TEMPORALLY CENTERED SIGREG IMPROVES LEWORLDMODEL REPRESENTATIONS FOR ROBOT POLICY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work on LeWorldModel (LeWM) has shown that the Sketched Isotropic Gaussian Regularizer (SIGReg) enables stable end-to-end world model learning from pixels by regularizing the latent representation toward an isotropic Gaussian. While effective for latent-space planning, the representations learned by Raw LeWM are poorly suited for downstream robot policy learning. In this paper, through Monte Carlo analysis, we show that the Raw LeWM objective biases variance allocation toward the temporally persistent component, thereby suppressing the variance of the temporally centered residual. Consistent with this analysis, trained Raw LeWM representations exhibit suppressed residual variation and reduced decodability of robot state and dynamics, particularly gripper dynamics, which are crucial for robotic manipulation. To address this issue, we apply SIGReg to **temporally centered residuals** rather than to the whole latent representation. This simple change decouples persistent and residual variance allocation while retaining an effective anti-collapse property. On the LIBERO benchmark, our method improves downstream policy success on the Goal suite by 1.66× and raises the average success rate across all suites from 63.6% to 83.8%. Without external pretraining, it also outperforms both Diffusion Policy trained from scratch and the pretrained OpenVLA baseline. These results associate the variance-allocation bias of Raw LeWM with the downstream policy gap, and show that decoupling persistent and residual variation yields representations better suited for downstream robot policy learning.

1 INTRODUCTION

World models (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020) learn predictive latent representations of environment dynamics, allowing agents to anticipate the consequences of their actions and enabling planning and policy learning. Many existing approaches (Hafner et al., 2019; 2020; 2025) rely on observation reconstruction, reward prediction, or other task-specific supervision. Although effective, these objectives may retain unnecessary visual details and complicate learning from large reward-free multi-task datasets (Deng et al., 2022; Balestriero & LeCun, 2024; Hutson et al., 2024).

Joint-embedding Predictive Architecture (Assran et al., 2023; 2025) offers an appealing alternative. LeWorldModel (LeWM) (Maes et al., 2026) jointly trains an encoder and an action-conditioned latent predictor directly from pixels. To prevent representation collapse, it uses the Sketched Isotropic Gaussian Regularizer (SIGReg) (Balestriero & LeCun, 2025), which applies an Epps–Pulley (EP) normality criterion (Epps & Pulley, 1983) to random latent projections and encourages the latent distribution to follow an isotropic Gaussian. This enables end-to-end world model learning without reconstruction, pretrained encoders, momentum targets, or reward supervision.

However, preventing representation collapse does not guarantee representations that are well suited for downstream robot policy learning. On the LIBERO-Long suite (Liu et al., 2023), behavior-cloning policies trained on Raw LeWM representations achieve only 26.1% average success under independent single-task training. Increasing the scale of world model training partially improves downstream policy performance, reaching 44.7% under suite-wise 10-task joint training and 52.0% when the world model is trained jointly on all 40 LIBERO tasks, but does not eliminate this perfor-

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.

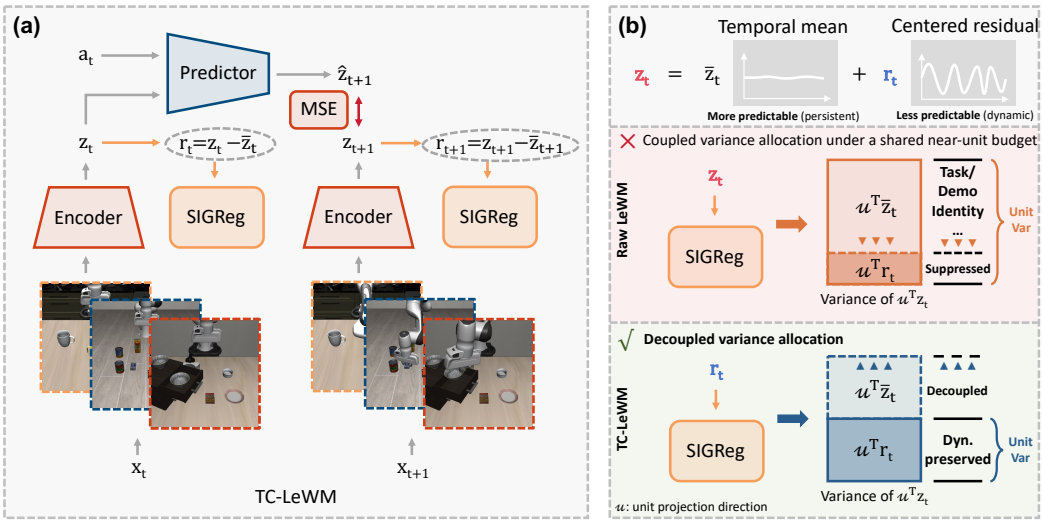

Figure 1: **Overview of TC-LeWM. (a)** TC-LeWM decomposes each latent representation as $\mathbf{z}_t = \bar{\mathbf{z}}_t + \mathbf{r}_t$, where $\bar{\mathbf{z}}_t$ is the temporally persistent component and $\mathbf{r}_t$ is the temporally centered residual. SIGReg is applied to $\mathbf{r}_t$ rather than directly to $\mathbf{z}_t$. **(b)** Along each regularized projection direction, the Raw LeWM objective couples the variance allocation of the persistent component and centered residual under a shared near-unit variance budget, whereas TC-LeWM decouples their variance allocation by regularizing the centered residual directly.

mance limitation. These observations motivate us to examine how the Raw LeWM training objective shapes the information retained in its learned representations.

We trace this representation limitation to a variance-allocation bias in the objective of Raw LeWM. We first decompose the latent representation into a temporally persistent component and a temporally centered residual. Based on this decomposition, we develop a controlled Monte Carlo analysis of the Raw LeWM objective along individual SIGReg projection directions. The analysis reveals a mechanism. Along each projection direction, SIGReg encourages the projected latent variance toward the unit-variance target, while the prediction objective favors allocating variance to more predictable temporally persistent variation. Their interaction therefore couples the variance allocation of the persistent component and centered residual under a shared near-unit variance budget, biasing the allocation toward the persistent component and suppressing residual variance.

To address this issue, we introduce **Temporally Centered LeWorldModel** (TC-LeWM), illustrated in Fig. 1. TC-LeWM applies SIGReg to the temporally centered residual rather than directly to the original latent representation. This simple change decouples persistent and residual variance allocation while retaining an effective anti-collapse property. Across the four LIBERO suites, TC-LeWM raises the suite-wise 10-task average success rate of the downstream policy from 63.6% to 83.8%. When the world model is jointly trained on all 40 LIBERO tasks, TC-LeWM improves the average downstream policy success rate from 72.6% to 85.4%.

Consistent with our Monte Carlo analysis, Raw LeWM representations exhibit suppressed centered-residual variation, whereas TC-LeWM preserves substantially more variation in the residual component. Content attribution further shows that, under multi-task training, Raw LeWM allocates more variance to task and trajectory identity, whereas TC-LeWM allocates relatively more variation to within-episode progress. Trajectory identity denotes the specific demonstration trajectory from which a frame is sampled and therefore captures trajectory-specific variation. The suppressed residual variation in Raw LeWM is also accompanied by lower decodability of robot state and dynamics, particularly gripper dynamics, which are crucial for closed-loop manipulation.

Contributions. **(i)** We introduce a temporal decomposition and controlled Monte Carlo analysis of the Raw LeWM objective, revealing its bias toward persistent variation and the resulting suppression of residual variance. **(ii)** We propose TC-LeWM, which applies SIGReg to the temporally centered residual to decouple persistent and residual variance allocation while retaining an effective

anti-collapse property. **(iii)** Experimentally, TC-LeWM consistently improves the closed-loop performance of downstream behavior-cloning policies across three different training scales on LIBERO. We further analyze the learned representations through variance allocation, content attribution, and the decodability of robot state and dynamics.

2 Related Work

Latent world models. World models learn compact predictive representations of environment dynamics for planning and policy learning (Ha & Schmidhuber, 2018; Hafner et al., 2020; Schrittwieser et al., 2020; Hafner et al., 2021; 2025). Early latent-dynamics approaches showed that environment dynamics could be learned and used for control in compact latent spaces (Watter et al., 2015; Hafner et al., 2019). Subsequent stochastic state-space models combined deterministic memory with latent uncertainty, enabling control directly from pixel observations (Hafner et al., 2020; 2019). Dreamer-style methods (Hafner et al., 2020; Wu et al., 2023; Hafner et al., 2025) further learned policies from trajectories imagined within the world model and extended this paradigm across continuous control, Atari, diverse domains, and physical robots.

Most of these methods rely on observation reconstruction, reward prediction, or other task-specific objectives. Although effective, reconstruction objectives may retain control-irrelevant visual information, while reward-dependent objectives require supervision unavailable in reward-free datasets (Fu et al., 2021; Wan et al., 2023; Zhu et al., 2023; Balestriero & LeCun, 2024; Hutson et al., 2024; Wang et al., 2024; Sobal et al., 2025).

Joint-embedding predictive learning. Joint-Embedding Predictive Architectures (JEPAs) learn representations by predicting target embeddings directly in latent space rather than reconstructing raw observations (Assran et al., 2023; Dawid & LeCun, 2024). I-JEPA (Assran et al., 2023) instantiated this paradigm for images, and V-JEPA (Bardes et al., 2024; Assran et al., 2025) extended it to spatiotemporal representation learning from video. Related approaches have adapted latent prediction to action-conditioned world modeling: DINO-WM (Zhou et al., 2025) learns dynamics over frozen pretrained visual features for zero-shot planning, whereas PLDM (Sobal et al., 2025) jointly learns reconstruction-free latent dynamics from reward-free trajectories. However, non-contrastive joint-embedding objectives admit collapsed solutions, motivating mechanisms such as momentum target encoders, stop-gradient asymmetry, explicit variance regularization, and redundancy-reduction objectives (Grill et al., 2020; Chen & He, 2021; Bardes et al., 2022; Zbontar et al., 2021).

LeWorldModel and anti-collapse regularization. LeWM (Maes et al., 2026) enables end-to-end reconstruction-free world model learning directly from pixels. It jointly trains an encoder and an action-conditioned predictor using latent next-state prediction together with SIGReg (Balestriero & LeCun, 2025), which prevents representation collapse without pretrained encoders, momentum targets, observation reconstruction, or reward supervision.

SIGReg is applied directly to latent representations and encourages their distribution toward an isotropic Gaussian through random one-dimensional projections. While effective for anti-collapse, how the Raw LeWM objective allocates variance across temporal components has received little attention. We show that the interaction between latent prediction and SIGReg couples the variance allocation of the temporally persistent component and centered residual, biasing the allocation toward persistent variation and suppressing residual variance. TC-LeWM addresses this issue by applying SIGReg to the temporally centered residual, thereby decoupling persistent and residual variance allocation while retaining an effective anti-collapse property.

3 Why the Raw LeWM Objective Can Suppress Residual Variation

In this section, we first characterize the performance limitation of Raw LeWM across different training scales. We then introduce a temporal decomposition of the latent representation and, based on this decomposition, construct a projected mixture model for analyzing the Raw LeWM objective. Through a Monte Carlo analysis, we show how the interaction between latent prediction and SIGReg biases variance allocation toward temporally persistent variation and suppresses residual variance.

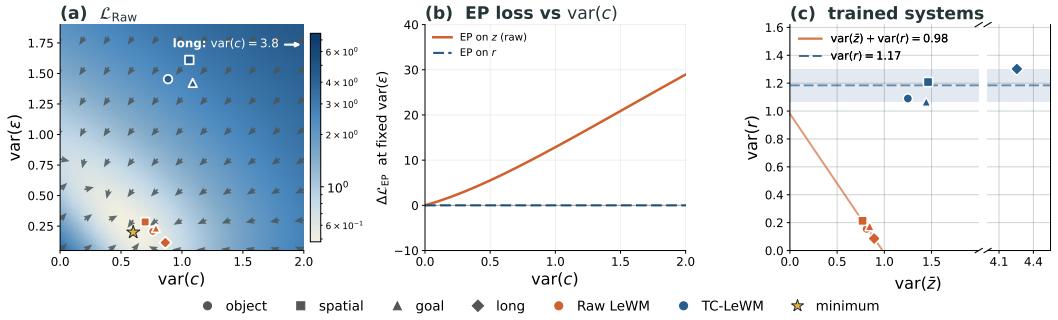

Figure 2: **Monte Carlo and empirical analysis of variance allocation in LeWM. (a)** The Raw LeWM objective $\mathcal{L}_{\text{Raw}}$ over the $(\text{var}(c), \text{var}(\varepsilon))$ plane, revealing a minimum biased toward persistent variation. **(b)** Change in EP loss relative to $\text{var}(c) = 0$ at fixed $\text{var}(\varepsilon)$. Increasing persistent variance raises the EP loss on $x = c + \varepsilon$. **(c)** Empirical variance allocation of trained LIBERO representations. Raw LeWM exhibits a nearly fixed total variance dominated by the persistent component, whereas TC-LeWM maintains a stable residual variance with persistent variance free to vary.

3.1 Downstream Policy Performance Across Training Scales

World model training and downstream evaluation. Consider a task set $\mathcal{T} = \{1, \ldots, N\}$, where task i is associated with a demonstration dataset $\mathcal{D}_i$. Raw LeWM jointly trains an encoder and an action-conditioned latent predictor on the aggregated dataset $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}_i$ using

$$\mathcal{L}_{\text{Raw}} = \mathcal{L}_{\text{pred}} + \lambda\, \mathcal{L}_{\text{SIGReg}}(\mathcal{Z}), \qquad \mathcal{Z} = \{\mathbf{z}_t \mid \mathbf{z}_t = f_\theta(\mathbf{x}_t)\}. \tag{1}$$

After world model training, the encoder is frozen and used for downstream robot policy learning with a task-conditioned behavior-cloning policy based on flow matching (Lipman et al., 2023).

Limited benefit from training scale. We evaluate Raw LeWM at three training scales of LIBERO-Long. When each of the ten tasks is trained independently, Raw LeWM achieves only 26.1% average success. Joint training over the ten tasks raises the success rate to 44.7%, while training the world model jointly on all 40 LIBERO tasks further raises it to 52.0%. Thus, increasing the amount and diversity of training data substantially improves Raw LeWM, but does not eliminate its performance limitation. These results show that avoiding representation collapse alone does not guarantee representations suitable for downstream manipulation policy learning.

3.2 Temporal Decomposition and Monte Carlo Analysis

Latent decomposition and mixture model. To analyze the problem, we decompose each latent representation into a local temporal mean and a complementary centered residual:

$$\bar{\mathbf{z}}_t \triangleq \frac{1}{|\mathcal{W}_t|} \sum_{s \in \mathcal{W}_t} \mathbf{z}_s, \qquad \mathbf{r}_t \triangleq \mathbf{z}_t - \bar{\mathbf{z}}_t, \qquad \mathbf{z}_t \equiv \bar{\mathbf{z}}_t + \mathbf{r}_t, \tag{2}$$

where $\mathcal{W}_t$ denotes a local temporal window. We refer to $\bar{\mathbf{z}}_t$ as the temporally persistent component and $\mathbf{r}_t$ as the temporally centered residual. The persistent component captures variation that is shared within the local temporal window, while the centered residual captures complementary within-window variation. SIGReg applies the Epps–Pulley (EP) criterion to random one-dimensional marginal projections of the latent representation. For a projection vector $\mathbf{w}$, linearity preserves the temporal decomposition:

$$x = \mathbf{w}^\top \mathbf{z}_t = \underbrace{\mathbf{w}^\top \bar{\mathbf{z}}_t}_{c} + \underbrace{\mathbf{w}^\top \mathbf{r}_t}_{\epsilon}. \tag{3}$$

We therefore study the Raw LeWM objective through the projected variables c and ϵ. For a controlled analysis, we model c as a balanced K-component Gaussian mixture and $\epsilon \sim \mathcal{N}(0, \sigma_r^2)$ as Gaussian residual variation, with $\text{Cov}(c, \epsilon) = 0$. This yields

$$x = c + \varepsilon, \qquad c \sim \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}\left(\mu_k,\, \tau^2\right), \qquad \varepsilon \sim \mathcal{N}\left(0,\, \sigma_r^2\right), \qquad \text{Cov}(c, \varepsilon) = 0, \tag{4}$$

where μ_k is the projected center of the k-th component, τ is the common within-component standard deviation of the persistent component, and σ_r is the scale of the residual variation.

Monte Carlo analysis of the Raw LeWM objective. For each Monte Carlo layout, we sample and center K component centers, normalize their variance to one, and rescale them by $\sqrt{\mathrm{var}(c)}$. The residual component is constructed analogously for a target $\mathrm{var}(\varepsilon)$. We set $\tau = 0$, so that $\mathrm{var}(c) = \frac{1}{K}\sum_k \mu_k^2$ and $\mathrm{var}(\varepsilon) = \sigma_r^2$ are directly controlled and swept over a grid. We use $K = 10$ and temporal windows of $W = 4$, matching the suite-wise training setup. For each projected direction, we approximate the prediction cost as $\mathcal{L}_{\mathrm{pred}} = \mathrm{var}(\varepsilon)$ and evaluate $\mathcal{L}_{\mathrm{Raw}} = \mathcal{L}_{\mathrm{pred}} + 0.09\,\mathcal{L}_{\mathrm{EP}}(x)$. Full details are provided in Appendix A.2.

Results. Figure 2(a) shows $\mathcal{L}_{\mathrm{Raw}}$ over the $(\mathrm{var}(c), \mathrm{var}(\varepsilon))$ plane, with a clear minimum around $\mathrm{var}(c) \approx 0.6$ and $\mathrm{var}(\varepsilon) \approx 0.2$. Along each projection direction, the EP criterion encourages the projected marginal toward a standard Gaussian and therefore favors variance near one. Under the zero-cross-covariance setting used in our controlled analysis, $\mathrm{var}(x) = \mathrm{var}(c) + \mathrm{var}(\varepsilon)$, so the persistent component and centered residual compete for the variance required by the projected marginal. Meanwhile, the prediction objective favors smaller residual variation. Their interaction therefore biases variance allocation toward the persistent component at the expense of the centered residual, suppressing residual variance.

Implication for policy learning. The centered residual captures complementary within-window variation that can encode fine-grained changes in robot and environment states, including gripper motion, that are crucial for closed-loop manipulation. Suppressing this variation may therefore reduce the accessibility of robot state and dynamics from the learned representation and degrade downstream control, even without representation collapse. Our experiments in Sec. 5 show that Raw LeWM representations have lower decodability of robot state and dynamics, particularly gripper dynamics, accompanied by poorer closed-loop manipulation performance.

4 METHOD

We introduce Temporally Centered LeWorldModel (TC-LeWM), which applies SIGReg to temporally centered residuals rather than directly to the original latent representations. This modification decouples persistent and residual variance allocation while retaining an effective anti-collapse property.

4.1 TEMPORALLY CENTERED LEWORLDMODEL

Model and training objective. As illustrated in Fig. 1, TC-LeWM retains the LeWM encoder and action-conditioned latent predictor, $\mathbf{z}_t = f_\theta(\mathbf{x}_t)$ and $\hat{\mathbf{z}}_{t+1} = g_\psi(\mathbf{z}_t, \mathbf{a}_t)$. For each latent sequence, we compute $\bar{\mathbf{z}}_t = \frac{1}{|\mathcal{W}_t|}\sum_{s\in\mathcal{W}_t} \mathbf{z}_s, \qquad \mathbf{r}_t = \mathbf{z}_t - \bar{\mathbf{z}}_t$, where $\mathcal{W}_t$ is a local temporal window. Let $\mathcal{R} = \{\mathbf{r}_t\}$ denote the residuals in a training batch. TC-LeWM optimizes

$$\mathcal{L}_{\mathrm{TC}} = \mathcal{L}_{\mathrm{pred}} + \lambda\,\mathcal{L}_{\mathrm{SIGReg}}(\mathcal{R}). \tag{5}$$

Thus, the only change from Raw LeWM is the regularization target: SIGReg is applied to $\mathcal{R}$ instead of the original latent representations $\mathcal{Z}$. Further architectural and behavior-cloning implementation details are provided in Appendix A.1.

4.2 WHY THE RESIDUAL TARGET WORKS

Applying SIGReg to the temporally centered residual has two key properties: it decouples persistent and residual variance allocation and retains an effective anti-collapse property.

Decoupling persistent and residual variance allocation. In Raw LeWM, the EP criterion is evaluated on the projected latent $x = c + \varepsilon$. Under the zero-cross-covariance setting in Sec. 3.2, $\mathrm{var}(x) = \mathrm{var}(c) + \mathrm{var}(\varepsilon)$, so the persistent component and centered residual share the near-unit projected variance budget induced by SIGReg. TC-LeWM instead evaluates the EP criterion on the projected residual $\varepsilon = \mathbf{w}^\top \mathbf{r}_t$, which does not directly contain the persistent component c. Figure 2(b) illustrates this difference: as $\mathrm{var}(c)$ increases, $\Delta\mathcal{L}_{\mathrm{EP}}$ increases for $x = c + \varepsilon$ but remains identically zero for ε. Thus, the residual remains regularized toward its standard-Gaussian target,

Table 1: Per-task closed-loop success rates (%) under **independent single-task training** on the ten LIBERO-Long tasks. Results are averaged over three training seeds and three evaluation seeds; standard deviations are omitted for space.

Method	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	Avg.
Raw LeWM	53.1	37.5	12.4	11.8	11.6	13.6	29.8	28.7	17.8	44.7	26.1
TC-LeWM (Ours)	**85.1**	**68.5**	**75.6**	**25.6**	**36.2**	**37.8**	**42.9**	**39.1**	**34.9**	**55.3**	**50.1**
Δ (TC $-$ Raw)	+32.0	+31.0	+63.2	+13.8	+24.6	+24.2	+13.1	+10.4	+17.1	+10.6	$+\mathbf{24.0}$

Table 2: Closed-loop success rates (%) under **suite-wise 10-task training**. Results for Diffusion Policy, Octo, and OpenVLA are taken from Kim et al. (2025), while the $\pi_{0.5}$ results are taken from Black et al. (2025). Raw LeWM and TC-LeWM are reported as mean$\pm$std over three full-pipeline training seeds.

Method	Spatial	Object	Goal	Long	Avg.
Pre-trained methods					
Octo (fine-tuned)	78.9	85.7	84.6	51.1	75.1
OpenVLA (fine-tuned)	84.7	88.4	79.2	53.7	76.5
$\pi_{0.5}$ (fine-tuned)	98.8	98.2	98.0	92.4	96.9
From-scratch methods					
Diffusion Policy	78.3	92.5	68.3	50.5	72.4
Raw LeWM	62.4$\pm$7.5	93.9$\pm$2.3	53.4$\pm$5.3	44.7$\pm$4.3	63.6
TC-LeWM (Ours)	**86.0**$\pm$4.1	**97.0**$\pm$0.1	**88.4**$\pm$7.5	**63.6**$\pm$6.8	**83.8**

while persistent variation no longer trades off against residual variance to satisfy the same projected target, decoupling their variance allocation.

Anti-collapse property retained. If the representation collapses within a temporal window, $\mathbf{z}_t = \bar{\mathbf{z}}_t$ and therefore $\mathbf{r}_t = \mathbf{0}$. All one-dimensional projections of $\mathcal{R}$ then degenerate at zero and are strongly penalized by the EP criterion relative to the standard-Gaussian target. TC-LeWM therefore retains an effective anti-collapse property. The prediction and residual-regularization objectives remain complementary: unstructured residual variation may help satisfy SIGReg but is difficult to predict and is penalized by $\mathcal{L}_{\text{pred}}$. Their combination therefore favors non-degenerate, predictable temporal variation without the persistent–residual coupling of Raw LeWM.

5 Experiments

We evaluate TC-LeWM on the LIBERO benchmark through closed-loop manipulation and analyze how temporal centering changes the learned representations.

5.1 Experimental Setup

Benchmark. LIBERO (Liu et al., 2023) contains four suites—Spatial, Object, Goal, and Long—each consisting of ten manipulation tasks. Each observation contains two RGB camera views. A shared ViT encoder (Dosovitskiy et al., 2021) processes the two views independently. We reconstruct the LIBERO datasets following OpenVLA (Kim et al., 2025).

World model training and downstream policy learning. We study downstream robot policy learning using a two-stage pipeline. First, the encoder and action-conditioned latent predictor are trained with the world model objective on reward-free demonstration trajectories. At each frame, the CLS tokens from the two camera views are concatenated and passed through a shared projector to obtain the latent representation regularized by SIGReg, while patch tokens are not directly regularized. After world model training, the encoder is frozen and a task-conditioned flow-matching behavior-cloning policy is trained on the frozen visual representations. For downstream policy learning, we use both the CLS and patch tokens from the two camera views.

Table 3: Closed-loop success rates (%) under **unified 40-task world model training**. All four LIBERO suites share one encoder and latent predictor, while a separate task-conditioned downstream policy is trained for each suite. Results are mean±standard deviation over three full-pipeline training seeds, each evaluated using three evaluation seeds with 50 rollouts per task.

Method	Spatial	Object	Goal	Long	Avg.
Raw LeWM	62.1±11.1	95.1±1.0	81.0±4.7	52.0±4.3	72.6
TC-LeWM (Ours)	**92.5**±3.0	**97.1**±1.3	**89.7**±2.5	**62.3**±4.3	**85.4**
Δ (TC − Raw)	+30.4	+2.0	+8.7	+10.3	**+12.8**

This setting allows us to examine how representations learned by the world model affect downstream robot policy performance. Raw LeWM and TC-LeWM use identical frozen-encoder and downstream-policy settings.

Task aggregation settings. We consider three levels of task aggregation. In the **single-task** setting, one world model and one downstream policy are trained independently for each LIBERO-Long task. In the **10-task** setting, the ten tasks within each suite share one encoder, latent predictor, and task-conditioned downstream policy. In the **40-task** setting, data from all four suites are aggregated to train one unified world model. After freezing the encoder, we train a separate task-conditioned downstream policy for each suite, such that each downstream policy still covers ten tasks as in the 10-task setting. This design isolates the effect of scaling the representation learner from jointly training a single 40-task action policy.

Compared methods. We mainly compare **Raw LeWM**, which applies SIGReg to the latent representations $\mathbf{z}_t$, with **TC-LeWM**, which applies the same regularizer to the temporally centered residual $\mathbf{r}_t = \mathbf{z}_t - \bar{\mathbf{z}}_t$. Raw LeWM and TC-LeWM share the same architecture, training data, and downstream policy protocol. TC-LeWM uses a temporal window of $W = 4$ frames.

Closed-loop evaluation. Each model is evaluated using three evaluation seeds with 50 rollouts per task. We report closed-loop task success rates as mean±standard deviation over three full-pipeline training seeds.

5.2 Closed-Loop Manipulation Performance

Single-task training. Table 1 shows that when each LIBERO-Long task is trained independently, the downstream policy using Raw LeWM representations achieves only 26.1% average success, whereas the policy using TC-LeWM representations reaches 50.1%. TC-LeWM improves all ten tasks, yielding an average gain of $+24.0$ percentage points. These results show that avoiding representation collapse alone does not guarantee representations suitable for downstream manipulation policy learning.

Suite-wise 10-task training. We next jointly train the ten tasks within each LIBERO suite, following the standard suite-wise evaluation setting of LIBERO. As shown in Table 2, TC-LeWM consistently outperforms Raw LeWM across all four suites, raising the average success rate from 63.6% to 83.8%, with particularly large gains on LIBERO-Goal and LIBERO-Spatial. Among methods trained from scratch, TC-LeWM achieves the highest average success rate, outperforming Diffusion Policy (Chi et al., 2023). It also surpasses the pretrained Octo (Ghosh et al., 2024) and OpenVLA baselines, while large-scale pretrained $\pi_{0.5}$ (Black et al., 2025) remains stronger.

Unified 40-task world model training. We further aggregate all 40 LIBERO tasks to train a single world model. Table 3 shows that TC-LeWM again outperforms Raw LeWM on all four suites, achieving 85.4% average success compared with 72.6% for Raw LeWM, an absolute improvement of 12.8 percentage points. The largest gain is observed on LIBERO-Spatial, where TC-LeWM improves success from 62.1% to 92.5%, while substantial gains also remain on LIBERO-Goal and LIBERO-Long. Raw LeWM benefits from the larger training set, most notably on LIBERO-Goal, but the absolute performance advantage of TC-LeWM persists even under unified 40-task world model training. This indicates that scaling the world-model training data alone does not remove the representation gap addressed by temporal centering.

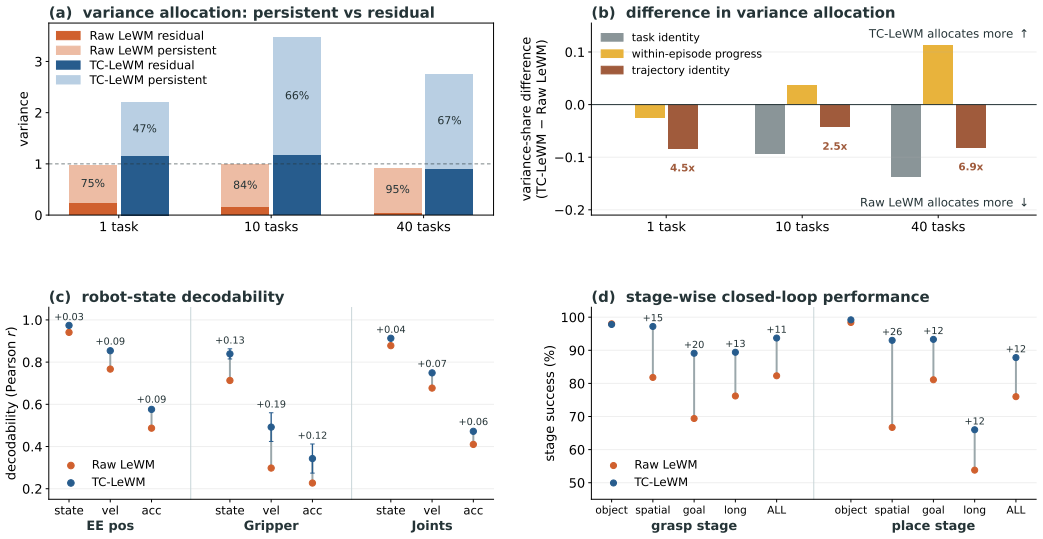

Figure 3: **Analysis of learned representations**, averaged over three training seeds. **(a)** Variance allocation between the temporally persistent component and centered residual across training scales. **(b)** Variance associated with task identity, within-episode progress, and trajectory identity. **(c)** Linear decodability of end-effector, gripper, and joint state and dynamics. **(d)** Closed-loop success at the grasping and placement stages.

5.3 Analysis of Learned Representations

We next analyze how differences in the learned representations are reflected in encoded information, robot-state decodability, and closed-loop behavior.

Variance allocation. Figure 2(c) directly measures $\mathrm{var}(\bar{\mathbf{z}})$ and $\mathrm{var}(\mathbf{r})$ in trained 10-task representations. Raw LeWM exhibits a nearly fixed total variance dominated by the persistent component, consistent with the coupled allocation predicted by the Monte Carlo analysis. In contrast, TC-LeWM maintains a stable residual variance while allowing persistent variance to vary across suites, consistent with decoupled variance allocation.

Figure 3(a) further shows that this distinction persists across all three training scales. In Raw LeWM, the temporally persistent component dominates the representation, while the centered residual accounts for less than approximately 25% of the total variance. TC-LeWM instead maintains substantial variation in both components: the persistent component accounts for approximately 47–67% of the total variance, while the centered residual retains a substantial share at every scale. Thus, neither component consistently dominates the learned representation under TC-LeWM.

Content attribution. Figure 3(b) analyzes how representation variance is associated with task identity, within-episode progress, and trajectory identity. At the multi-task scales, Raw LeWM allocates more variance to task identity than TC-LeWM, and across all training scales it allocates 2.5–$6.9\times$ more variance to trajectory identity. Task identity is partly redundant because the downstream policy already receives the task label explicitly, while trajectory identity largely captures trajectory-specific variation that is less relevant to fine-grained manipulation.

In contrast, TC-LeWM allocates relatively more variance to within-episode progress under multi-task training. Such variation reflects the temporal evolution of the manipulation trajectory and is more directly aligned with the changing state required for closed-loop control. These results suggest that TC-LeWM not only preserves more residual variation, but also shifts the representation away from identity-specific variation toward temporally structured information.

Decodability of robot state and dynamics. We next test whether the preserved temporal variation makes robot state and dynamics more accessible from the learned representation. Figure 3(c) reports linear decodability from the 10-task representations, averaged over three training seeds, for

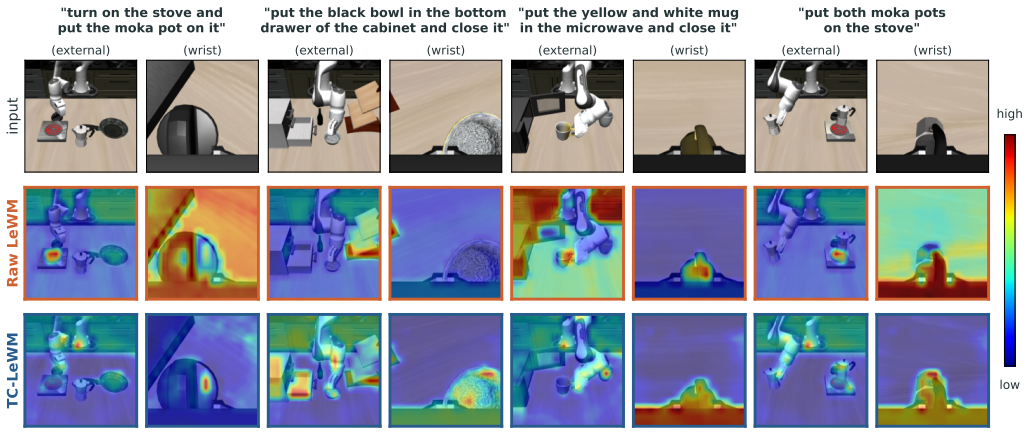

Figure 4: **Attention-rollout visualization at the grasping moment** for the first four LIBERO-Long tasks. TC-LeWM more consistently attends to the robot arm, gripper, and manipulated object, whereas Raw LeWM more frequently attends to background or task-irrelevant regions.

end-effector, gripper, and joint quantities at the state, velocity, and acceleration levels. TC-LeWM achieves higher decodability than Raw LeWM for all three quantities and at all derivative orders. The advantage is particularly pronounced for gripper dynamics, especially at the velocity level. Reduced centered-residual variation in Raw LeWM is accompanied by lower accessibility of fine-grained robot state and dynamics, whereas TC-LeWM preserves these quantities more effectively.

Behavioral analysis. Given the improved decodability of robot state and dynamics, we further examine where the downstream performance gap emerges during closed-loop execution. Figure 3(d) decomposes manipulation into grasping and placement stages. TC-LeWM achieves substantially higher success than Raw LeWM at both stages.

Both grasping and placement require fine-grained control of the robot arm and depend strongly on the evolving robot state and dynamics. The consistent performance advantage across both stages is therefore aligned with the improved decodability of robot state and dynamics from TC-LeWM representations observed in Fig. 3(c).

Attention visualization. Finally, we qualitatively examine the visual features used by the two encoders through attention rollout (Abnar & Zuidema, 2020) at the grasping moment. As shown in Fig. 4, TC-LeWM more consistently attends to manipulation-relevant regions, including the robot arm, gripper, and manipulated object, whereas Raw LeWM more often attends to background or task-irrelevant regions. The same tendency appears in the wrist-camera view, where TC-LeWM more consistently attends jointly to the gripper and grasped object. These observations provide complementary evidence that TC-LeWM learns visual representations more closely aligned with robot-object interaction. Implementation details and additional latent-space analyses are provided in Appendices A.3 and A.4.

6 Conclusion

In this work, we identified a variance-allocation bias in the Raw LeWM objective: the interaction between latent prediction and SIGReg favors temporally persistent variation while suppressing the centered residual. Based on this analysis, we introduced TC-LeWM, which applies SIGReg to temporally centered residuals to decouple their variance allocation while retaining an effective anti-collapse property. Across single-task, suite-wise 10-task, and unified 40-task LIBERO settings, TC-LeWM consistently improves the closed-loop performance of downstream behavior-cloning policies and yields representations with more linearly decodable robot state and dynamics, particularly gripper dynamics. Future work will scale TC-LeWM to larger reward-free datasets and explore its integration with Vision-Language-Action and World-Action models for open-world manipulation.

AI USE STATEMENT

We used generative AI tools to assist with literature collection for relevant prior work, language editing and polishing of the manuscript, and limited code implementation. All AI-assisted outputs were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all claims, results, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed model architectures, training objectives, behavior-cloning policy configurations, and training hyperparameters in Appendix A.1. Details of the controlled Monte Carlo analysis, including its assumptions and simulation setup, are provided in Appendix A.2. The experimental sections specify the training and evaluation protocols, including the number of training and evaluation seeds and the number of rollouts used for closed-loop evaluation. Raw LeWM and TC-LeWM use the same architecture, training data, and downstream policy protocol, differing only in the target to which SIGReg is applied.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4190–4197, 2020.

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

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Randall Balestriero and Yann LeCun. How learning by reconstruction produces uninformative features for perception. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 2566–2585. PMLR, 2024.

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

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Featured Certification.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin C. M. Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

Anna Dawid and Yann LeCun. Introduction to latent variable energy-based models: A path toward autonomous machine intelligence. *Journal of Statistical Mechanics: Theory and Experiment*, 2024(10):104011, 2024. doi: 10.1088/1742-5468/ad292b.

Fei Deng, Ingook Jang, and Sungjin Ahn. DreamerPro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 4956–4975. PMLR, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

T. W. Epps and Lawrence B. Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983. doi: 10.1093/biomet/70.3.723.

Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3480–3491. PMLR, 2021.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284, 2020.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2455–2467, 2018.

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.

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

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Miles Hutson, Isaac Kauvar, and Nick Haber. Policy-shaped prediction: Avoiding distractions in model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 13124–13148, 2024. doi: 10.52202/079017-0418.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4.

Uladzislau 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. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Shenghua Wan, Yucen Wang, Minghao Shao, Ruying Chen, and De-Chuan Zhan. SeMAIL: Eliminating distractors in visual imitation via separated models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35426–35443. PMLR, 2023.

Yucen Wang, Shenghua Wan, Le Gan, Shuai Feng, and De-Chuan Zhan. AD3: Implicit action is the key for world models to distinguish the diverse visual distractors. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 51546–51568. PMLR, 2024.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2746–2754, 2015.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2226–2240. PMLR, 2023.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 2021.

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 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025.

Chuning Zhu, Max Simchowitz, Siri Gadipudi, and Abhishek Gupta. RePo: Resilient model-based reinforcement learning by regularizing posterior predictability. In *Advances in Neural Information Processing Systems*, volume 36, pp. 32445–32467, 2023.

A Appendix

A.1 Model and Implementation Details

Multi-view extension. We retain the encoder and latent predictor architecture of LeWM and extend its visual input to the two RGB views used in LIBERO: an external camera and a wrist-mounted camera. A parameter-shared ViT encoder processes the two views independently. Let

$$\mathbf{h}_t^{(k)} = \mathrm{CLS}\left(E_\theta(\mathbf{x}_t^{(k)})\right), \qquad k \in \{1, \dots, K_{\mathrm{cam}}\}, \tag{6}$$

denote the CLS token of camera view k. The per-view CLS tokens are concatenated and passed through the LeWM projector to form a single latent representation:

$$\mathbf{z}_t = p_\theta\left([\mathbf{h}_t^{(1)}; \dots; \mathbf{h}_t^{(K_{\mathrm{cam}})}]\right). \tag{7}$$

For LIBERO, $K_{\mathrm{cam}} = 2$. Patch tokens are not directly regularized by SIGReg but are retained for downstream policy learning. Apart from this multi-view extension, we keep the LeWM encoder and predictor architectures unchanged.

Prediction objective. The action-conditioned latent predictor operates on the fused representation,

$$\hat{\mathbf{z}}_{t+1} = g_\psi(\mathbf{z}_{t-h+1:t}, \mathbf{a}_{t-h+1:t}), \tag{8}$$

and is trained with the mean-squared next-latent prediction loss

$$\mathcal{L}_{\mathrm{pred}} = \mathrm{MSE}\left(\hat{\mathbf{z}}_{t+1}, \mathbf{z}_{t+1}\right). \tag{9}$$

SIGReg and training objectives. We use the original SIGReg implementation of LeWM without modification. SIGReg evaluates the Epps–Pulley normality criterion on random one-dimensional projections of the latent representations, encouraging each projected marginal toward a standard Gaussian. We use $J = 1024$ random projection directions, following LeWM, and evaluate the regularizer independently at each temporal position across the training batch.

Raw LeWM applies SIGReg directly to the latent representations:

$$\mathcal{L}_{\mathrm{Raw}} = \mathcal{L}_{\mathrm{pred}} + \lambda\, \mathcal{L}_{\mathrm{SIGReg}}(\mathcal{Z}). \tag{10}$$

For TC-LeWM, temporal centering is applied to the same latent sequence,

$$\bar{\mathbf{z}}_t = \frac{1}{|\mathcal{W}_t|} \sum_{s \in \mathcal{W}_t} \mathbf{z}_s, \qquad \mathbf{r}_t = \mathbf{z}_t - \bar{\mathbf{z}}_t, \tag{11}$$

and the objective becomes

$$\mathcal{L}_{\mathrm{TC}} = \mathcal{L}_{\mathrm{pred}} + \lambda\, \mathcal{L}_{\mathrm{SIGReg}}(\mathcal{R}). \tag{12}$$

Thus, Raw LeWM and TC-LeWM use the same architecture, prediction objective, and multi-view treatment, differing only in the SIGReg target: $\mathcal{Z}$ versus $\mathcal{R}$.

Behavior-cloning head. After world model training, the encoder is frozen and used with a task-conditioned flow-matching behavior-cloning policy. For each camera view, we retain the CLS token together with a 4×4 spatially pooled grid of patch tokens from the frozen ViT encoder. The tokens from both views are concatenated into a visual token sequence $\mathbf{V}_t$ and provided to a DiT-style action head (Peebles & Xie, 2023).

The policy predicts an action chunk

$$\mathbf{A}_t = [\mathbf{a}_t, \dots, \mathbf{a}_{t+H-1}] \in \mathbb{R}^{H \times 7}. \tag{13}$$

Action tokens are processed by self-attention and cross-attend to $\mathbf{V}_t$, while the flow time and task identity condition the transformer blocks. We train the policy with the flow-matching objective

$$\mathbf{X}_\gamma = (1-\gamma)\boldsymbol{\xi} + \gamma \mathbf{A}_t, \qquad \mathcal{L}_{\mathrm{BC}} = \|v_\phi(\mathbf{X}_\gamma, \gamma, \mathbf{V}_t, i) - (\mathbf{A}_t - \boldsymbol{\xi})\|_2^2, \tag{14}$$

where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\gamma \sim \mathcal{U}(0,1)$, and i denotes the task identity. Actions are normalized using the training-set mean and standard deviation. During evaluation, action chunks are generated from Gaussian noise using 10 Euler integration steps and mapped back to the original action scale. Raw LeWM and TC-LeWM use the same policy architecture and training protocol.

Training hyperparameters. Table 4 summarizes the main training hyperparameters used in our experiments.

Table 4: Main training hyperparameters used in our experiments.

Stage	Hyperparameter	Value
World model	Architecture	ViT-Tiny
	Frame skip	4
	Predictor context	$h = 3$
	SIGReg projections	$J = 1024$
	SIGReg weight λ	0.09
	Centering window	$W = 4$ (TC-LeWM)
	Optimizer	AdamW, lr 5×10^{-5}
	Batch size	128
	Training steps	10k (30k for 40-task)
BC policy	Architecture	DiT
	Conditioning	34 visual tokens via cross-attention + task embedding
	Action horizon	$H = 8$
	Optimizer	AdamW, lr 2×10^{-4}
	Batch size	256
	Training steps	40k (4k for single-task)
	Inference	10 Euler steps

A.2 Monte Carlo Analysis Details

Projected mixture model. Following the projected temporal decomposition in Sec. 3.2,

$$x = \mathbf{w}^\top \mathbf{z}_t = \underbrace{\mathbf{w}^\top \bar{\mathbf{z}}_t}_{c} + \underbrace{\mathbf{w}^\top \mathbf{r}_t}_{\varepsilon}, \tag{15}$$

we construct a finite-window Monte Carlo surrogate in which each temporal window has a shared component center and independent within-window variation. Specifically, before temporal centering, projected samples are generated as

$$x_t = c_0 + \eta_t, \qquad c_0 \sim \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}(\mu_k, \tau^2), \qquad \eta_t \sim \mathcal{N}(0, \sigma_r^2), \qquad \mathrm{Cov}(c_0, \eta_t) = 0. \tag{16}$$

Here, c_0 is shared by all samples within a temporal window and η_t represents independent within-window variation. Temporal centering gives

$$c = \bar{x} = c_0 + \bar{\eta}, \qquad \varepsilon_t = x_t - \bar{x} = \eta_t - \bar{\eta}. \tag{17}$$

In the Monte Carlo analysis, we set $\tau = 0$ so that the variance of the shared component is determined entirely by the spread of the component centers. We denote the two controlled variance parameters by

$$v_c \triangleq \mathrm{var}(c_0), \qquad v_\varepsilon \triangleq \mathrm{var}(\eta_t) = \sigma_r^2. \tag{18}$$

For compactness, these controlled parameters are denoted by $\mathrm{var}(c)$ and $\mathrm{var}(\varepsilon)$ in the main-text Monte Carlo analysis and Fig. 2(a–b). Their relation to the finite-window centered quantities is given below.

Randomized center construction. For each Monte Carlo layout m, we sample K raw component centers independently from a uniform distribution,

$$\widetilde{\mu}_k^{(m)} \overset{\text{i.i.d.}}{\sim} \mathcal{U}(-1, 1), \qquad k = 1, \ldots, K. \tag{19}$$

We then center and normalize the sampled centers to obtain $\{\widehat{\mu}_k^{(m)}\}_{k=1}^K$ with zero mean and unit variance. For a target persistent variance v_c, we rescale them as

$$\mu_k^{(m)}(v_c) = \sqrt{v_c}\, \widehat{\mu}_k^{(m)}. \tag{20}$$

Thus, each layout satisfies $\frac{1}{K}\sum_k \left(\mu_k^{(m)}\right)^2 = v_c$. The normalized centers determine the relative component geometry, while v_c directly controls its overall variance.

Monte Carlo protocol. We use $K = 10$ components, matching the number of tasks in each LIBERO suite. Each batch contains $N = 128$ projected samples grouped into temporal windows of length $W = 4$. Each window is assigned to one component, and its samples are generated as

$$x_t = \mu_k + \sqrt{v_\varepsilon}\,\xi_t, \qquad \xi_t \sim \mathcal{N}(0, 1), \tag{21}$$

where v_ε controls the pre-centering within-window variation. We average over $M = 400$ randomized center layouts and $R = 20$ finite-batch realizations per layout. The same component assignments and Gaussian samples are reused across the variance grid for paired comparisons.

For Fig. 2(a), we sweep $v_c \in [0, 4.2]$ with step 0.1 and $v_\varepsilon \in [0.05, 2.5]$ with step 0.05. We instantiate the Raw LeWM objective in the projected Monte Carlo model as

$$\mathcal{L}^{\mathrm{MC}}_{\mathrm{Raw}} = v_\varepsilon + \lambda\,\mathcal{L}_{\mathrm{EP}}(x), \qquad \lambda = 0.09. \tag{22}$$

Here, v_ε serves as a surrogate for $\mathcal{L}_{\mathrm{pred}}$, while $\mathcal{L}_{\mathrm{EP}}(x)$ corresponds to the SIGReg term evaluated on the one-dimensional projected marginal x. The resulting objective landscape is shown in Fig. 2(a). This provides a scale-consistent surrogate for the original objective: the prediction term is represented by the scale of within-window variation along a projected direction, while the EP term is evaluated on the corresponding one-dimensional projected marginal.

EP loss under temporal centering. For Fig. 2(b), we fix $v_\varepsilon = 1.30$ and vary only the persistent variance v_c. For $W = 4$, temporal centering converts this to a centered-residual variance of

$$\mathrm{var}(\varepsilon) = \frac{3}{4} \times 1.30 = 0.975, \tag{23}$$

which is approximately one. For each setting, we evaluate the change in EP loss relative to $v_c = 0$ on both the projected latent x and its temporally centered residual. Because the component center is shared by all samples within a temporal window, it is removed exactly by temporal centering. Consequently, the residual EP loss is invariant to v_c under this construction, whereas the EP loss on x changes with persistent variance.

Finite-window variance conversion. The Monte Carlo parameters (v_c, v_ε) control the variance of the window-shared component c_0 and the pre-centering within-window variation η_t, respectively. The temporal decomposition in the main text instead measures the finite-window projected mean $c = \bar{x}$ and centered residual $\varepsilon_t = x_t - \bar{x}$. For independent η_t within a window of length W,

$$\mathrm{var}(c) = \mathrm{var}(\bar{x}) = v_c + \frac{1}{W} v_\varepsilon, \qquad \mathrm{var}(\varepsilon) = \frac{W - 1}{W} v_\varepsilon. \tag{24}$$

Therefore, for the $W = 4$ windows used in our experiments,

$$v_\varepsilon = \frac{4}{3}\mathrm{var}(\varepsilon), \qquad v_c = \mathrm{var}(c) - \frac{1}{3}\mathrm{var}(\varepsilon). \tag{25}$$

At the representation level, $\mathrm{var}(c)$ and $\mathrm{var}(\varepsilon)$ correspond to the per-dimension projected counterparts of $\mathrm{var}(\bar{\mathbf{z}})$ and $\mathrm{var}(\mathbf{r})$, respectively. We use equation 25 when mapping trained-system measurements onto the Monte Carlo coordinates in Fig. 2(a). Figure 2(c) instead reports the directly measured $\mathrm{var}(\bar{\mathbf{z}})$ and $\mathrm{var}(\mathbf{r})$.

A.3 Representation Analysis Details

Empirical variance allocation. For Fig. 3(a), we apply the same empirical variance calculation to the projector outputs at each of the three training scales. Consecutive cached representations are grouped into windows of four frames, matching the temporal windows used during world model training. For each window, we compute the persistent component $\bar{\mathbf{z}}_t$ and centered residual $\mathbf{r}_t$ using the decomposition defined in the main text.

We compute the empirical variance independently for each latent dimension and then average across dimensions to obtain $\mathrm{var}(\bar{\mathbf{z}})$ and $\mathrm{var}(\mathbf{r})$. The resulting quantities are therefore reported on the same per-dimension variance scale as the unit-variance target of SIGReg. The same calculation is used across all training scales, and the displayed values are averaged over three independently trained seeds.

Latent content attribution. For Fig. 3(b), we quantify how latent variation is associated with task identity, within-episode progress, and trajectory identity. For each evaluation cell, we first perform

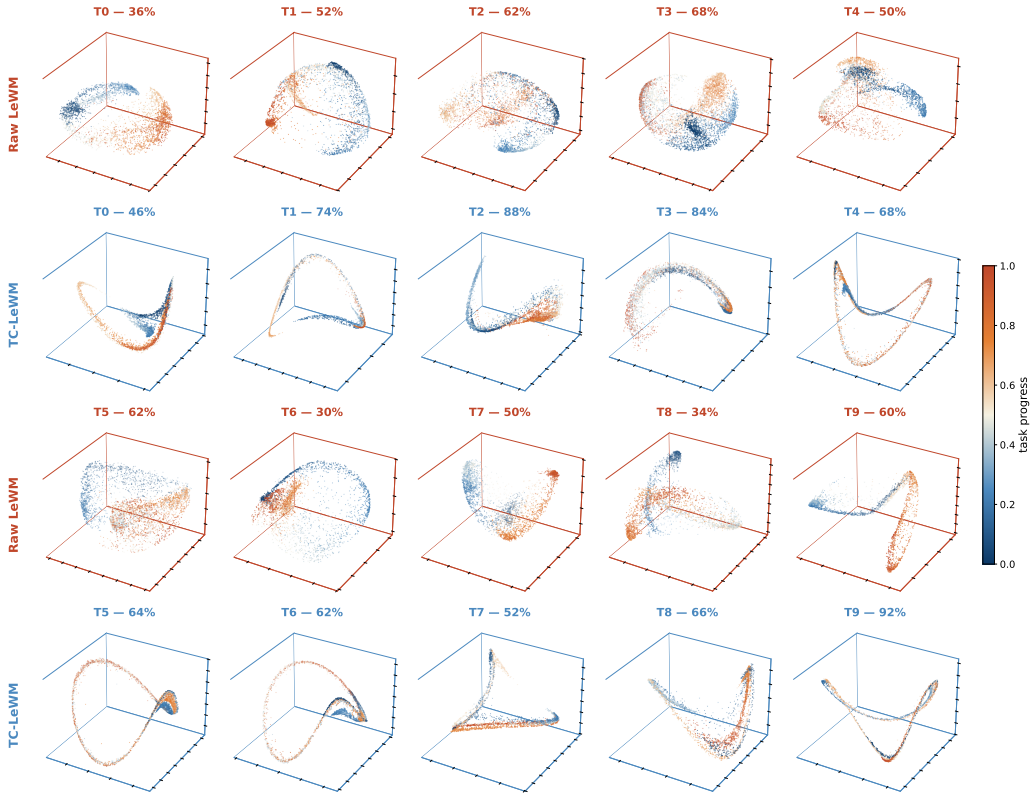

Figure 5: **Per-task 3D PCA visualization of latent representations** on LIBERO-Long under suite-wise 10-task training. Points are colored by normalized within-trajectory progress. Across many tasks, TC-LeWM exhibits smoother and more clearly progress-aligned structures, whereas Raw LeWM more often produces diffuse geometry. The percentages in the titles indicate the corresponding per-task closed-loop success rates for the displayed training seed, averaged over three evaluation seeds.

PCA on the frozen latent representations and retain the top eight principal components. We then apply sequential ANOVA independently to each principal component, using the fixed order task identity $\rightarrow$ within-episode progress $\rightarrow$ trajectory identity. Within-episode progress is discretized into 10 ordered progress stages. Under this ordering, trajectory identity captures only variation beyond that already explained by task identity and within-episode progress, avoiding double counting of progress-related variation.

The attribution for each principal component is weighted by its explained-variance ratio before aggregation across the top eight components. Figure 3(b) reports the TC-LeWM minus Raw LeWM difference in the resulting variance share, such that positive values indicate that TC-LeWM allocates relatively more latent variation to the corresponding factor. The $4.5\times$, $2.5\times$, and $6.9\times$ annotations report the corresponding Raw-to-TC ratios of the trajectory-identity share.

Linear decodability of robot state and dynamics. For Fig. 3(c), we evaluate how linearly decodable robot state and dynamics are from the frozen representations. The analysis is performed separately for each task, with 20% of complete episodes held out for evaluation.

We use ridge regression with regularization coefficient $\lambda = 0.1$. Both latent inputs and regression targets are z-score normalized using statistics from the fitting split. State is decoded directly from $\mathbf{z}_t$. For velocity, we decode the first-order latent difference $\mathbf{z}_{t+1} - \mathbf{z}_t$ into the corresponding first-order robot-state difference, while acceleration analogously uses second-order temporal differences.

Table 5: EP loss evaluated on the full latent representation $\mathbf{z}$ extracted from the frozen encoder after suite-wise 10-task training. Results are mean $\pm$ standard deviation over three training seeds and are reported in units of 10^{-2}. Lower values indicate a distribution closer to the standard Gaussian target.

Method	Object	Spatial	Goal	Long	Avg.
Raw LeWM	0.45 ± 0.02	0.39 ± 0.05	0.40 ± 0.01	0.39 ± 0.02	0.41
TC-LeWM	1.89 ± 0.10	2.16 ± 0.96	2.24 ± 0.15	2.66 ± 1.36	2.24

Decodability is measured using Pearson correlation on the held-out episodes. Pearson correlation is computed independently for each output dimension on the held-out episodes and then averaged across dimensions. The reported points are averaged over the four LIBERO suites and three independently trained seeds, with error bars indicating the standard deviation across training seeds.

Stage-wise closed-loop analysis. For Fig. 3(d), we decompose each rollout into grasping and placement stages using simulator object states. For each task, we identify the movable object(s) of interest from the task definition and track their 3D positions throughout evaluation. A rollout is considered to have passed the grasping stage if any target object is lifted by at least $3\,\mathrm{cm}$ relative to its settled initial height. The placement-stage success rate is then defined conditionally as the fraction of grasp-successful rollouts that eventually complete the task.

We apply this decomposition only to tasks whose successful executions involve lifting a target object. In LIBERO-Goal, three tasks do not satisfy this criterion and are therefore excluded from the stage-wise analysis, leaving seven grasp-and-place tasks.

A.4 Additional Latent-Space Analysis

Per-task PCA visualization. For the frozen LIBERO-Long representations learned under suite-wise 10-task training, we perform PCA separately for each task and visualize the first three principal components. Figure 5 compares Raw LeWM and TC-LeWM on one training seed, with points colored by normalized within-trajectory progress. Across many tasks, TC-LeWM exhibits connected and more clearly progress-aligned structures, whereas Raw LeWM more often produces diffuse shapes. In the per-task comparison between Raw LeWM and TC-LeWM, improvements in closed-loop success are often accompanied by more organized and manifold-like latent trajectories.

Gaussianity of the full latent. After world model training, we freeze the encoder and evaluate the EP loss of the resulting full latent representation $\mathbf{z}$ under suite-wise 10-task training. As shown in Table 5, Raw LeWM consistently exhibits a much lower EP loss, as expected because SIGReg directly regularizes the full latent toward the Gaussian target during training. In contrast, TC-LeWM applies SIGReg only to the centered residual and removes the Gaussian constraint from the full latent $\mathbf{z}$. Consequently, the frozen TC-LeWM latent deviates more strongly from the Gaussian target. This higher EP loss is therefore an intended consequence of the method: releasing the full latent from the Gaussian target allows structured and potentially multimodal organization to emerge in $\mathbf{z}$, while the centered residual remains the target of SIGReg.

Latent radial structure. To further characterize the non-Gaussian structure of the full latent, Figure 6 compares the latent norm distributions of Raw LeWM and TC-LeWM on LIBERO-Long under 10-task training. We first report the dimension-normalized norm $\|\mathbf{z}_t\|/\sqrt{D}$, where D is the latent dimension. This normalization preserves the relative scale difference between the two methods while providing a natural reference: for a high-dimensional standard isotropic Gaussian, the norm concentrates near $\sqrt{D}$, and hence $\|\mathbf{z}_t\|/\sqrt{D}$ is concentrated near one. Consistent with its low EP loss, Raw LeWM is tightly concentrated around this Gaussian radial scale, whereas TC-LeWM spans a substantially broader range and exhibits multiple radial modes.

To separate this structural difference from a simple change in global scale, we additionally normalize each representation by its own mean norm, $\|\mathbf{z}_t\|/\overline{\|\mathbf{z}\|}$. After removing the difference in average scale, TC-LeWM still exhibits a clearly more multimodal radial distribution, while Raw LeWM remains largely unimodal. These results indicate that releasing the full latent from the Gaussian target changes not only its overall radial scale but also its distributional shape, allowing richer radial organization to emerge.

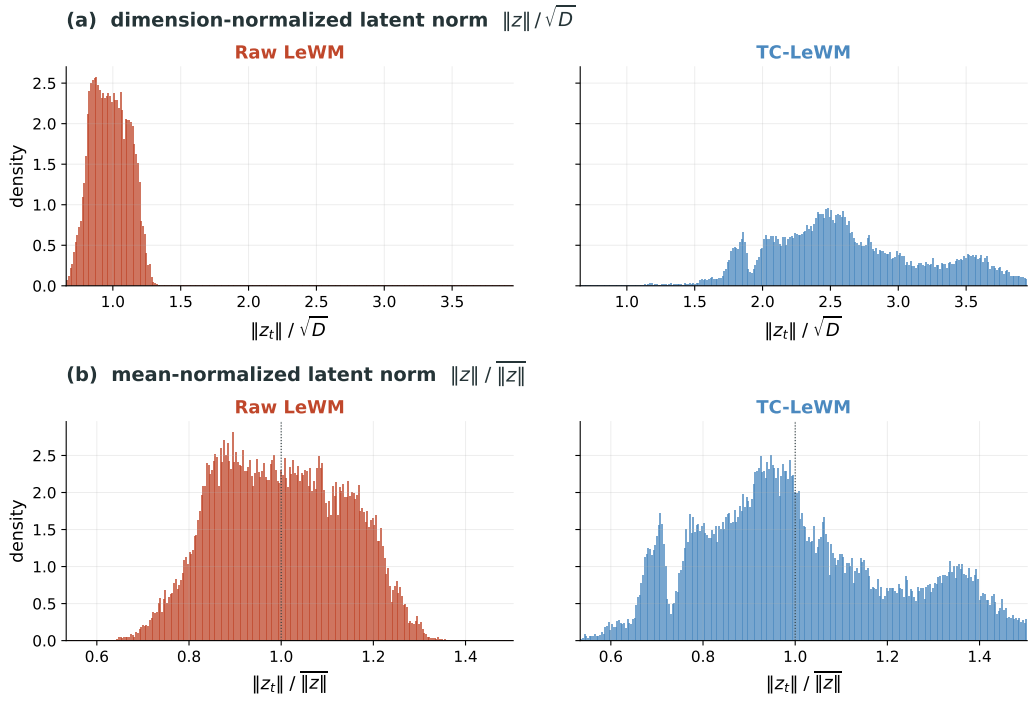

Figure 6: Latent norm distributions on LIBERO-Long under 10-task training. (a) Dimension-normalized latent norm $\|\mathbf{z}_t\|/\sqrt{D}$, which preserves the relative latent scale while using the typical norm of a high-dimensional standard Gaussian as a reference near one. Raw LeWM remains concentrated near this Gaussian radial scale, whereas TC-LeWM spans a substantially broader range with multiple radial modes. (b) Mean-normalized latent norm $\|\mathbf{z}_t\|/\overline{\|\mathbf{z}\|}$, which removes the difference in average scale between the two representations. The multimodal radial structure of TC-LeWM remains clearly visible after normalization, showing that the difference cannot be explained solely by global rescaling.

Table 6: Sensitivity of TC-LeWM to the temporal-centering window size W on LIBERO-Long under suite-wise 10-task training. Results are mean±standard deviation over three training seeds. The temporal context used by the predictor and the remaining training setup are kept fixed.

W	2	4 (default)	8	16
Success (%)	62.0 ± 3.4	63.6 ± 6.8	$\mathbf{65.8 \pm 3.5}$	52.1 ± 3.8

A.5 Ablation Studies

Sensitivity to temporal window size. We evaluate the sensitivity of TC-LeWM to the temporal-centering window size W on LIBERO-Long under suite-wise 10-task training. As shown in Table 6, performance is similar for $W = 2$, 4, and 8, with average success rates of 62.0%, 63.6%, and 65.8%, respectively. This suggests that TC-LeWM is not particularly sensitive to the precise window size within a relatively short temporal range. Performance decreases to 52.1% when the window is increased to $W = 16$, indicating that overly long windows are less suitable for the local temporal decomposition.

We use $W = 4$ by default because it matches the four-frame training sequence used to organize world model training batches. We do not evaluate $W = 1$, since temporal centering then gives $\mathbf{r}_t = \mathbf{0}$ identically and therefore does not provide a meaningful residual regularization target.

Sensitivity to SIGReg strength in Raw LeWM. We further examine whether the downstream limitation of Raw LeWM can be addressed simply by reducing the SIGReg strength. As shown in

Table 7: Sensitivity of Raw LeWM to the SIGReg weight λ on LIBERO-Long under suite-wise 10-task training. Results are mean±standard deviation over three training seeds. The model architecture and the remaining training setup are kept fixed.

λ	0.005	0.01	0.03	0.06	0.09
Success (%)	43.4 ± 1.5	$\mathbf{50.3 \pm 6.8}$	49.5 ± 2.9	47.1 ± 0.6	44.7 ± 4.3

Table 7, moderately decreasing λ from the default value of 0.09 improves downstream success, with the best performance obtained at $\lambda = 0.01$. However, further reducing the regularization strength to $\lambda = 0.005$ degrades performance, indicating that the improvement does not continue as the regularization pressure is reduced. Moreover, even the best Raw LeWM result remains substantially below TC-LeWM. These results suggest that reducing the regularization pressure alone cannot fully resolve the downstream representation limitation of Raw LeWM.