i})$$

where $\delta$ is the Dirac delta function [1]. $\mu_t(\mathbf{b})$ is the fundamental permutation-invariant function of $\mathbf{b}_{t,i}$'s and encodes the spatial configuration of the visually identical objects. Its Fourier-component associated with a spatial-frequency $\boldsymbol{\omega}$ is,

$$\hat{\mu}_t(\boldsymbol{\omega}) = \int_{\mathbb{R}^8} e^{i\boldsymbol{\omega}\cdot\mathbf{b}} \mu_t(\mathbf{b}) d\mathbf{b} = \sum_{i=1}^{N_t^{\text{id}}} e^{i\boldsymbol{\omega}\cdot\mathbf{b}_{t,i}} = \sum_{i=1}^{N_t^{\text{id}}} \cos(\boldsymbol{\omega}\cdot\mathbf{b}_{t,i}) + i\sin(\boldsymbol{\omega}\cdot\mathbf{b}_{t,i}),$$

and captures detailed aspects of the spatial configuration while inheriting the permutation invariance of $\mu_t(\mathbf{b})$. To obtain a discrete target, we sample this continuous function over a set of $d$ Fourier frequencies by generating a frequency matrix $W = [\omega_1, ..., \omega_d] \in \mathbb{R}^{4\times d}$ where each column $\omega_k$ represents a sampled frequency. We define the symmetrized bounding box target as:

$$y_t^{\text{spec}} := [\text{Re}(\hat{\mu}_t(W)), \text{Im}(\hat{\mu}_t(W))] = \sum_{i=1}^{N_t^{\text{id}}} \Phi_W(\mathbf{b}_{t,i}), \quad \Phi_W(\mathbf{b}) = \left[\cos(W^\top\mathbf{b}), \sin(W^\top\mathbf{b})\right] \quad (3)$$

The uniqueness theorem for Fourier transforms states that a measure is uniquely determined by its Fourier transform, i.e., $\mu_1 = \mu_2 \iff \hat{\mu}_1(\omega) = \hat{\mu}_2(\omega)\ \forall\omega \in \mathbb{R}^d$. Consequently, two different bounding box configurations for any set of objects, regardless of distinguishability, will differ by at least one Fourier component. Our method samples a finite set of frequencies so the resulting

[1] Dirac delta function satisfies the sifting property $\int_{\mathbb{R}^{d_\mathbf{b}}} f(\mathbf{b})\,\delta(\mathbf{b} - \mathbf{b}_{t,i})\,d\mathbf{b} = f(\mathbf{b}_{t,i})$ for all continuous compactly supported functions $f$.

representation $y_t^{\text{spec}}$ is not necessarily injective, but increasing $d$ yields a richer approximation of the full Fourier transform representation, reducing degeneracy among distinct configurations. As opposed to a simple pixel-space object location sum, which is also a permutation-invariant target, the Fourier target exhibits high, almost guaranteed, uniqueness.

### 4.3 Architecture of the Auxiliary head

The auxiliary head $A_\psi$ is designed as a simple 2-layer MLP that predicts the spectral targets from the latents:

$$\hat{y}_t^{\text{spec}} = A_\psi(z_t),$$

which leads to an auxiliary loss for each latent state $z_t$:

$$\mathcal{L}_{\text{Aux},t} \coloneqq ||y_t^{\text{spec}} - \hat{y}_t^{\text{spec}}||_2^2. \tag{4}$$

Propagation of the auxiliary losses results in separate loss terms for the auxiliary head, encoder, and predictor. The auxiliary head is trained only on the loss term $\mathcal{L}_{Aux,t}$, as predicting the spectral target from the predicted latent could result in instability. Hyperparameters $\lambda_{A1}$ and $\lambda_{A2}$ are used to integrate the auxiliary losses.

$$\mathcal{L}_{P_\phi} = \mathcal{L}_{\text{pred}} + \lambda_{A2}\mathcal{L}_{\text{Aux},t+1}, \quad \mathcal{L}_{E_\theta} = \mathcal{L}_{P_\phi} + \lambda_{A1}\mathcal{L}_{\text{Aux},t} + \lambda_S \text{SIGReg}(Z) \tag{5}$$

The auxiliary heads are thus used to further optimize the encoder and predictor during training time, and are removed during inference. We leave the auxiliary head as shallow as possible to offload learning to the main world model, as we notice laziness onset with slightly deeper architectures.

### 4.4 Trajectory Planning

To evaluate the performance of the models, the Cross-Entropy Method (CEM) (Rubinstein & Kroese, 2004) is used to optimize trajectories in the world model latent space, identically to the method utilized in LeWM. An initial observation $o_1$ and a randomly sampled candidate action sequence $a_{1:H}$ are used to roll out a series of latent states using the predictor, up to the planning horizon $H$. CEM iteratively samples action sequences by selecting the best plans and using their statistics to update the sampling distribution. This is used to optimize the action sequence to minimize the objective of a terminal latent goal, as shown in the equations below.

$$C(\hat{z}_H) = ||\hat{z}_H - z_g||_2^2, \quad a_{1:H}^* = \arg\min_{a_{1:H}} C(\hat{z}_H)$$

For certain environments where clear rules are provided for the planning, the auxiliary head can also be used during inference to predict geometric properties that can augment this planning objective, which we examine in the Appendix A.3.

## 5 Results

### 5.1 Evaluation Setup

We tested a variety of environments with the auxiliary head architecture to ensure its robustness across tasks and settings. The environments included 1,3, and 6 environment ball(s) versions of the ball environment (100k training) described in Section 3 as well as Push-T (500k training), Two Room (730k training), and OGBench Cube (500k training).

#### 5.1.1 Evaluation Metrics

- **Planning Success Rate.** As described in section 4.4, a standardized planning protocol is used to optimize the trajectories using each trained model in the exact same evaluation data

set. The success rates of the trajectories across environments demonstrate the utility of the auxiliary heads in training for latent-space planning. We perform planning on 5 separate seeds and average the results in the tables below.

- **Latent Distance Correlations.** To evaluate the ability of the latent space to effectively represent object locations and velocities, we sample 2000 observations and create pairs $o_1, o_2$, obtain their states $s_1, s_2$, and encode them into their latent representations $z_1, z_2$. For these pairs, we compare the difference in states between observations and the distances between the latents, and calculate the Spearman rank correlation coefficient $\rho$ to quantify the correlation. A higher $\rho$ indicates a greater ability of the latent space to represent the object states.

### 5.2 Planning Success

To evaluate the direct benefit of the auxiliary head architecture in task completion, we report planning success using the methodology described in section 4.4.

Table 1: Planning success rates for the n-ball environments at various prediction horizons. Results are mean $\pm$ standard deviation over CEM seeds 38-42. Best results are in bold.

| | $H = 4$ | | $H = 6$ | | $H = 8$ | |
|---|---|---|---|---|---|---|
| **Task** | **Ours** | **LeWM** | **Ours** | **LeWM** | **Ours** | **LeWM** |
| Bouncing Balls ($n = 1$) | $\mathbf{92.4 \pm 0.7\%}$ | $1.1 \pm 0.7\%$ | $\mathbf{78.7 \pm 0.3\%}$ | $0.2 \pm 0.4\%$ | $\mathbf{62.1 \pm 1.6\%}$ | $0.0 \pm 0.0\%$ |
| Bouncing Balls ($n = 3$) | $\mathbf{81.2 \pm 1.0\%}$ | $2.7 \pm 0.8\%$ | $\mathbf{59.4 \pm 1.2\%}$ | $0.8 \pm 0.3\%$ | $\mathbf{29.4 \pm 2.0\%}$ | $0.0 \pm 0.0\%$ |
| Bouncing Balls ($n = 6$) | $\mathbf{67.2 \pm 1.8\%}$ | $1.9 \pm 1.0\%$ | $\mathbf{43.7 \pm 0.8\%}$ | $0.1 \pm 0.2\%$ | $\mathbf{13.0 \pm 0.7\%}$ | $0.4 \pm 0.4\%$ |

Table 2: Spearman $\rho$ correlations between latent and pixel space in n-ball environments.

| | Control Ball | | Env. Ball | | All Balls | |
|---|---|---|---|---|---|---|
| **Task** | **Ours** | **LeWM** | **Ours** | **LeWM** | **Ours** | **LeWM** |
| Bouncing Balls ($n = 1$) | **0.431** | -0.010 | 0.325 | **0.423** | **0.506** | 0.247 |
| Bouncing Balls ($n = 3$) | **0.405** | -0.004 | 0.336 | **0.417** | **0.476** | 0.351 |
| Bouncing Balls ($n = 6$) | **0.380** | 0.017 | **0.275** | 0.004 | **0.377** | 0.007 |

The results indicate improvements in planning success rates across all environments from the LeWM base formulation. This is particularly evident in our n-ball environment in Table 1, where LeWM has the physical representation laziness across any number of balls. Table 2 demonstrates poor control ball positional understanding in the latent space regardless of the number of balls, which the addition of the auxiliary target fixes by better organizing the latent space. This comes at the cost of slightly worse environment ball positional understanding for $n = 1$ and $n = 3$, where the default LeWM is able to capture their positions, but overall the correlations are greatly improved across all balls.

Table 3: Planning success rates and Spearman $\rho$ correlations on other environments. Success rates report mean $\pm$ standard deviation across five CEM evaluation seeds.

| | Success Rate | | Spearman $\rho$ | |
|---|---|---|---|---|
| **Task** | **Ours** | **LeWM** | **Ours** | **LeWM** |
| Push-T | $\mathbf{95.2\% \pm 2.7\%}$ | $91.2\% \pm 4.1\%$ | **0.144** | 0.138 |
| Two Room | $\mathbf{86.0\% \pm 6.2\%}$ | $84.0\% \pm 5.1\%$ | **0.425** | 0.410 |
| Reacher | $\mathbf{89.6\% \pm 2.2\%}$ | $84.8\% \pm 5.6\%$ | 0.008 | **0.010** |
| OG-Bench Cube | $\mathbf{70.0\% \pm 7.9\%}$ | $66.4\% \pm 5.5\%$ | 0.003 | **0.020** |

In other environments, shown in Table 3, more modest but consistent increases in planning performance are reported, demonstrating the utility of this method in both 2D and 3D environments. Spearman correlation is also improved for Push-T and Two Room, though the latent spaces collapse

for Reacher and OG-Bench Cube despite having improved success rates, potentially indicating the ability of the auxiliary heads to continue improving the latent spaces in other useful ways, or the model simply being capable of 'brute-forcing' the task learning without strong intrinsic physical representation. The latter is corroborated by strong correlations for Reacher and OG-Bench Cube at lower data amounts (even 500k).

### 5.3 Data Efficiency

One of the dimensions across which we ablate the auxiliary head method is the training data. For the purposes of direct comparison with LeWM, we trained both architectures with nested datasets of various sizes to identify how our technique improves results at lower data regimes. The Push-T results in Figure 4 demonstrate that the performance improvements of the auxiliary head sustain at smaller data sizes. In real-world scenarios where data is limited, this method can still provide benefit with no inference time cost. However, it should be noted that if the data is low enough, the auxiliary target is unable to improve the baseline results due to overall underdetermination.

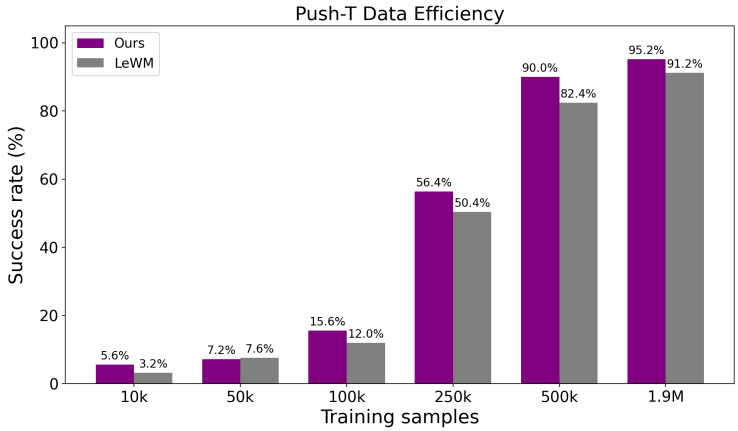


Figure 4: Push-T success rate across various training data amounts. The Fourier auxiliary head yields improvements in success rates compared to LeWorldModel in lower data regimes.

## 6 Conclusion

We introduced Fourier auxiliary heads for JEPA-style world models, a training time supervision method that organizes the latent space for better object-centric understanding of the environment without any additional inference cost. Despite SIGReg, we first show that JEPA-based encoders often fail to properly capture physical properties of key objects in the environments, then introduced a Fourier-symmetrized target derived from bounding boxes, making it generalizable to both 2D and 3D environments. We compared our method with the n-balls environment, where the laziness of the encoder in the LeWM baseline leads to poor planning success, while our Fourier auxiliary head yielded an *increase* of planning success rate by 91.3%, 78.5% and 65.3% (H=4) for 1, 3 and 6 environmental balls, respectively. Further analyzing correlations of latents with pixel-space physical properties, LeWM led to representation laziness with near-0 correlations of the control ball, while our method successfully enabled the latent space to adequately correspond to the ball position. We additionally tested our method on more common environments, where it consistently demonstrated improved task performance at full and low-data scenarios and improved latent space encoding of physical properties in certain benchmarks. While we provide a generalized formulation for the spectral target, we do not test our method on real-life environments. Additionally, we have only tested on LeWM as a baseline, whereas this method can be implemented with any JEPA-style world model. We leave examining the Fourier auxiliary head's combinations with additional supervision, regularization, and architectures, to future works.

### AI USE STATEMENT

Generative AI tools were used to help with writing and modifying code for the environment generator and the implementation of parts of the model architecture, and to improve the clarity and readability of the manuscript. The research ideas, model architecture, methodology, experimental design, and interpretation of the results were determined by the authors. All AI-assisted code was reviewed and tested by the authors, and all AI-edited text was reviewed for correctness and accuracy. The authors take full responsibility for the final content of this work.

### REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training objective, optimization settings, and evaluation protocol in the main text and appendices. The hyperparameters and implementation details used in all experiments are reported in Appendix. The source code, environment generator, experiment configurations, and trained model checkpoints will be released upon publication.

## REFERENCES


Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023a. URL https://arxiv.org/abs/2301.08243.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023b.

Jiaxin Bai and Jiaxuan Xiong. Temporal-distance JEPA: Learning planning-aligned latent geometry from offline interaction. *arXiv preprint arXiv:2607.25337*, 2026. URL https://arxiv.org/abs/2607.25337.

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

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. URL https://arxiv.org/abs/2404.08471.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020. URL https://arxiv.org/abs/2005.12872.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2026. URL https://arxiv.org/abs/2511.16719.

Niccolò Caselli, Francesco Massafra, Samuele Punzo, Salvatore Lo Sardo, Ippokratis Pantelidis, and Sathya Kamesh Bhethanabhotla. Mind the gap: Promises and pitfalls of hierarchical planning in LeWorldModel. *arXiv preprint arXiv:2607.12547*, 2026. URL https://arxiv.org/abs/2607.12547.

Thomas W Epps and Lawrence B Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983.

Yuntian Gao and Xiangyu Xu. Fast LeWorldModel: Action-prefix prediction for efficient latent world models. *arXiv preprint arXiv:2606.26217*, 2026. URL https://arxiv.org/abs/2606.26217.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

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, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. URL https://arxiv.org/abs/2301.04104.

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

Sonia Joseph, Quentin Garrido, Randall Balestriero, Matthew Kowal, Thomas Fel, Shahab Bakhtiari, Blake Aaron Richards, and Michael Rabbat. Interpreting physics in video world models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=aijGVmEG9Y.

Usman M. Khan. Depth-regularized JEPA world models learn more transferable representations from real outdoor robot data. *arXiv preprint arXiv:2607.16314*, 2026. URL https://arxiv.org/abs/2607.16314.

Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *CoRR*, abs/2111.12594, 2021. URL https://arxiv.org/abs/2111.12594.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced wasserstein distances. *Advances in neural information processing systems*, 32, 2019.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. doi: https://doi.org/10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Predictive but not plannable: RC-aux for latent world models. *arXiv preprint arXiv:2605.07278*, 2026. URL https://arxiv.org/abs/2605.07278.

Chang Liu, Fei Suo, Yanzhou Jin, Yusuke Iwasawa, Yutaka Matsuo, and Yaonan Zhu. Temporally centered SIGReg for multi-task joint-embedding predictive world models. *arXiv preprint arXiv:2607.26924*, 2026. URL https://arxiv.org/abs/2607.26924.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *CoRR*, abs/2006.15055, 2020. URL https://arxiv.org/abs/2006.15055.

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

Claudia Malzer and Marcus Baum. A hybrid approach to hierarchical density-based cluster selection. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 223–228. IEEE, 2020. doi: 10.1109/mfi49285.2020.9235263. URL http://dx.doi.org/10.1109/MFI49285.2020.9235263.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

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

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

Xudong Wang, Ishan Misra, Ziyun Zeng, Rohit Girdhar, and Trevor Darrell. Videocutler: Surprisingly simple unsupervised video instance segmentation, 2023. URL https://arxiv.org/abs/2308.14710.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. AdaJEPA: Test-time adaptation for joint-embedding predictive world models. *arXiv preprint arXiv:2606.32026*, 2026. URL https://arxiv.org/abs/2606.32026.

Haiyu Wu, Randall Balestriero, and Morgan Levine. Visreg: Variance-invariance-sketching regularization for jepa training. *arXiv preprint arXiv:2606.02572*, 2026.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization, 2019. URL https://arxiv.org/abs/1911.07013.

Haodong Yan, Jiaguan Zhu, Mingyuan Jia, Ruiqing Yin, Junjie He, Zhide Zhong, Junfeng Li, Jinxuan Lu, Hengtao Li, Tianran Zhang, Jiayi Chen, Wenxuan Song, Wen Chen, Yuxiang Gao, and Haoang Li. Is forward prediction enough? physical state grounding for JEPA world models. *arXiv preprint arXiv:2608.06799*, 2026. URL https://arxiv.org/abs/2608.06799.

Jiacan Yu, Siyi Chen, Mingrui Liu, Nono Horiuchi, Vladimir Braverman, Zicheng Xu, Dan Haramati, and Randall Balestriero. Why and how auxiliary tasks improve jepa representations. *arXiv preprint arXiv:2509.12249*, 2025.

Zhoushun Yu, Xiaoyu Hu, and Xiangyu Xu. QQWorld: Quantile-based distribution matching for joint-embedding predictive world models. *arXiv preprint arXiv:2607.28415*, 2026. URL https://arxiv.org/abs/2607.28415.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2411.04983.

# A Appendix

## A.1 Training and Evaluation Pipeline Configuration

There are two variations of the architecture: one used for the n-balls environment, and another used for the remaining environments, which is kept as close to the original LeWM implementation as possible.

### A.1.1 N-Balls Architecture

Rendered RGB frames are input at $64 \times 64$ resolution, and normalized between $[-1, 1]$. A batch size of 64 is used, and all models are trained for 200 epochs. 4 sub-trajectories are used for training, with 4 observations and 4 actions each. No frameskip is used.

**Encoder Architecture**. A small custom vision transformer is used, with a patch size of 8, token width of 64, and 4 transformer blocks with 4 attention heads. The feedforward networks (FFNs) used are 256-D, with one learnable CLS token that is projected to a 128-D output latent.

**Predictor Architecture**. The predictor is implemented as a small, action-conditioned causal transformer. It takes in 3 history frames as input, using a 128-D latent that is projected to a dimension of 64. Like the encoder, it uses 4 transformer blocks with 4 attention heads and 256-D FFNs, but includes Adaptive Layer Norm (AdaLN) Xu et al. (2019) for action conditioning. It also uses causal masking to prevent future frames from attending to the previous frame. The output is projected back to 128 dimensions.

**Auxiliary Head Architecture**. The auxiliary heads are implemented as simple, 2-layer MLPs, where the first layer projects the 128-D latent into a 64-D feature, and the second layer outputs the 18-D spectral target. We set $\lambda_{\text{Aux}} = 0.1$, and set $\lambda_{\text{SIGReg}} = 0.09$, matching the value used in both official SIGReg and VISReg repositories.

**Planning Protocol**. We use the Cross-Entropy Method for planning. We sample 100 candidate action sequences, retain 10 elites per iteration, and run it for 20 iterations. The standard deviation of the action selection is set to 2.0. 200 episodes are evaluated per seed. Additionally, the norm of the actions are clipped to 3.0, limiting the speed of the control ball. The planning horizons are set to 4, 6, and 8 for the experiments, with no replanning. The initial states are set to exactly 4, 6, or 8 steps behind a target respectively, and thus success is defined as when the control ball is within a 1-pixel radius of the true target position.

### A.1.2 LeWM Architecture

For the other environments, the architecture closely resembles LeWM. Rendered RGB frames are input at $224 \times 224$ resolution, and normalized between $[0, 1]$. A batch size of 128 is used, and all models are trained for 10 epochs. A frameskip of 5 is used, meaning 5 actions are predicted for every frame. 4 sub-trajectories are used for training, with 4 observations and 4 blocks of 5 actions each.

**Encoder Architecture**. A Vision Transformer Tiny (ViT-Tiny) model from Hugging Face is used with a patch size of 14 for the encoder. The 192-D CLS token is used as the latent.

**Predictor Architecture**. A Vision Transformer Small (ViT-Small) model from Hugging Face is used as backbone for the predictor. Causal masking and learned position embeddings are used to accommodate a history length of 3.

**Auxiliary Head Architecture**. The auxiliary heads are implemented as 2-layer MLPs, where the first layer projects the 192-D latent into a 64-D feature, and the second layer outputs the 18-D spectral target. We set $\lambda_{\text{Aux}} = 0.1$, and set $\lambda_{\text{SIGReg}} = 0.09$, matching the value used in both official SIGReg and VISReg repositories.

**Planning Protocol**. Identically to LeWM, we use the Cross-Entropy Method for planning. We sample 300 candidate action sequences, retain 30 elites per iteration, and run it for 30 iterations. A variance scale of 1.0 is used for action sampling. 50 scenes are evaluated per seed, and the standard LeWM action scaler is used to bind actions. Model Predictive Control (MPC) is conducted using

CEM, with replanning every 5 blocks of 5 actions, consisting of 25 actions altogether. A total action budget of 50 steps is given for each environment, though they can be solved in only 25 steps.

### A.1.3 Target Creation

The bounding box targets are well-defined for each environment. For the n-ball environments, all the balls are used as targets. For the remaining environments, only dynamic objects are selected. These bounding boxes are not selected with zero-shot methods, but rather obtained as ground-truth from the environment simulators, demonstrating an ideal use-case of auxiliary supervision. Samples for each environment can be seen in Figure 5.

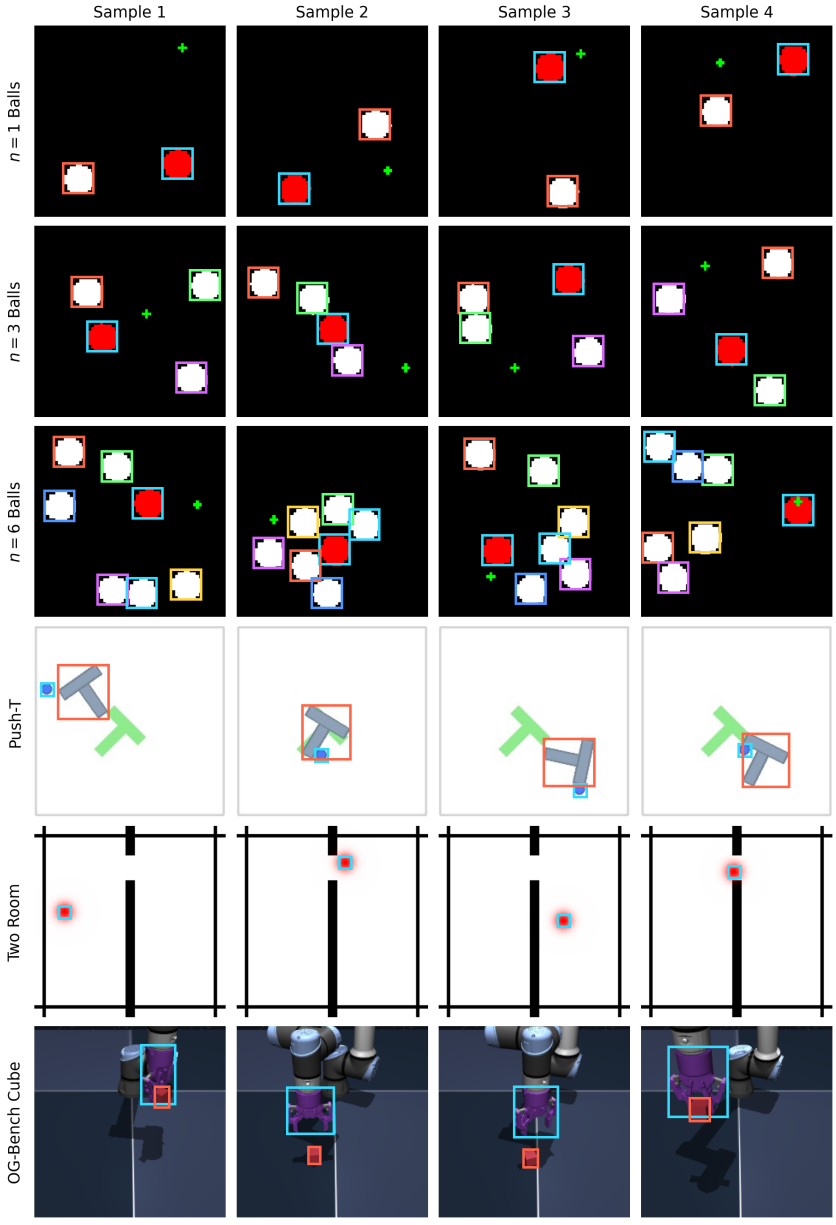


Figure 5: Sample bounding box targets for each tested environment.

## A.2 THEORETICAL EXAMINATION OF VISREG + AUX HEAD

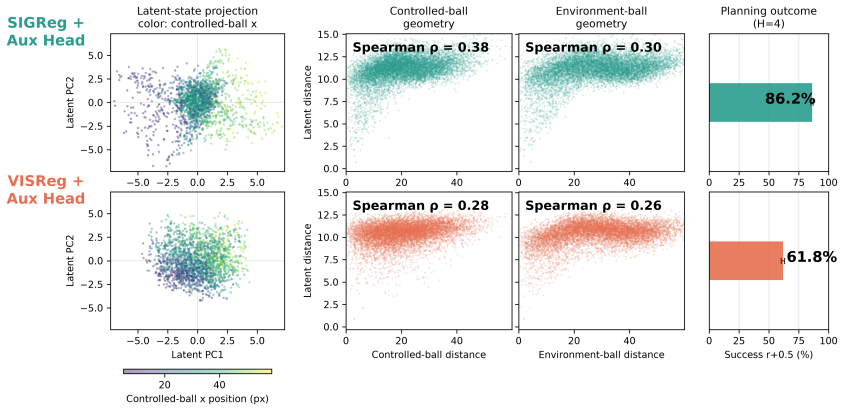


Figure 6: LeWM configurations for two-ball environment task across latent state projection (left), latent state vs position correlation (middle), and planning outcome (right). Our auxiliary head method performs worse with VISReg than with SIGReg.

Interestingly, our Fourier auxiliary head training empirically yields improved planning performance with SIGReg (Balestriero & LeCun, 2025) instead of VISReg (Wu et al., 2026), even though VISReg exhibits higher performance than SIGReg on the standard LeWorldModel. This implies that our Fourier method's benefits are not necessarily entirely additive with those of VISReg's. To examine why, we explore the theoretical relationship between VISReg and our Fourier method. VISReg utilizes the Sliced Wasserstein Distance (SWD) (Kolouri et al., 2019) which is reliant on Radon transform injectivity via the Cramér–Wold Theorem:

**Lemma 1** (Cramér–Wold Theorem)**.** *Let $\mu$ and $\nu$ be probability measures on $\mathbb{R}^d$, and let*

$$\mathcal{R}\mu(\theta, t) = \int_{\mathbb{R}^d} \delta(t - \langle x, \theta \rangle)\, d\mu(x)$$

*denote the Radon transform for $\theta \in \mathbb{S}^{d-1}$. Then $\mathcal{R}$ is injective, and*

$$\mu = \nu \quad \Longleftrightarrow \quad \mathcal{R}\mu(\theta, \cdot) = \mathcal{R}\nu(\theta, \cdot) \quad \forall \theta \in \mathbb{S}^{d-1}.$$

This is utilized in VISReg via sampling 1D directions $\theta_k$, and attempting:

$$\forall w_k : \mathcal{R}\mu_z(w_k, \cdot) \to \mathcal{N}(0, 1)$$

This means that across randomly sampled directions $w_k$, VISReg takes the Radon projection $\mathcal{R}\mu_z(w_k, \cdot)$ along these directions and attempts to enforce Gaussian shape and unit variance in the zero-loss limit. Since the Radon transform is the full collection $\mathcal{R}\mu_z(\theta, t)$ across all directions, VISReg stochastically regularizes the latent distribution's Radon transform.

To connect to how this regularization principle relates to our Fourier method, the Radon transform is in fact related to the Fourier transform via the Fourier slice theorem:

**Lemma 2** (Fourier Slice Theorem)**.** *Let $f \in L^1(\mathbb{R}^d)$, and let*

$$\mathcal{R}f(\theta, t) = \int_{\mathbb{R}^d} f(x)\, \delta(t - \langle x, \theta \rangle)\, dx$$

*denote its Radon transform for $\theta \in \mathbb{S}^{d-1}$. Then the one-dimensional Fourier transform of the Radon projection $\mathcal{R}f(\theta, \cdot)$ satisfies*

$$\mathcal{F}_t \left[\mathcal{R}f(\theta, t)\right](\omega) = \hat{f}(\omega\theta), \qquad \omega \in \mathbb{R},$$

*where*

$$\hat{f}(\xi) = \int_{\mathbb{R}^d} f(x) e^{-i\langle \xi, x \rangle}\, dx$$

*is the $d$-dimensional Fourier transform of $f$. Equivalently,*

$$\mathcal{R}f(\theta, t) = \mathcal{F}_\omega^{-1} \left[\hat{f}(\omega\theta)\right](t).$$

To get a 1D Radon projection, we take a radial slice through the Fourier transform and take a 1D inverse Fourier transform to bring it back to real space. VISReg samples individual Radon projections, equivalent to these radial slices inverse-transformed, and enforces each to have unit variance and be Gaussian. This relates to our method since VISReg is essentially trying to make the Fourier Transform of the latent distribution isotropic via radial symmetry. Our method, in contrast, targets spectral *anisotropy*, reflective of the spatial distribution, and supervises the latent to reflect this anisotropy.

$$\text{VISReg in Radon space} \xleftrightarrow{\text{Fourier Slice Theorem}} \text{isotropy in spectral space}$$

$$\text{Aux. supervision in spatial state-space} \xleftrightarrow{\text{Fourier-feature Transform}} \text{anisotropy in spectral space}$$

We therefore hypothesize that our method works better with SIGReg since the SIGReg regularization gradient tends to be weak for low variance regimes, whereas VISReg is explicitly designed to impose a strong corrective gradient along sampled low-variance directions. As a result, SIGReg's weak gradient corrections thus may permit task-induced latent anisotropy to persist. This is ordinarily a weakness that VISReg targets, but our Fourier method may actually benefit from this.

### A.3 AUXILIARY HEAD IN CEM PLANNING

As discussed earlier, the auxiliary head can be used both indirectly in planning by forcing the encoder to be spatially aware, and directly by physically informing the CEM during planning. We examine the direct method in this section, where the predicted physical state of the auxiliary head can be used to impose physicallly-grounded task-specific constraints on imagined latent trajectories. We choose to examine this setting in the $n$-ball environments, where a ball must be controlled to reach a target while under the constraint of avoiding collision with moving environment balls.

The baseline LeWorldModel planning algorithm uses CEM to minimize the latent distance between an action sequence and a goal.

$$C_{\text{latent}} = \|\hat{z}_H - z_g\|_2^2$$

While the main method explored in our paper improves the physical information present in latent $\hat{z}_H$, thus indirectly physically grounding the CEM, this only places a soft constraint on collision avoidance. As an example, consider two configurations which are visually and spatially similar, but one configuration is near-collision and the other is colliding. Since these configurations are spatially similar, they may be close in latent space, but the impact on CEM success is stepwise. The planner receives a smooth objective from this latent loss, and does not necessarily place a hard constraint on rule-violating trajectories. To address this, we incorporate the auxiliary head prediction during the planning phase. First we reconstruct object locations from the auxiliary objective using a generalized inverse reconstruction map $\Psi_W : y_t^{\text{spec}} \mapsto \{\hat{b}_{t,i}\}_{i=1}^{N}$:

$$\left(\hat{p}_t^{\text{ctrl}}, \{\hat{p}_{t,j}^{\text{env}}\}_{j=1}^{N_{\text{env}}}\right) = \Psi_W A_\psi(\hat{z}_t)$$

For simplicity in this experiment, we directly train the auxiliary head to predict the positions of the balls rather than the Fourier signature, so $\Psi_W = \mathrm{Id}$. For the Fourier auxiliary head, $\Psi_W$ can either be a simple learned decoder trained using synthetic mapping data, or an algorithmic minimization of plausible Fourier sums. Once the centers of the objects are recovered from the auxiliary head, for balls of radius $r$, we penalize any predicted control ball–environment ball pair whose center separation falls below the collision distance $2r$,

$$C_{\text{coll}} = \sum_{t=1}^{H} \sum_{j=1}^{N_{\text{env}}} \mathrm{ReLU}\Big(2r - \big\|\hat{p}_t^{\text{ctrl}} - \hat{p}_{t,j}^{\text{env}}\big\|_2\Big)^2$$

The auxiliary-informed CEM objective thus becomes

$$C_{\text{PI-CEM}} = \|\hat{z}_H - z_g\|_2^2 + \lambda_{\text{coll}} C_{\text{coll}}.$$

This collision term is solely derived from the model's imagined latent trajectory and auxiliary predictions; no ground-truth ball positions are provided to CEM during planning. Therefore, the auxiliary head provides a predictive and robust way of imposing the known collision constraint on latent-space planning sequences.

Table 4: Effect of auxiliary-head-informed CEM in the $n$-ball environments using 100K training samples. Values are mean ± standard deviation over CEM seeds 38–42. Lower final distance is better; higher collision-free and success rates are better.

| | | **Final Dist.** ↓ | | **Collision-Free** ↑ | | **Success** $(r+1)$ ↑ | |
|---|---|---|---|---|---|---|---|
| **Env. Balls** | **Samples** | **CEM** | **Aux-CEM** | **CEM** | **Aux-CEM** | **CEM** | **Aux-CEM** |
| $n = 1$ | 100K | $5.029 \pm 0.041$ | $\mathbf{4.907 \pm 0.015}$ | $95.1\% \pm 0.4\%$ | $\mathbf{98.7\% \pm 0.6\%}$ | $77.8\% \pm 0.4\%$ | $\mathbf{80.9\% \pm 0.4\%}$ |
| $n = 3$ | 100K | $\mathbf{5.921 \pm 0.024}$ | $5.948 \pm 0.044$ | $80.5\% \pm 0.6\%$ | $\mathbf{93.2\% \pm 1.0\%}$ | $58.6\% \pm 0.9\%$ | $\mathbf{62.3\% \pm 1.1\%}$ |
| $n = 6$ | 100K | $\mathbf{7.801 \pm 0.103}$ | $8.169 \pm 0.101$ | $71.1\% \pm 1.2\%$ | $\mathbf{84.1\% \pm 1.2\%}$ | $\mathbf{21.0\% \pm 1.2\%}$ | $20.0\% \pm 1.7\%$ |

We see that the auxiliary-informed CEM improves collision avoidance for all experiments, especially when the number of balls increases. For these higher $n$ environments, the soft collision avoidance in CEM may not be enough to push the selection of imagined trajectories away from a collision sequence. This improvement comes at a slight tradeoff; while terminal distance is virtually similar for $n = 1$, for the higher $n$ environments the auxiliary-informed CEM ends up farther from the

target location than the baseline CEM on average, and the success rate decreases slightly, as is seen for $n = 6$. Despite this, the collision avoidance provides enough benefit such that overall success rates are higher for $n = 1$ and $n = 3$ with the auxiliary CEM. We thus show in this experiment that the auxiliary objective can be used both in training and in planning to penalize constraint- or physically-invalid imagined latent trajectories.

### A.4 TARGET ABLATIONS

One key ablation is the choice of target for the auxiliary head prediction. These include:

- **Exact.** The exact environment-specific physical state available from the simulator.
- **Compact7.** A generalized set of values for each object in the environment: $b = [p_{\text{obj}}, x, y, v_x, v_y, \sin(\theta), \cos(\theta)]$. $p_{\text{obj}}$ is a boolean flag indicating whether the object is currently visible in frame, and any quantities that do not apply are masked when the loss is calculated, often used in bounding box detection models such as DETR (Carion et al., 2020).
- **Compact13.** An expanded version of Compact7 intended for 3D environments, with roll $\phi$, pitch $\psi$, and yaw $\theta$:
  $b = [p_{\text{obj}}, x, y, z, vx, vy, vz, \sin(\phi), \cos(\phi), \sin(\psi), \cos(\psi), \sin(\theta), \cos(\theta)]$.
- **Bounding Boxes.** Our paper's target, regular bounding boxes.
- **Oriented Bounding Boxes ($\theta$).** Bounding boxes can also be detected in an oriented fashion, with an additional detected angle $\theta$. This changes the target to $b = [x, y, w, h, \sin(\theta), \cos(\theta)]$.
- **Oriented Bounding Boxes ($2\theta$).** Since bounding boxes are rectangular, it is symmetric over a rotation of $\pi$. Therefore, using $2\theta$ can prevent instability from this rotational symmetry. The target is $b = [x, y, w, h, \sin(2\theta), \cos(2\theta)]$.
- **Bounding Boxes + Center of Mass.** Given that bounding boxes are available for the objects, it can be assumed that semantic masks may also be extracted in an unsupervised manner. This can be used to calculate a center of mass that acts as a proxy for the orientation of the object. The target is therefore $b = [x, y, w, h, COM_x, COM_y]$.
- **Oriented Bounding Boxes ($\theta$) + Center of Mass.** We also add the COM to the oriented bounding box target, making it $b = [x, y, w, h, \sin(\theta), \cos(\theta), COM_x, COM_y]$.
- **Oriented Bounding Boxes ($2\theta$) + Center of Mass.** We can also add the COM to the $2\theta$ oriented bounding box target, making it $b = [x, y, w, h, \sin(2\theta), \cos(2\theta), COM_x, COM_y]$.

| Target | Push-T | OG-Bench Cube | 3-Balls (H=4) |
|---|---|---|---|
| Exact | 93.6% ± 3.8% | 71.2% ± 6.6% | 77.2% ± 0.8% |
| Compact7 | 89.6% ± 4.3% | 70.4% ± 3.6% | 55.0% ± 0.9% |
| Compact13 | – | **74.8% ± 3.6%** | – |
| BB | **95.2% ± 3.3%** | 70.4% ± 6.1% | **81.2% ± 1.0%** |
| OBB ($\theta$) | 94.0% ± 3.2% | 71.2% ± 6.1% | 66.1% ± 0.9% |
| OBB ($2\theta$) | 94.0% ± 4.0% | 68.4% ± 4.3% | 77.5% ± 0.9% |
| BB + COM | 93.6% ± 2.2% | 69.6% ± 3.6% | 77.3% ± 1.8% |
| OBB ($\theta$) + COM | 93.6% ± 3.0% | 67.2% ± 4.6% | 73.2% ± 1.1% |
| OBB ($2\theta$) + COM | 93.2% ± 1.8% | 68.4% ± 6.8% | 76.3% ± 1.0% |

Table 5: Auxiliary target ablation results. Results report mean ± standard deviation across five CEM evaluation seeds.

The main conclusion from the table is that different targets achieve comparable success rates, demonstrating the robustness and broad applicability of the auxiliary-head method. More specifically, for the above 2D environments, Push-T and 3-Balls, it is clear that the bounding box target provides the highest success rates. Additional information from oriented bounding boxes or semantic mask centers, or even privileged environment information in the Exact and Compact7 representations do not improve results. For OG-Bench Cube, the 3D representations of the Compact13 target yield the highest success rate, indicating that the encoder is able to capture 3D information when supervised using an appropriate auxiliary head. Even though the 2D bounding box target does change in size and therefore captures a proxy of object depth, it is unable to structure the latent as effectively as explicit 3D state supervision. Regardless, this information may not be available to the model at either training or inference time, and therefore cannot be generalized effectively.

### A.5 AUX HEAD ARCHITECTURE ABLATIONS

The final method uses a shared auxiliary head on both the decoded and predicted latents. To isolate the impact of the predictor-side auxiliary head, we evaluate a variant that removes its auxiliary loss, leaving only encoder-side supervision. The encoded loss is simplified accordingly.

$$\mathcal{L}_{P_\phi} = \mathcal{L}_{\text{pred}}, \quad \mathcal{L}_{E_\theta} = \mathcal{L}_{\text{pred}} + \lambda_{A1}\mathcal{L}_{\text{Aux},t} + \lambda_S \text{SIGReg}(Z)$$

The results indicate that the gradients from the auxiliary head into the predictor enable it to predict better structured latents, improving planning performance over using a single auxiliary head. However, using a single auxiliary head still shows significant improvement over the baseline LeWM success rates by better structuring latents and preventing collapse in the n-balls environment.

| **Architecture** | **Push-T** | **OG-Bench Cube** | **3-Balls (H=4)** |
|---|---|---|---|
| Both Aux Heads | **95.2% ± 2.7%** | **70.0% ± 7.9%** | **81.2% ± 1.0%** |
| Encoded Latent Only | 94.4% ± 1.7% | 69.6% ± 4.6% | 73.6% ± 1.5% |
| LeWM | 91.2% ± 4.1% | 66.4% ± 5.5% | 2.7% ± 0.8% |

Table 6: Performance comparison of auxiliary head architectures. Results report mean ± standard deviation across five CEM evaluation seeds.

### A.6 Object Matching Ablations

The fourier-symmetrization is designed to prevent the arbitrary order of auxiliary target predictions of visually similar or identical objects from being incongruent with the ground truth order and incorrectly penalizing the model. We ablate this by removing the fourier-symmetrization altogether, and by implementing the Hungarian algorithm for target-object matching (Kuhn, 1955).

The Hungarian algorithm is used on the set of $N$ objects to find a permutation $\sigma \in \mathfrak{S}_N$ that minimizes cost:

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{match}(b_{i,t}, \hat{b}_{\sigma(i,t)})$$

Where $\mathcal{L}_{match}$ is a pair-wise matching cost calculated similarly to $\mathcal{L}_{Aux_t}$ on the ground truth and predicted targets. This optimized permutation is then used to calculate the auxiliary loss.

Two variants of the Hungarian algorithm matching are used. First, a frame-wise matching where the algorithm is used on every frame. Second, a sequence-wise matching where identities are assigned at the beginning of each 4-frame training sequence.

| **Matching** | **Push-T** | **OG-Bench Cube** | **3-Balls (H=4)** |
|---|---|---|---|
| No Matching | $94.8\% \pm 1.8\%$ | $66.0\% \pm 3.7\%$ | $76.8\% \pm 0.9\%$ |
| Frame Hungarian | $\mathbf{96.4\% \pm 1.7\%}$ | $65.2\% \pm 4.1\%$ | $1.7\% \pm 0.6\%$ |
| Sequence Hungarian | $94.4\% \pm 1.7\%$ | $64.0\% \pm 6.2\%$ | $2.6\% \pm 1.1\%$ |
| Fourier | $95.2\% \pm 2.7\%$ | $\mathbf{70.0\% \pm 7.9\%}$ | $\mathbf{81.2\% \pm 1.0\%}$ |

Table 7: Comparison of object-matching strategies. Results report mean $\pm$ standard deviation across five CEM evaluation seeds.

On Push-T, all variants achieve similarly high success rates. Since LeWM already performs strongly on this task, its performance is likely close to saturation, leaving limited room for further improvement. Consequently, the specific architecture of the auxiliary head has only a minor impact, which is also consistent with the modest gain (SR $+4\%$) observed when adding an auxiliary head to LeWM. Interestingly, Hungarian matching even underperforms no-matching for OG-Bench Cube, and 3-Balls collapses despite being an environment where visually similar objects are present. The fourier symmetrization consistently gives better results than no matching at all.

Figure 7 demonstrates how object assignment instability results in poorer performance for Hungarian algorithm-based matching.

- **Assignment differs from final** counts frames where the object assignment is different from the final stable order. Here, frame-level matching takes significantly longer to stabilize than sequence-level matching.
- **Assignment changed since prior epoch** indicates assignment flips from the previous epoch. Both frame-level and sequence-level matching shows instability.
- **Clips with framewise optimum flip** measures clips where assignment changes between consecutive frames, which occurs for both matching strategies throughout, demonstrating early-training stability.
- **Predicted optimum equals encoder assignment** indicates clips where the encoded latent and predicted latent have the same object assignment. Instability in frame-level matching demonstrates poor latent structuring and predictor performance in early training stages, where the predictor is unable to carry forward the understanding of the target objects from the encoded latent to the next predicted latent.

The improvement from the fourier symmetrization for the full-data results can be explained by the object order-invariant target preventing assignment instability in early training. Object assignments stabilize in later epochs, especially at full data, allowing performance to approach that of the symmetrized target, but its value is particularly notable at low-data scenarios such as in Figure 7.




Figure 7: Frame-level and sequence-level Hungarian matching diagnostics on Push-T at 250k data points for the first 4 epochs.

### A.7 Depth Ablations

For 3D environments, depth may also be available using open-source pre-trained video or image depth models, which can then be used in the auxiliary target. For our experiments, we obtain depth for OG-Bench Cube from the MuJoCo environment itself. For each object $i$, we summarize its camera depth using selected percentiles $z_i^{(p)}$. Using the mean and standard deviation statistics $\mu_p$ and $\sigma_p$ for all the visible objects in the training set, we obtain statistical depth values:

$$\hat{z}_i^{(p)} = \frac{\log z_i^{(p)} - \mu_p}{\sigma_p}$$

We test three training targets:

- BB + $\log(\text{median depth})$: $b = [x, y, w, h, \hat{z}^{(50)}]$.
- BB + $\log(z_{25}) + \log(z_{75})$: $b = [x, y, w, h, \hat{z}^{(25)}, \hat{z}^{(75)}]$.
- BB + $\log(z_{05}) + \log(z_{95})$: $b = [x, y, w, h, \hat{z}^{(05)}, \hat{z}^{(95)}]$.

The results show that adding the 25th and 75th percentile depths marginally improves performance over the bounding box, but does still not provide sufficient 3D geometric information to be as useful as the object centroids in the Compact13 target in ablation A.4.

| **Target** | **OG-Bench Cube** |
|---|---|
| LeWM | $66.4\% \pm 5.5\%$ |
| BB | $70.4\% \pm 6.1\%$ |
| BB + $\log(\text{median depth})$ | $69.6\% \pm 6.2\%$ |
| BB + $\log(z_{25}) + \log(z_{75})$ | $\mathbf{70.8\%} \pm \mathbf{5.0\%}$ |
| BB + $\log(z_{05}) + \log(z_{95})$ | $69.6\% \pm 5.9\%$ |

Table 8: Depth ablation on OG-Bench Cube. Results report mean $\pm$ standard deviation across five CEM evaluation seeds.

### A.8 NOISY BOUNDING BOX ABLATION

Detected bounding boxes may be noisy in certain cases. We perturb the ground truth bounding boxes by applying a gaussian jitter to each edge $e$ of up to 1% of the image width, therefore being 2.24 pixels for the LeWM environments and 0.64 pixels for the n-ball environments.

$$e' = e + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, 0.01^2)$$

Results indicate that while noising the bounding boxes decreases performance slightly in some cases, it still improves success rate over the baseline method, and still prevents collapse in the 3-Balls environment, demonstrating the validity of this method with imperfect bounding box annotations.

| **Ablation** | **Push-T** | **OG-Bench Cube** | **3-Balls (H=4)** |
|---|---|---|---|
| Bounding Boxes | **95.2% ± 3.3%** | **70.4% ± 6.1%** | **81.2% ± 1.0%** |
| Noisy Bounding Boxes | 93.6% ± 2.6% | 70.4% ± 6.8% | 75.9% ± 1.7% |
| LeWM | 91.2% ± 4.1% | 66.4% ± 5.5% | 2.7% ± 0.8% |

Table 9: Performance comparison between bounding boxes, noisy bounding boxes, and LeWM. Results report mean ± standard deviation across five CEM evaluation seeds.

### A.9 Zero-shot Object Detection Pipeline for Bounding Box Data

To demonstrate that bounding-box supervision does not depend on simulator states or manual annotation, we implement an object discovery, tracking, and bounding-box labeling pipeline using pretrained visual models. Here, *zero-shot* indicates that the pipeline is completely unsupervised, with no environment-specific object labels, semantic class names, object counts, simulator states, or model fine-tuning.

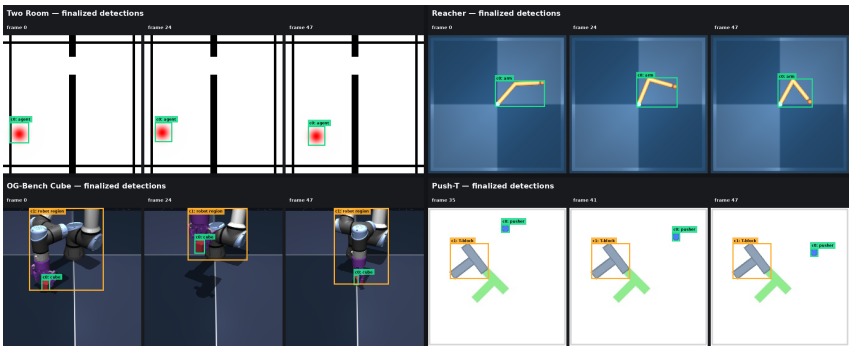


Figure 8: Sample bounding box detections on LeWM environments using the zero-shot object detection pipeline.

For object discovery, we first apply VideoCutLER (Wang et al., 2023) to obtain a collection of object-mask proposals from sparsely sampled video frames. These proposals are then passed through a DINOv2 (Oquab et al., 2024) encoder to obtain embeddings for each mask. Principal Component Analysis (PCA) and the HDBSCAN (Malzer & Baum, 2020) clustering algorithms are used to infer object categories within each environment, producing a local object vocabulary without requiring semantic names or a pre-defined number of objects.

For consistently tracking the objects and labeling the bounding boxes, we construct an appearance prototype by averaging normalized DINOv2 embeddings from each object over multiple frames. These appearance prototypes are passed into a pretrained Segment Anything 3 (SAM3) video tracker (Carion et al., 2026) that propagates the object masks throughout each sequence of frames. Heuristic filters are applied to combine masks that frequently overlap and remove masks deemed relevant by a variety of pre-determined, environment-agnostic factors. If a retained category is missing from a frame, the SAM3 video tracker is re-initialized with the object's DINOv2 embedding to re-acquire its position in subsequent frames.

Finally, the masks are converted into the axis-aligned bounding box representation by finding the bounds of the tracked semantic mask for each frame. These are stored along with the inferred object category and instance identifier for each clip. These masks can also be converted deterministically into oriented bounding boxes and can be used to calculate centers of mass using similar geometric post-processing techniques.

Sample bounding box detections are shown in Figure 8 using this zero-shot object detection pipeline, which closely resemble the simulator ground truth labels shown in Appendix A.1.3. Though the principal experiments use bounding box labels derived from the simulator, this is used as a demonstration of feasibility for a generalized, unsupervised application of this auxiliary head method without requiring manual frame-level annotation or a simulator to provide the bounding box labels.

### A.10 SLOT ATTENTION AUXILIARY HEAD

Slot attention (Locatello et al., 2020) is commonly used in object-centric architectures to organize a latent space to better represent target objects. It factorizes the latent space by learning a predefined number of slots for each input image, assigning objects to them in an unsupervised manner. Slot Attention for Video (SAVi) (Kipf et al., 2021) builds on this by improving slot initialization and persisting them from one frame to the next.

To improve the ability of the auxiliary head to extract object representations from the latent, we implement SaVi-style slot attention into the auxiliary head. The auxiliary head applies recurrent slot attention to the encoder's spatial tokens, followed by a shared 2-layer MLP decoder to output targets. The slots are initialized from ordered ground-truth bounding boxes and propagated between consecutive frames to maintain stable object identities.

| **Ablation** | **Push-T** | **3-Balls (H=4)** |
|---|---|---|
| Ours | **95.2%** | **81.2%** |
| Slot Attention Auxiliary Head | 1.6% | 3.8% |
| LeWM | 91.2% | 1.1% |

Table 10: Performance comparison between Ours, LeWM, and slot attention for the auxiliary head architecture.

The results indicate that the slots did not improve the latent structuring as the 3-ball environment still collapsed. Additionally, Push-T also collapsed due to the slot attention auxiliary head. Figure 9 indicates that the slots were able to learn and persist assignments for the target objects. However, this removed the need for the encoder and predictor to learn well structured latents to minimize the auxiliary objective, leading to collapse for both environments. This points to the need for auxiliary heads to remain simple, as an overparametrized decoder will learn the auxiliary target from poor latents and prevent the model from improving.

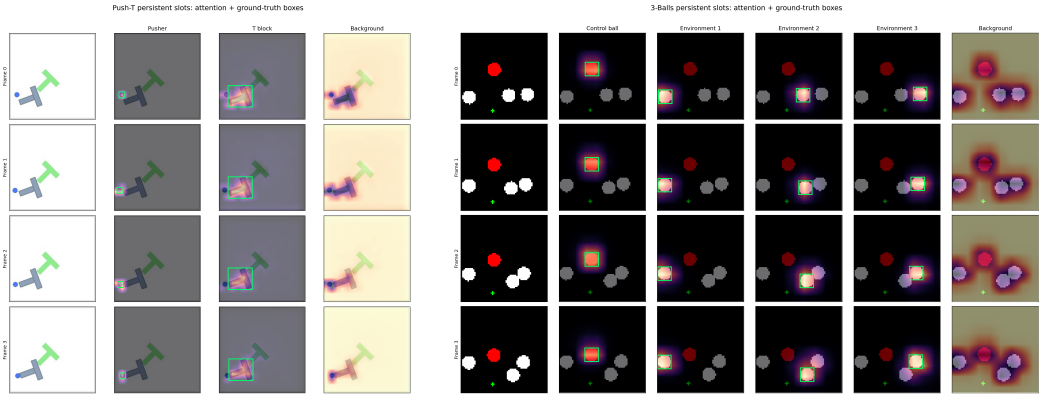


Figure 9: Push-T and 3-Ball slot assignments.

Table 11: Linear and MLP probe results for the final models. Ours uses Fourier-supervised bounding-box prediction with the predictor-side auxiliary head frozen. The shaded rows are the primary metrics. Bold indicates lower MSE or higher Pearson correlation $r$.

| Environment | Target | Linear MSE ↓ | | Linear $r$ ↑ | | MLP MSE ↓ | | MLP $r$ ↑ | |
|---|---|---|---|---|---|---|---|---|---|
| | | Ours | LeWM | Ours | LeWM | Ours | LeWM | Ours | LeWM |
| **Push-T** | **Full state (primary)** | **0.02998** | 0.05672 | **0.98484** | 0.97108 | 0.00576 | **0.00547** | 0.99714 | **0.99729** |
| | Pusher XY | **0.02203** | 0.05629 | **0.98900** | 0.97160 | **0.00981** | 0.01014 | **0.99515** | 0.99500 |
| | T-block XY | **0.01568** | 0.02598 | **0.99224** | 0.98711 | 0.00123 | **0.00108** | 0.99941 | **0.99948** |
| | T-block yaw | **0.05221** | 0.08790 | **0.97327** | 0.95453 | 0.00624 | **0.00519** | 0.99686 | **0.99738** |
| **Two Room** | **Full state (primary)** | **0.00259** | 0.00379 | **0.99870** | 0.99811 | **0.00012** | 0.00018 | **0.99994** | 0.99992 |
| **Reacher** | **Full state (primary)** | **0.22869** | 0.22958 | **0.87226** | 0.87172 | **0.22789** | 0.22789 | 0.87268 | **0.87269** |
| | Joint angles | **0.45409** | 0.45531 | **0.74620** | 0.74541 | **0.45570** | 0.45571 | 0.74540 | **0.74540** |
| | Finger location | **0.00329** | 0.00386 | **0.99833** | 0.99804 | 0.00007 | **0.00006** | 0.99997 | **0.99997** |
| **OG-Bench Cube** | **Full state (primary)** | **0.36245** | 0.36666 | **0.66975** | 0.66651 | 0.39748 | **0.39610** | **0.66696** | 0.66414 |
| | Effector location | **0.00946** | 0.01668 | **0.99521** | 0.99154 | **0.00203** | 0.00255 | **0.99898** | 0.99871 |
| | Effector yaw | **0.99748** | 0.99850 | **0.10474** | 0.09879 | 1.14190 | **1.13966** | **0.05170** | 0.04293 |
| | Gripper opening | **0.10291** | 0.11178 | **0.94705** | 0.94234 | **0.03405** | 0.03922 | **0.98286** | 0.98033 |
| | Gripper contact | **0.18313** | 0.18816 | **0.90385** | 0.90105 | **0.10303** | 0.10856 | **0.94722** | 0.94440 |
| | Cube location | **0.00715** | 0.01143 | **0.99639** | 0.99422 | **0.00146** | 0.00183 | **0.99927** | 0.99908 |
| | Cube yaw | 1.00929 | **1.00929** | **0.00088** | -0.00009 | 1.16924 | **1.15646** | **-0.01238** | -0.01713 |
| $n = 1$ **balls** | **Full positions (primary)** | **0.00859** | 0.49398 | **0.99572** | 0.71278 | **0.00053** | 0.50527 | **0.99974** | 0.70494 |
| | Control ball | **0.00637** | 0.98284 | **0.99684** | 0.15859 | **0.00042** | 1.00990 | **0.99979** | 0.05758 |
| | Environment balls | 0.01080 | **0.00512** | 0.99459 | **0.99744** | 0.00043 | **0.00020** | 0.99979 | **0.99990** |
| $n = 3$ **balls** | **Full positions (primary)** | **0.19330** | 0.36222 | **0.89821** | 0.79862 | **0.10821** | 0.33956 | **0.94443** | 0.81328 |
| | Control ball | **0.00726** | 0.96038 | **0.99643** | 0.23830 | **0.00040** | 1.10246 | **0.99981** | 0.11921 |
| | Environment balls | 0.25531 | **0.16283** | 0.86209 | **0.91443** | 0.14405 | **0.09313** | 0.92483 | **0.95201** |
| $n = 6$ **balls** | **Full positions (primary)** | **0.47547** | 0.95494 | **0.72685** | 0.23115 | **0.48770** | 1.05168 | **0.71948** | 0.14460 |
| | Control ball | **0.00527** | 0.83346 | **0.99740** | 0.42038 | **0.00039** | 0.96106 | **0.99981** | 0.33427 |
| | Environment balls | **0.55384** | 0.97519 | **0.67093** | 0.18149 | **0.56514** | 1.08073 | **0.66404** | 0.09657 |

### A.11 Linear and MLP Probing

Following the LeWM evaluation protocol, and inspired by Joseph et al. (2026) probing, we train additional linear and MLP probes to explicitly quantify how well physical quantities can be extracted from latent embeddings. Better performance on linear probes indicate a better linear representation of the physical quantity within the latent space, while higher performance on an MLP probe indicates a more entangled representation.

**Implementation Details**. Probes are trained using 50000 latents from each environment, and tested on 5000 latents. The linear probe is trained using ridge regression with $\lambda = 10^{-6}$. The MLP probe uses a simple 2-layer MLP, projecting the latent to a 256-D feature before predicting the output state. It is trained for 100 epochs with a batch size of 1024. They are initialized with seeds 3072, 3073, and 3074, and their outputs are averaged for the final results. Probes are trained for each individual state, which are determined by the environments themselves. We report the mean squared error (MSE), and the Pearson correlation coefficient between the ground-truth and predicted states.

**Results**. Table 11 shows improvements in all environments for for most quantities, demonstrating the effectiveness of the auxiliary target in better organizing the latent space with respect to key physical properties. Push-T and the 6-ball environment benefit the most, while 1-ball and 3-ball environments demonstrate a decrease in the prediction performance of the environment balls' positions, which is in line with the reduced spearman correlation in the main results. Two-Room shows poorer linear probe performance, but improved MLP probe performance, indicating the auxiliary target results in an entangled, but improved encoding of positional information in the latent space.