DDP-WM: Disentangled Dynamics Prediction for Efficient World Models

Shicheng Yin [*1] **Kaixuan Yin** [*1] **Weixing Chen** [1] **Yang Liu** [1 2 †] **Guanbin Li** [1] **Liang Lin** [1 2]

Abstract

World models are essential for autonomous robotic planning. However, the substantial computational overhead of existing dense Transformer-based models significantly hinders real-time deployment. To address this efficiency-performance bottleneck, we introduce DDP-WM, a novel world model centered on the principle of Disentangled Dynamics Prediction (DDP). We hypothesize that latent state evolution in observed scenes is heterogeneous and can be decomposed into sparse primary dynamics driven by physical interactions and secondary context-driven background updates. DDP-WM realizes this decomposition through an architecture that integrates efficient historical processing with dynamic localization to isolate primary dynamics. By employing a cross-attention mechanism for background updates, the framework optimizes resource allocation and provides a smooth optimization landscape for planners. Extensive experiments demonstrate that DDP-WM achieves significant efficiency and performance across diverse tasks, including navigation, precise tabletop manipulation, and complex deformable or multi-body interactions. Specifically, on the challenging Push-T task, DDP-WM achieves an approximately $9\times$ inference speedup and improves the MPC success rate from 90% to 98% compared to state-of-the-art dense models. The results establish a promising path for developing efficient, high-fidelity world models. Code is available at https://hcplab-sysu.github.io/DDP-WM/.

1. Introduction

Endowing machines with the ability to perform autonomous planning and decision-making in complex, dynamic environments is a core objective of embodied intelligence research. In this pursuit, world models, which are capable of predicting future world states, play a pivotal role. By learning the dynamic laws of an environment directly from high-dimensional pixel inputs, world models enable an agent to mentally simulate the potential consequences of different action sequences without real-world interaction. This capability is the cornerstone of advanced planning algorithms such as Model Predictive Control (MPC) (Garcia et al., 1989), providing a powerful theoretical framework for solving complex robotic manipulation and navigation tasks (Liu et al., 2025).

In recent years, building world models upon pre-trained visual representations (e.g., DINOv2 (Oquab et al., 2024)) has become a frontier direction. Cutting-edge works like DINO-WM (Zhou et al., 2025) have demonstrated that architectures based on Vision Transformers (ViTs (Dosovitskiy et al., 2021)) exhibit strong zero-shot planning capabilities by accurately capturing complex physical dynamics. However, this **indiscriminate** computational paradigm introduces a significant efficiency bottleneck: the model applies the same expensive self-attention computation to all image patches, regardless of whether they correspond to moving objects or static backgrounds. In most physical interaction scenarios, the regions undergoing actual change constitute only a small fraction, meaning the vast majority of computation is wasted on redundant recalculations for static backgrounds. For real-time MPC applications, which require hundreds or even thousands of simulations per second, prediction inference speed is critical for the deployment and application of advanced world models in real robotic systems. Indeed, our experiments reveal a significant practical bottleneck: even the current state-of-the-art dense model (DINO-WM) requires nearly two minutes (120 seconds) for a single MPC decision cycle on a representative manipulation task like Push-T. For many applications requiring continuous interaction with the physical world, such latency poses a fundamental challenge, providing direct motivation for our work.

To intuitively reveal the nature of this computational redundancy, we conducted an analysis of the internal workings of a state-of-the-art (SOTA) dense model (DINO-WM) and the dynamic data it processes. As shown in Figure 1a, we visualized the evolution of features in each layer of its predictor

[*]Equal contribution [1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China [2]X-Era AI Lab. Correspondence to: Yang Liu <liuy856@mail.sysu.edu.cn>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

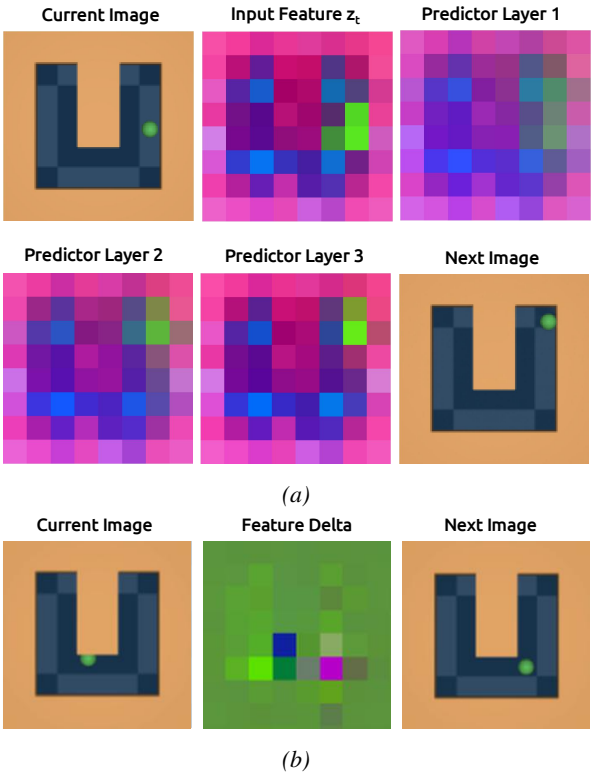


Figure 1. **(a) PCA visualization of internal feature evolution in a dense model. (b) PCA of the difference between consecutive ground-truth features.** In (a), PCA projection of features from each predictor layer shows background regions remain largely static, revealing the computational redundancy of dense models. In (b), the PCA of the feature difference shows most regions (green) have near-zero change, demonstrating the inherent sparsity of physical dynamics.

relative to the input using PCA. The results indicate that for the vast majority of background regions, the change in their features is almost negligible after passing through multiple layers of expensive self-attention computations. This confirms that dense models waste a significant amount of computational power on redundant recalculations for static regions. Furthermore, we find that the root cause of this computational waste lies in the inherent sparsity of physical dynamics. As depicted in Figure 1b, we computed and visualized the difference between the feature maps of two consecutive frames. Only a small portion of feature patches undergo significant changes.

Based on this insight, we propose DDP-WM (Disentangled Dynamics Prediction World Model), a framework that allocates computational resources according to the nature of the dynamics. This framework identifies the sparse regions where primary dynamics occur via a dynamic localization network and focuses computational resources on them using a powerful primary predictor. Concurrently, an

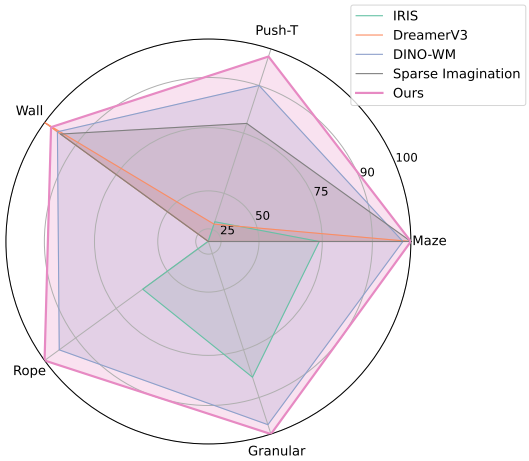


Figure 2. **Overall Performance on Key Benchmarks.** For comparability, Success Rates are plotted directly, while Chamfer Distance (CD) values are normalized via the formula $(\max_{CD} - CD_i)/(\max_{CD} - \min_{CD}) \times 100$. Some baselines are omitted from certain environments due to near-zero performance; see Table 1 for full values.

efficient Low-Rank Correction Module (LRM) handles the context-driven background updates induced by the primary dynamics at a very low computational cost, thereby optimally allocating computational resources while providing a smooth optimization landscape for the planner. Figure 2 provides a high-level summary of our method's comprehensive performance gains across key benchmarks.

Our main contributions can be summarized as follows:

- We introduce the Disentangled Dynamics Prediction (DDP) paradigm, which posits that scene dynamics can be fundamentally decoupled into sparse, action-driven "primary dynamics" and broader "context-driven background updates."
- We propose DDP-WM, a novel architecture that instantiates the Disentangled Dynamics Paradigm (DDP). Its key innovation is the Low-Rank Correction Module (LRM), which leverages a unidirectional, causal cross-attention mechanism to efficiently capture background dynamics with minimal computational cost, thereby ensuring feature-space consistency.
- We demonstrate that DDP-WM establishes a new state of the art in both performance and efficiency. On the challenging Push-T benchmark, for instance, it improves the success rate from 90% to a near-perfect 98% while achieving an approximately 9x speedup. Our analysis further reveals that this closed-loop success is critically enabled by the smooth, tractable optimization landscape our method provides for the planner.

We will release our code, models, and supplementary mate-

rials to ensure the reproducibility of our results.

Conflict of Interest Disclosure. The authors have no financial conflicts of interest to disclose.

2. Related Work

2.1. Visual World Models

Model-based decision-making, where an agent makes informed choices by simulating the future, has been a long-standing goal in reinforcement learning and robotics (Ha & Schmidhuber, 2018; Hafner et al., 2019). The field has undergone a paradigm shift from early approaches that directly predicted pixels (Łukasz Kaiser et al., 2020) to modeling dynamics in a compact latent space. Starting with the pioneering work of Ha & Schmidhuber (2018), PlaNet (Hafner et al., 2019) and the subsequent Dreamer series (Hafner et al., 2020; 2021; 2025) have matured this latent dynamics modeling paradigm. They compress high-dimensional observations into a low-dimensional latent space using a Variational Autoencoder or similar methods, and then perform dynamics prediction with recurrent or sequence models. Subsequently, MWM (Seo et al., 2023) further demonstrated the effectiveness of decoupling representation and dynamics learning through masked autoencoders. However, these models rely on image reconstruction as a training objective, which can be affected by background noise in images, making it difficult to capture the fine-grained physical details required for high-precision robotic manipulation.

In recent years, Transformer-based sequence models, such as IRIS (Micheli et al., 2023), STORM (Zhang et al., 2023), and V-JEPA 2 (Assran et al., 2025), have become the state-of-the-art approach due to their powerful modeling capabilities, achieving unprecedented high accuracy in capturing complex physical dynamics. An important trend is to perform dynamics prediction directly in the rich feature space provided by pre-trained visual models (e.g., DINOv2 (Oquab et al., 2024)) to avoid information loss from reconstruction, as exemplified by **DINO-WM** (Zhou et al., 2025). We note that IRIS and STORM adopt a discrete token prediction paradigm, whereas our work and DINO-WM perform continuous latent-space prediction. This paradigm difference makes direct numerical comparison across families less straightforward. Meanwhile, large-scale video generation models, such as Genie (Bruce et al., 2024) and GAIA-2 (Russell et al., 2025), are also considered a form of generalized world models. However, they focus more on open-ended video generation rather than the precise dynamics prediction for closed-loop planning that is the focus of this paper.

Despite the powerful performance of the aforementioned Transformer-based models, they invariably rely on dense self-attention computation over all visual tokens. Their computational complexity, which is quadratic with respect to the sequence length, severely hinders their application in real-time Model Predictive Control (MPC) that requires high-frequency simulations. Our work aims to significantly improve computational efficiency through a decoupled, sparse prediction framework, while maintaining or even improving upon their performance.

A broader discussion of related work, including efficiency optimizations for general Transformers and other sparse or structured modeling approaches, is provided in Appendix B.

3. Methodology

To address the inherent contradiction between computational efficiency and planning performance in existing world models, we propose an innovative decoupled dynamics prediction framework. The core idea is to decompose complex visual dynamics into two sub-problems of different natures and to design specialized, efficient computational modules for them. Figure 3 provides a complete overview of the decoupled dynamics prediction framework.

3.1. Background and Problem Formulation

We formalize the visual control task as a Partially Observable Markov Decision Process (POMDP). Within this framework, a world model built upon pre-trained features typically consists of two core components:

1. A fixed, pre-trained observation model g_ϕ: It is responsible for mapping high-dimensional image observations $o_t \in \mathbb{R}^{H\times W\times 3}$ to a series of latent features (patch tokens) containing rich spatial information, i.e., $z_t = g_\phi(o_t)$, where $z_t \in \mathbb{R}^{N\times D}$. In this paper, we adopt a frozen DINOv2 as our observation model, which is consistent with the choice in cutting-edge works like DINO-WM.

2. A learnable transition model f_θ: Its task is to predict the latent state at the next time step, $\hat{z}_{t+1} = f_\theta(z_{\leq t}, a_{\leq t})$, based on the history of state-action sequences $(z_{\leq t}, a_{\leq t})$.

Our core contribution revolves around designing a novel transition model, f_θ, that far surpasses existing state-of-the-art (SOTA) methods in both performance and efficiency.

3.2. Core Insight: Decoupling Primary Dynamics and Context-driven Background Updates

Dense world models, such as DINO-WM, predict the future by applying a uniform transformation to all tokens, which leads to massive computational redundancy in scenarios

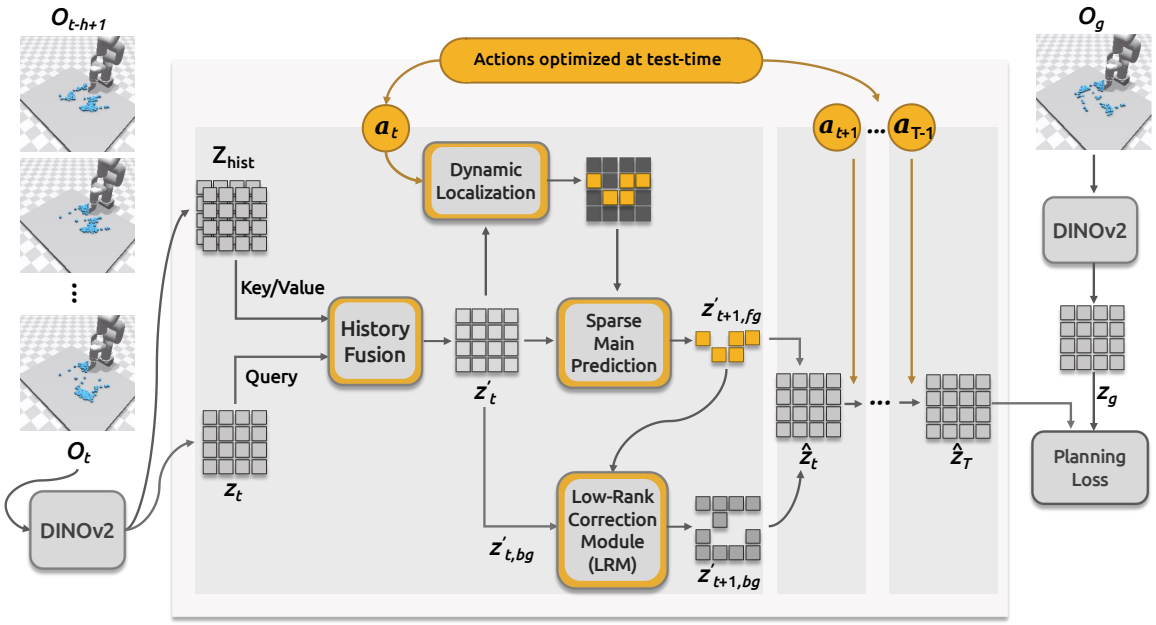


Figure 3. **Overview of the DDP-WM Framework.** Our framework performs prediction through a four-stage decoupled process. **(1) Historical Information Fusion:** The features of the current frame, $\mathbf{z}_t$, query historical frame features $\mathbf{Z}_{\text{hist}}$ via cross-attention to obtain temporally-aware augmented features $\mathbf{z}'_t$. **(2) Dynamic Localization:** A lightweight network receives $\mathbf{z}'_t$ and the action $\mathbf{a}_t$ to predict a sparse mask $\mathbf{M}$ that contains only the primary dynamic regions. **(3) Sparse Primary Dynamics Prediction:** A powerful primary predictor focuses all computation on the sparse foreground features identified by $\mathbf{M}$ to predict the next frame's foreground features, $\mathbf{z}'_{t+1,\text{fg}}$, with high precision. **(4) Contextual Background Update:** The background features of the current frame, $\mathbf{z}_{t,\text{bg}}$, are updated at a very low cost by querying the newly predicted foreground features $\mathbf{z}'_{t+1,\text{fg}}$ via cross-attention. Finally, the updated foreground and background are combined to constitute the complete latent state of the next frame.

with sparse changes. More severely, this fully-connected computation pattern can cause the model to learn non-causal spurious correlations within the scene, harming its generalization ability and robustness. An intuitive optimization is to employ sparse computation, i.e., only predicting for regions undergoing change while performing feature copying for static regions. However, our preliminary experiments (detailed in Section 4.3) reveal a critical phenomenon: while such simple sparse models achieve lower error in open-loop prediction, their planning success rate in closed-loop Model Predictive Control (MPC) plummets.

We argue that a key factor in this problem is an overlooked phenomenon: in pre-trained representations based on self-attention mechanisms (like DINOv2), any local feature implicitly encodes its relationship with the global context. Consequently, when a primary object moves, even if the pixels in static regions remain unchanged, their features must undergo subtle adjustments due to the change in spatial context (see Figure 1a and Appendix A for empirical evidence). We term this context-aware adjustment of the background, triggered by foreground changes, as context-driven background updates.

Simple sparse models, with their copy-paste rule, violate this intrinsic property of the feature space. This results in discontinuous cliffs in the cost landscape provided to the planner, which is the root cause of planning failures.

Based on this insight, we decouple and separately model two types of dynamics:

1. **Primary Dynamics:** High-frequency, non-linear changes on foreground objects caused by direct physical interactions, reflecting the core causal chain of the scene.

2. **Context-driven Background Updates:** Low-frequency, context-driven feature adjustments in background regions, induced by the primary dynamics. We further make a key assumption that this seemingly complex global adjustment is inherently **low-rank**. More formally, this posits that the set of all background update vectors, $\{\Delta\mathbf{z}_i\}$, lies in a low-dimensional subspace, which is mathematically equivalent to their corresponding Gram matrix being of low rank. We provide decisive empirical evidence for this foundational assumption in Appendix A, where a Principal Component Analysis (PCA) directly inspects the effective rank of this structure.

To this end, we have designed a decoupled dynamics prediction framework.

3.3. Decoupled Dynamics Prediction Framework

Our framework (as shown in Figure 3) achieves efficient and high-fidelity dynamics prediction through a four-stage process. It first injects temporal dynamics into the current state through a historical information fusion module, followed by the Dynamic Localization Network, Sparse Primary Dynamics Predictor, and Low-Rank Correction Module, which jointly complete the prediction for the next frame.

3.3.1. Stage 1: Historical Information Fusion Module

To enable the model to understand higher-order dynamics such as velocity and acceleration, we introduce an efficient historical information fusion module before the core prediction process begins.

- **Mechanism and Efficiency:** Unlike dense models such as DINO-WM, which simply stack the features of all historical frames and feed them into a full Transformer, our method accomplishes the injection of historical information via a single layer of cross-attention (CA).
  - **Query:** The latent features of the current frame, $\mathbf{z}_t$.
  - **Key/Value:** The set of latent features from all historical frames, $\mathbf{Z}_{\text{hist}} = \{\mathbf{z}_{t-h+1}, ..., \mathbf{z}_{t-1}\}$.
- **Workflow:** Each feature vector of the current frame queries the complete history information pool via cross-attention to aggregate relevant temporal dynamics, and updates itself through a residual connection:

$$\mathbf{z}'_t = \mathbf{z}_t + \text{CA}(\text{Q} = \mathbf{z}_t, \text{K} = \mathbf{Z}_{\text{hist}}, \text{V} = \mathbf{Z}_{\text{hist}}) \quad (1)$$

  The features $\mathbf{z}'_t$, augmented by this module, contain an implicit encoding of the current dynamics and serve as the input for the subsequent prediction stages.

3.3.2. Stage 2: Dynamic Localization Network

This module is responsible for efficiently and accurately identifying the sparse regions where primary dynamics will occur in the next frame.

- **Architecture and Task:** This module is responsible for efficiently and accurately identifying the sparse regions where primary dynamics will occur. A dedicated ViT receives the temporally-aware current state latent features $\mathbf{z}'_t$ and the action $\mathbf{a}_t$. To improve localization accuracy, it predicts change probabilities for the corresponding 2×2 sub-regions of each image patch, outputting a probability map $P_{\text{sub}} \in \mathbb{R}^{N\times 4}$. The final sparse binary mask $\mathbf{M} \in \{0,1\}^N$ is then generated by thresholding these probabilities: a patch is marked as changed if any of its sub-regions' predicted change exceeds a preset threshold τ. This process is defined as:

$$m_i = \begin{cases} 1 & \text{if } \max(P_{\text{sub},i}) > \tau \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i = 1, \ldots, N \quad (2)$$

  where m_i is the i-th element of the mask $\mathbf{M}$, and $P_{\text{sub},i}$ denotes the 4 change probabilities for the sub-regions of the i-th patch.
- **Training Supervision:** The Dynamic Localization Network is trained with a Binary Cross-Entropy (BCE) loss. Ground-truth masks are generated by computing per-pixel absolute differences between adjacent frames $|o_{t+1} - o_t|$ and marking patches where the difference exceeds a fixed threshold as dynamic.

3.3.3. Stage 3: Sparse Primary Dynamics Predictor

This module is the primary computational unit, responsible for modeling the primary dynamics identified by the mask $\mathbf{M}$ with high precision.

Implementation: From the complete, history-fused input features $\mathbf{z}'_t$, we use the mask $\mathbf{M}$ to extract the subset of dynamic patch tokens, $\mathbf{z}'_{t,\text{fg}}$. A powerful primary predictor (e.g., a ViT-based model) then focuses its computation entirely on this sparse subset to predict the high-precision next-frame foreground features $\mathbf{z}'_{t+1,\text{fg}}$.

Adaptive Sparse Size: We employ an adaptive sizing strategy to handle dynamically varying sparse inputs, balancing hardware efficiency with computational accuracy. The detailed mechanism is deferred to Appendix E.

3.3.4. Stage 4: Low-Rank Correction Module (LRM)

This module is designed to efficiently model the context-driven background updates induced by the primary dynamics, at a very low computational cost. Its core design architecturally embodies an inductive bias that mimics the unidirectional causal flow of physics: the primary dynamics are computed first, and the background update must proceed based on their result, rendering the information flow irreversible.

Specifically, we employ a single-layer cross-attention mechanism, where the information flow is asymmetrically designed as follows:

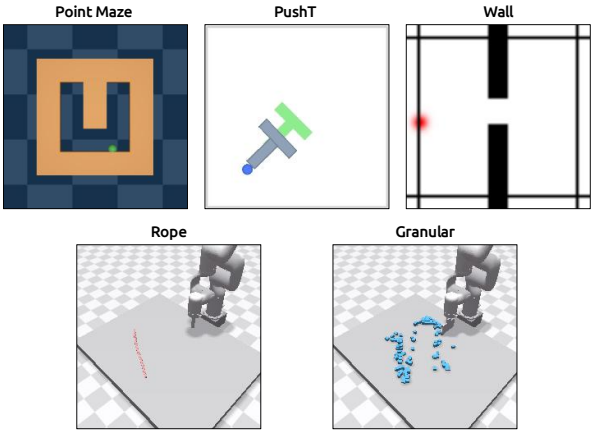


Figure 4. **Overview of Evaluation Environments.** Sample frames from the five simulated task domains used in our experiments.

- **Query:** The set of background patch features $\mathbf{z}'_{t,\text{bg}}$.
- **Key/Value:** The newly predicted foreground patch features $\mathbf{z}'_{t+1,\text{fg}}$.

This asymmetric setup forces each background token to passively query the completed foreground predictions, thereby modeling the background update as a direct consequence of the primary dynamics. Each background patch aggregates information about the foreground changes via the attention mechanism to generate an update vector. Finally, the new background features are updated through a residual connection:

$$\mathbf{z}'_{t+1,\text{bg}} = \mathbf{z}'_{t,\text{bg}} + \text{CA}(\text{Q} = \mathbf{z}'_{t,\text{bg}}, \text{K} = \mathbf{z}'_{t+1,\text{fg}}, \text{V} = \mathbf{z}'_{t+1,\text{fg}}) \tag{3}$$

This operation achieves a self-consistent update of the background features at a very low computational cost, providing a smooth optimization landscape for the downstream planner.

3.4. Planning with DDP-WM: Model Predictive Control

We embed the trained DDP-WM as the dynamics model within a Model Predictive Control (MPC) framework for online trajectory planning. We employ the standard Cross-Entropy Method (CEM) as the optimizer. The detailed algorithm is deferred to Appendix D.

Sparse MPC Cost Mask. For the cost function design, the conventional approach is to compute the Mean Squared Error (MSE) between all features of the predicted final state and the goal state. However, we find that this dense computation introduces unnecessary noise. Therefore, we propose a **Sparse MPC Cost Mask** strategy. Specifically, we first compute the pixel-wise difference between the current observation image and the goal image to generate a binary mask $\mathbf{M}_{\text{task}}$, which has a value of 1 only in the differing regions. When calculating the MPC cost, we only consider the feature error in these task-relevant regions:

$$\mathcal{L}_{\text{MPC}} = \text{MSE}(\hat{\mathbf{z}}_T \odot \mathbf{M}_{\text{task}}, \mathbf{z}_{\text{goal}} \odot \mathbf{M}_{\text{task}}) \tag{4}$$

This strategy focuses the planner's cost evaluation on the regions that truly need to change, filtering out interference from the static background and making the optimization process more efficient and stable. In the ablation studies in Section 4, we will quantitatively analyze the effectiveness of this strategy.

4. Experiments

4.1. Experimental Setup

Environments. To comprehensively evaluate the generalization ability of our method, we select five simulated environments with diverse dynamic characteristics and task complexities (shown in Figure 4), covering a wide range of scenarios from simple navigation to complex physical interactions:

- **PointMaze & Wall:** Two 2D navigation tasks to evaluate the model's capabilities in basic kinematics and spatial reasoning.
- **Push-T:** A representative table-top manipulation task that requires the model to understand rigid-body contact dynamics to achieve precise pose control.
- **Rope & Granular:** Two more challenging manipulation tasks involving the complex dynamics of a deformable body (rope) and a multi-body system (granular materials), respectively, placing higher demands on the model's understanding of physics.

Evaluation Metrics. We primarily focus on metrics across the following three dimensions:

- **Model Predictive Control Success Rate (SR ↑):** For the navigation and Push-T tasks, we evaluate the agent's success rate in reaching the designated goal state under MPC planning.
- **Chamfer Distance (CD ↓):** In tasks such as Rope and Granular, where defining a binary success state is difficult, we evaluate the Chamfer Distance between the final state and the goal state. A lower value for this metric signifies more precise manipulation.
- **Computational Efficiency:** We measure efficiency gains in terms of theoretical Floating-Point Operations (FLOPs), single-step inference throughput (Throughput), and single MPC decision time (Latency).

Table 1. Comparison of MPC planning performance in the five simulated environments. Results for IRIS, DreamerV3, and Sparse Imagination are from their original papers; DINO-WM is reproduced under the same protocol.

Model	PointMaze (SR ↑)	Push-T (SR ↑)	Wall (SR ↑)	Rope (CD ↓)	Granular (CD ↓)
IRIS	74%	32%	4%	1.11	0.37
DreamerV3	100%	30%	100%	2.49	1.05
Sparse Imagination	100%	78.3%	95%	-	-
DINO-WM	98%	90%	96%	0.41	0.26
DDP-WM (Ours)	**100%**	**98%**	**98%**	**0.31**	**0.24**

Baselines. Our primary baseline is DINO-WM, the current state-of-the-art dense world model based on pre-trained features. To ensure a fair and direct comparison, all our experimental environments, datasets, and the core parameters of the MPC planner (CEM) strictly adhere to the official DINO-WM settings. This allows us to precisely attribute the differences in performance and efficiency to the fundamental distinction between our proposed decoupled dynamics framework and the dense framework.

Implementation Details. Specific implementation details, including model architectures, training hyperparameters, and planner parameters, are deferred to Appendix C.

4.2. Quantitative Comparison: Planning Performance and Computational Efficiency

4.2.1. Planning Performance

We conducted a comprehensive performance comparison between our full method (Ours) and the DINO-WM across the five environments. The results are presented in Table 1.

Table 1 shows that our method matches or surpasses the SOTA dense model on all tasks. The most significant gain is on the challenging Push-T task, where our 98% success rate substantially outperforms DINO-WM's 90%[1], demonstrating that our approach provides the planner with higher-quality future predictions.

To further demonstrate the generality of our DDP framework, we applied it to a real-world robotic setting that uses DINOv3 ViT-L/16 as the visual encoder, with the predictor trained on the DROID dataset (Khazatsky et al., 2024), and evaluated following the protocol of Terver et al. (2026). Our full pipeline achieves an Action Score of 47.90 ± 0.62 with a $4.26\times$ speedup, surpassing both the JEPA-WMs dense baseline (46.5 ± 0.4, $1\times$) and V-JEPA 2-AC (42.9 ± 2.5, as reported therein). The dynamic localization network correctly classifies illumination changes and background human movement as static, confirming robustness under real-world noise.

[1]Over 13 random seeds, the Push-T success rate is $91.23 \pm 5.51\%$ for DDP-WM vs. $88.31 \pm 6.42\%$ for DINO-WM (paired t-test: $t(12) = 2.91$, $p \approx 0.013$), confirming statistical significance.

Table 2. Comparison of theoretical computational cost (FLOPs) for a single forward inference step.

Task	DINO-WM	Ours	FLOPs Reduction
Push-T	23 G	**2.5 G**	**9.2**×
Wall	7.8 G	**2.5 G**	**3.1**×

Table 3. Comparison of single-step inference throughput (samples/sec, ↑). Tested on a single NVIDIA 2080 Ti with a batch size of 128.

Task	DINO-WM	Ours	Speedup
Push-T	170	**1563**	**9.2**×
Wall	802	**2170**	**2.7**×

4.2.2. Computational Efficiency

As shown in Table 2, our method achieves a massive reduction in theoretical computational cost (FLOPs). In the dynamically complex Push-T task, DDP-WM's computational cost is only one-tenth that of the dense model.

This theoretical advantage translates directly into faster practical inference speed. In the single-step inference throughput test (Table 3), DDP-WM achieves a 9.2x improvement on the Push-T task.

This efficiency advantage is further reflected in the complete MPC decision loop (Table 4). For the Push-T task, which requires 30 CEM iterations, our decision time is only 16 s, a 7.5x speedup compared to DINO-WM's 120 s, enabling higher-frequency control.

These data demonstrate that our decoupled framework leads to significant gains in computational efficiency while also improving planning performance.

4.3. Ablation Study

To systematically validate the necessity of each design within our framework, we conducted a series of detailed ablation studies on the most challenging Push-T task. We primarily investigate the effectiveness of two core components: the LRM, the Sparse MPC Cost Mask (MPC Mask) and the Dynamic Localization Network.

We compare our full method (denoted as ✓ for both compo-

Table 4. Comparison of single MPC decision loop time (s, ↓). Tested on a single NVIDIA A5880.

Task / Iterations	DINO-WM	Ours	Speedup
PointMaze / 10	39 s	**5.5 s**	**7.1**×
Push-T / 30	120 s	**16 s**	**7.5**×
Wall / 10	12 s	**4.2 s**	**2.9**×
Rope / 10	12 s	**4.3 s**	**2.8**×
Granular / 30	35 s	**13 s**	**2.7**×

Table 5. Impact of localization quality on 5-step open-loop prediction pixel error (see Appendix F for details) in Push-T, ↓.

Localization Method	w/o LRM	w/ LRM
Patch-level	788	468
High-Precision	427	361

nents) with variants where these components are removed. The results are presented in Table 7.

We further ablate the high-precision localization mechanism. As shown in Table 8, the mechanism brings a decisive improvement to the quality of mask prediction. More importantly, this improvement in localization accuracy translates directly into more accurate overall dynamics prediction, as shown in Table 5.

4.4. Analysis

The ablation study reveals a core question: Why does the LRM bring such a significant improvement in closed-loop performance? To explain this at a deeper level, we design experiments to probe and visualize the optimization landscape that different models create for the planner.

4.4.1. The Paradox of Open-Loop Prediction

To investigate this, we first compare the open-loop prediction accuracy. For this specific analysis, we intentionally measure error in pixel space (see Appendix F for calculation details). This choice is crucial for a fair ablation, as the 'Naive Sparse' model does not predict background tokens, and a direct feature MSE comparison would be inequitable. Pixel error, in contrast, provides an objective measure of physical accuracy. We task the models with predicting a 5-step trajectory; the results are in Table 6.

As shown in Table 6, a noteworthy discrepancy emerges: in open-loop prediction, the error of the Naive Sparse model (w/o LRM) is nearly identical to that of our full DDP-WM model, and both are significantly lower than the dense DINO-WM. This indicates that the LRM provides almost no help in improving open-loop prediction accuracy. Why, then, is this module, seemingly useless in open-loop, the key to success in closed-loop planning?

Table 6. Comparison of 5-step open-loop prediction pixel error (number of error pixels, ↓).

Model	PointMaze	Push-T	Wall
DINO-WM (Dense)	81	524	111
DDP-WM (Ours)	36	361	9
Naive Sparse (w/o LRM)	41	427	15

Table 7. Ablation study on the Push-T task, showing the contribution of the Low-Rank Correction Module (LRM) and Sparse MPC Cost Mask to planning success rate.

Model Variant	LRM (c)	MPC Mask (m)	SR ↑
DDP-WM (Ours)	✓	✓	**98%**
	✓	×	90%
	×	✓	70%
Naive Sparse	×	×	62%

4.4.2. Optimization Landscape Smoothness

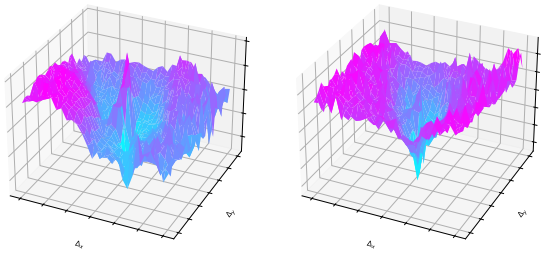
We posit that the stark contrast between open-loop and closed-loop performance lies in the smoothness of the optimization landscape that different models create for the planner. To verify this hypothesis, we designed a series of experiments to visualize this cost function landscape. We first performed one-dimensional action space scan experiments, where we applied perturbations to a single dimension of a successful action sequence and observed the change in cost. To obtain a more comprehensive view, we further conducted a two-dimensional scan: we fixed the first 4 steps of a 5-step action sequence and applied grid-sampled perturbations to the last action along two orthogonal dimensions, plotting the MPC cost corresponding to each perturbed action as a 3D surface map.

As shown in Figure 5, the results provide clear and intuitive evidence. The landscape generated by the Naive Sparse model (left) is highly rugged and noisy, lacking a stable global minimum for convergence. On such a landscape, any sampling-based optimizer is akin to blindly searching in a treacherous, trap-filled terrain, making it extremely easy to fall into local optima. This provides a clear explanation for its failure in closed-loop planning.

In stark contrast, the cost landscape generated by our full DDP-WM model (right) is exceptionally smooth and exhibits a clear, funnel-like macroscopic structure, with a single, deep global minimum at its center. This landscape provides a clear "gravity well" for the optimizer, enabling it to converge stably and efficiently towards the optimal solution.

Table 8. Ablation study on the High-Precision Localization mechanism on the Push-T task (Mask Quality).

Localization Method	IoU ↑	Precision ↑	Recall ↑
Patch-level	0.3446	0.6971	0.5172
High-Precision	0.8935	0.9058	0.9845



(a) *(b)*

Figure 5. **Comparison of MPC cost function landscapes created by different models on the Push-T task.** Both plots show the cost surfaces after 2D perturbation of the action space. **(Left) Naive Sparse model (w/o LRM):** The cost landscape is rugged and noisy, trapping the optimizer in local minima. **(Right) Our DDP-WM model (w/ LRM):** In contrast, the landscape is smooth with a clear, funnel-shaped global minimum, enabling efficient optimization.

5. Conclusion

The core contribution of this paper is an efficient world model paradigm that focuses computation on sparse primary dynamics via localization. We identify that the key to making such a sparse approach succeed in closed-loop planning is to solve the un-smooth optimization landscape problem it introduces. Our proposed Low-Rank Correction Module (LRM) is the architectural solution designed for this purpose, ensuring a plannable landscape by maintaining feature-space consistency. This synergy between sparse prediction and landscape-smoothing correction is responsible for DDP-WM's state-of-the-art results in both performance and speed. Our DDP-WM establishes a promising path for developing efficient, high-fidelity world models.

Limitations. Our current framework assumes a fixed third-person camera, which is the dominant paradigm in tabletop manipulation. Extending to egocentric settings is an active research direction; the core idea is to predict whether changing patches can be characterized by translational transforms (camera motion) versus genuine physical dynamics. Additionally, when drastic global illumination changes cause significant background pixel differences, the dynamic masks expand and the model gracefully degrades to approximately dense prediction, ensuring robustness no worse than the dense baseline.

Acknowledgements

This work is supported by National Key Research and Development Program of China (2024YFE0203100), in part by the National Natural Science Foundation of China under Grant No. 62325605, No. 62572498, No. 62536010 and No. 62322608, in part by the open research fund of Pengcheng Laboratory under Grant 2025KF1B0050, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant NO. 2025A1515011874. This work was supported in part by the Key Development Project of the Artificial Intelligence Institute, Sun Yat-sen University, under Grant 2025RGZN009. We also thank the National Supercomputer Center in Guangzhou for computational support.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Komeili, M., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., Arnaud, S., Gejji, A., Martin, A., Robert Hogan, F., Dugas, D., Bojanowski, P., Khalidov, V., Labatut, P., Massa, F., Szafraniec, M., Krishnakumar, K., Li, Y., Ma, X., Chandar, S., Meier, F., LeCun, Y., Rabbat, M., and Ballas, N. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Bruce, J., Dennis, M. D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., Aytar, Y., Bechtle, S. M. E., Behbahani, F., Chan, S. C., Heess, N., Gonzalez, L., Osindero, S., Ozair, S., Reed, S., Zhang, J., Zolna, K., Clune, J., Freitas, N. D., Singh, S., and Rockt"aschel, T. Genie: Generative interactive environments. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11): 1684–1704, 2024.

Chun, J., Jeong, Y., and Kim, T. Sparse imagination for efficient visual world model planning. In *The Fourteenth*

International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=faxcxKINBC.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Ferraro, S., Mazzaglia, P., Verbelen, T., and Dhoedt, B. Focus: object-centric world models for robotic manipulation. *Frontiers in Neurorobotics*, 19:1585386, 2025.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Garcia, C. E., Prett, D. M., and Morari, M. Model predictive control: Theory and practice—a survey. *Automatica*, 25 (3):335–348, 1989.

Guo, J., Ma, X., Wang, Y., Yang, M., Liu, H., and Li, Q. FlowDreamer: A RGB-D world model with flow-based motion representations for robot manipulation. *IEEE Robotics and Automation Letters*, 2026.

Gupta, A., Tian, S., Zhang, Y., Wu, J., Mart'ın-Mart'ın, R., and Fei-Fei, L. MaskViT: Masked visual pre-training for video prediction. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=QAV2CcLEDh.

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565, 2019.

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

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

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse control tasks through world models. *Nature*, pp. 1–7, 2025.

He, K., Chen, X., Xie, S., Li, Y., Doll'ar, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M. K., Chen, L. Y., Ellis, K., Fagan, P. D., Hejna, J., Itkina, M., Lepert, M., Ma, Y. J., Miller, P. T., Wu, J., Belkhale, S., Dass, S., Ha, H., Jain, A., Lee, A., Lee, Y., Memmel, M., Park, S., Radosavovic, I., Wang, K., Zhan, A., Black, K., Chi, C., Hatch, K. B., Lin, S., Lu, J., Mercat, J., Rehman, A., Sanketi, P. R., Sharma, A., Simpson, C., Vuong, Q., Walke, H. R., Wulfe, B., Xiao, T., Yang, J. H., Yavary, A., Zhao, T. Z., Agia, C., Baijal, R., Castro, M. G., Chen, D., Chen, Q., Chung, T., Drake, J., Foster, E. P., Gao, J., Herrera, D. A., Heo, M., Hsu, K., Hu, J., Jackson, D., Le, C., Li, Y., Lin, X., Ma, Z., Maddukuri, A., Mirchandani, S., Morton, D., Nguyen, T. K., O'Neill, A., Scalise, R., Seale, D., Son, V., Tian, S., Tran, E., Wang, A. E., Wu, Y., Xie, A., Yang, J., Yin, P., Zhang, Y., Bastani, O., Berseth, G., Bohg, J., Goldberg, K., Gupta, A., Gupta, A., Jayaraman, D., Lim, J. J., Malik, J., Martín-Martín, R., Ramamoorthy, S., Sadigh, D., Song, S., Wu, J., Yip, M. C., Zhu, Y., Kollar, T., Levine, S., and Finn, C. DROID: A large-scale in-the-wild robot manipulation dataset. In *RSS 2024 Workshop: Data Generation for Robotics*, 2024. URL https://openreview.net/forum?id=Ml2pTYLNLi.

Kipf, T., van der Pol, E., and Welling, M. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1gax6VtDB.

Liu, Y., Huang, B., Zhu, Z., Tian, H., Gong, M., Yu, Y., and Zhang, K. Learning world models with identifiable factorization. *Advances in Neural Information Processing Systems*, 36:31831–31864, 2023.

Liu, Y., Chen, W., Bai, Y., Liang, X., Li, G., Gao, W., and Lin, L. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Transactions on Mechatronics*, 30(6):7253–7274, 2025. doi: 10.1109/TMECH.2025.3574943.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf,

T. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

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

Micheli, V., Alonso, E., and Fleuret, F. Efficient world models with context-aware tokenization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=BiWIERWBFX.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., and Hsieh, C.-J. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Russell, L., Hu, A., Bertoni, L., Fedoseev, G., Shotton, J., Arani, E., and Corrado, G. Gaia-2: A controllable multi-view generative world model for autonomous driving, 2025. URL https://arxiv.org/abs/2503.20523.

Ryoo, M., Piergiovanni, A., Arnab, A., Dehghani, M., and Angelova, A. TokenLearner: Adaptive space-time tokenization for videos. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12786–12797. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/6a30e32e56fce5cf381895dfe6ca7b6f-Paper.pdf.

Seo, Y., Hafner, D., Liu, H., Liu, F., James, S., Lee, K., and Abbeel, P. Masked world models for visual control. In *Conference on Robot Learning*, pp. 1332–1344. PMLR, 2023.

Terver, B., Yang, T.-Y., Ponce, J., Bardes, A., and LeCun, Y. What drives success in physical planning with joint-embedding predictive world models? *arXiv preprint arXiv:2512.24497*, 2026.

Tong, Z., Song, Y., Wang, J., and Wang, L. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022.

Wang, Y., Lv, K., Huang, R., Song, S., Yang, L., and Huang, G. Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. *Advances in Neural Information Processing Systems*, 33:2432–2444, 2020.

Zhang, K., Li, B., Hauser, K., and Li, Y. AdaptiGraph: Material-adaptive graph-based neural dynamics for robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

Zhang, W., Wang, G., Sun, J., Yuan, Y., and Huang, G. STORM: Efficient stochastic transformer based world models for reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=WxnrX42rnS.

Zhang, W., Jelley, A., McInroe, T., Storkey, A., and Wang, G. Object-centric world models from few-shot annotations for sample-efficient reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qmEyJadwHA.

Zhou, G., Pan, H., Lecun, Y., and Pinto, L. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

Łukasz Kaiser, Babaeizadeh, M., Miłos, P., Osiński, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., and Michalewski, H. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

A. Empirical Verification: The Low-Rank Structure of Background Updates

Our method's core hypothesis is that the context-driven background updates, induced by primary dynamics, possess an inherently low-rank structure. To both (1) empirically validate this central assumption and (2) examine whether our LRM has successfully learned this structure, we conducted a parallel Principal Component Analysis (PCA) on the ground-truth background updates and the updates generated by our LRM.

As depicted in Figure 6, the results provide decisive, twofold evidence for our claims. First, the cumulative explained variance curve for the Ground Truth Updates (right plot) exhibits a sharp rise followed by rapid saturation, strongly verifying our core hypothesis that the true physical dynamics possess a very low intrinsic dimensionality. Second, and most critically, the PCA curve for the updates generated by our LRM (left plot) is strikingly similar to that of the ground truth.

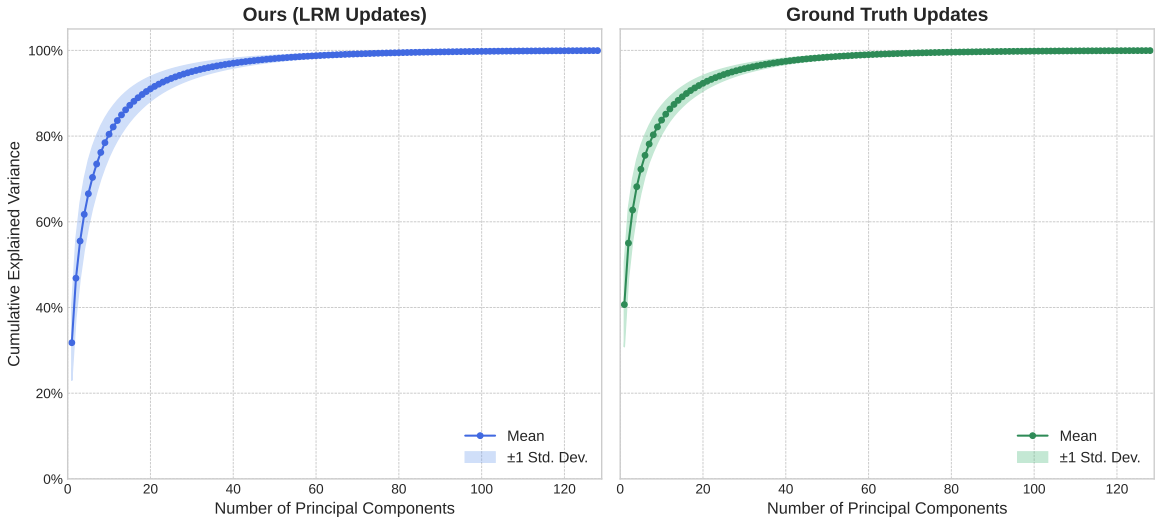


Figure 6. **LRM successfully learns the true low-dimensional structure.** Parallel PCA performed on (Left) the background updates predicted by our LRM and (Right) the ground-truth background updates, on the Push-T task. The remarkable similarity between the two curves indicates that our model has accurately captured and replicated the inherent low-rank nature of the physical dynamics.

This high degree of alignment convincingly demonstrates that our LRM has successfully and faithfully learned to replicate the intrinsic low-dimensional structure present in the true world dynamics. This not only provides the strongest justification for naming our module the "Low-Rank Correction Module" but also fundamentally explains our method's effectiveness and efficiency: it precisely exploits the inherent "simplicity" underlying the physical dynamics.

B. Additional Related Work

B.1. Efficiency Optimization of Transformer Models

With the popularization of Vision Transformers (ViT) (Dosovitskiy et al., 2021), their high computational cost has spurred a large body of research on efficiency optimization. One class of methods involves general token sparsification techniques, which dynamically remove tokens deemed unimportant during inference. These methods either learn token importance through a lightweight network (Rao et al., 2021; Ryoo et al., 2021), or perform pruning based on heuristic rules such as attention scores (Wang et al., 2020). Another class of methods introduces masked autoencoding mechanisms during the training phase, such as MAE (He et al., 2022) and VideoMAE (Tong et al., 2022), which have shown that models can recover full information from partially visible tokens.

These ideas have also been adapted to world models to improve efficiency. For example, MaskViT (Gupta et al., 2023) applies the masked generation paradigm to video prediction and accelerates generation through iterative decoding. Closer to our work is Sparse Imagination (Chun et al., 2026), which randomly drops a portion of image patch tokens during the imagination (rollout) phase of MPC to accelerate computation. Another related work (Micheli et al., 2024) shortens the sequence length by predicting the delta between consecutive states.

However, these general or random sparsification methods do not fully leverage the intrinsic structure of physical dynamics in robotic interaction scenarios. Unlike them, our DDP-WM proposes a structured sparsity scheme grounded in physical insight,

designed to leverage the intrinsic structure of physical dynamics in robotic interaction. The framework explicitly decouples the dynamics into primary dynamics and context-driven background updates, and assigns specialized computational modules to each.

B.2. Sparse and Structured Dynamics Modeling

Leveraging the intrinsic sparsity of dynamics to build more efficient and interpretable models is a natural and attractive research direction. Among these, a mainstream paradigm is object-centric learning. Starting from the pioneering Slot Attention (Locatello et al., 2020), many works have attempted to decompose the world into independent objects and model dynamics at the object level, such as C-SWMs (Kipf et al., 2020), FOCUS (Ferraro et al., 2025), and OC-STORM (Zhang et al., 2026). These methods, by introducing inductive biases, hold the promise of enhancing the model's generalization capabilities and data efficiency. Other works attempt to decouple dynamics from different perspectives; for example, IFactor (Liu et al., 2023) decomposes latent variables based on their interaction with actions and rewards, while FlowDreamer (Guo et al., 2026) explicitly decomposes dynamics into the prediction of 3D scene flow and subsequent rendering.

Our decoupling paradigm offers a more flexible and general alternative to object-centric methods. By automatically separating primary and secondary dynamics at the feature level in a data-driven way, our framework naturally applies to complex scenarios such as deformable bodies and granular materials, where object segmentation is ill-defined.

C. Implementation Details and Hyperparameters

All our models are implemented based on PyTorch. The observation model g_ϕ uses the pre-trained DINOv2-ViT-S/14 model loaded from the HuggingFace Hub, with its weights kept frozen throughout all experiments. All our transition model components (History Fusion, Dynamic Localization, Main Predictor, LRM) are based on a standard ViT architecture. Detailed hyperparameters are provided in Table 9 and Table 10.

Table 9. Shared Training Hyperparameters

Hyperparameter	Value
Optimizer	AdamW
Learning Rate	7e-4
Weight Decay	0.01
Batch Size	64
Training Epochs	100
Learning Rate Scheduler	Constant

Table 10. Model Architecture Hyperparameters

Module	Num Layers	Embed Dim	MLP Ratio
Dynamic Localization Network	6	192	4.0
Sparse Main Predictor	6	404	4.0
History Fusion Module (Cross-Attention)	1	404	N/A
Low-Rank Correction Module (LRM)	1	404	N/A

D. CEM Planning Algorithm

In Section 3.4 of the main text, we briefly introduced the use of CEM for planning. Algorithm 1 provides its detailed, step-by-step procedure. This algorithm is invoked at each decision step to search for the optimal action sequence.

E. Adaptive Sparse Size Mechanism

In Section 3.3.3 of the main text, we mentioned an adaptive sizing strategy. Here we provide its detailed mechanism. This strategy aims to strike an optimal balance between computational efficiency and hardware utilization. We first set

Algorithm 1 CEM for Trajectory Planning

Input: current latent state z_t, goal latent z_g, planning horizon H, number of iterations K, number of samples N, number of elites E.
Initialize Gaussian distribution for action sequence $\mathcal{N}(\mu, \Sigma)$.
for $i = 1$ **to** K **do**
  Sample N action sequences $\{a^{(j)}_{t:t+H-1}\}_{j=1}^{N}$ from $\mathcal{N}(\mu, \Sigma)$.
  For each sequence j, perform rollout using DDP-WM to get future latents $\{\hat{z}^{(j)}_{t+1}, ..., \hat{z}^{(j)}_{t+H}\}$.
  Calculate cost $C_j = \mathcal{L}_{\text{MPC}}(\hat{z}^{(j)}_{t+H}, z_g)$ for each trajectory using the Sparse MPC Cost Mask.
  Select the top E elite sequences with the lowest costs.
  Update μ and Σ of the Gaussian distribution based on the elite set.
end for
Output: The mean of the final action distribution μ as the action for the first step.

a hardware-friendly minimum sparse size, $k_{\text{min}} = 32$. For a given batch, the actual sequence length fed to the primary predictor, k_{batch}, is dynamically set as:

$$k_{\text{batch}} = \max(k_{\text{min}}, \max_{i \in \text{batch}}(k'_i)) \tag{5}$$

where k'_i is the number of changing regions actually detected for the i-th sample in the batch. If the number of changing regions for a sample, k'_i, is less than k_{batch}, we randomly sample $k_{\text{batch}} - k'_i$ patches from its static background regions to pad the input, ensuring that all input sequences in the batch have a regular length of k_{batch}.

This strategy offers a dual advantage: 1) In the vast majority of scenarios where $k'_i < k_{\text{min}}$, this padding ensures the regularity of the input tensors, thereby maximizing the utilization of optimizations in modern parallel computing hardware (like static computation graphs). 2) When encountering complex scenes with drastic changes that result in $k'_i > k_{\text{min}}$, the mechanism automatically and smoothly expands its computational capacity to handle these cases, without clipping important dynamic information due to a fixed sparsity budget.

F. Pixel Error Calculation Details

The pixel error is a metric we use to quantify the fidelity of the predicted visual features.

For any given task, we use a single, frozen pixel decoder to reconstruct images from features. This decoder is typically the one co-trained with the baseline DINO-WM model.

It has a limited capacity. This is a deliberate choice to ensure that the reconstruction quality genuinely reflects the fidelity of the input features, rather than being an artifact of an overly powerful decoder that could overfit and mask underlying feature-level inaccuracies.

To compute the metric, we follow a consistent procedure: we randomly sample 400 episodes from the validation set, using a fixed random seed for fair comparison across all models. For each episode, we perform a 5-step open-loop rollout. The feature map from the final (fifth) predicted step is passed through the frozen decoder. The resulting image is compared pixel-by-pixel against the ground-truth image. We count the number of pixels where the absolute difference exceeds a predefined, task-agnostic threshold. The final "Pixel Error" score is the average of this pixel count over the 400 samples.

G. Environment and Dataset Details

G.1. PointMaze

This environment, introduced by Fu et al. (2020), tasks a force-actuated 2-DoF ball to reach a target goal. Unlike simple kinematic models, the agent's dynamics incorporate realistic physical properties such as velocity, acceleration, and inertia. This requires the world model not merely to infer positions, but to model a dynamical system with physical momentum. For brevity, we refer to this task as Maze in our tables.

G.2. Wall

This is a custom 2D navigation environment that tests the model's spatial reasoning capabilities. The agent must plan within a non-convex space divided by a wall, finding and passing through a narrow door to navigate from one room to another.

G.3. Push-T

This environment, introduced by Chi et al. (2024), is a tabletop manipulation task that places high demands on physical reasoning. Success requires a precise understanding of the complex, contact-rich dynamics between the pusher agent and the T-shaped block, posing a significant challenge to the model's ability to capture the physics of rigid-body interaction.

G.4. Rope Manipulation

Introduced in Zhang et al. (2024) and simulated in Nvidia Flex, this task involves interaction with a deformable body (a soft rope), a notoriously difficult problem in robotic manipulation. The model must learn and predict high-dimensional, non-rigid deformations induced by the robotic arm's actions.

G.5. Granular Manipulation

This environment uses the same simulation setup as Rope Manipulation but elevates the challenge to manipulating a multi-body system of approximately one hundred discrete particles. The model needs to understand the complex collective dynamics and collisional phenomena caused by pushing actions to gather the disordered particles into a desired shape.

H. Training Strategy Details

We adopt a stepwise, decoupled training strategy, primarily for its simplicity and robustness in implementation. A fully end-to-end joint training scheme would necessitate the introduction and careful tuning of multiple hyperparameter weights to balance the different loss functions for each module (e.g., localization, primary prediction, and LRM losses). Fine-tuning these weights is a complex and often brittle process that requires extensive, task-specific hyperparameter sweeps. Our stepwise approach circumvents this challenge entirely by decomposing the joint optimization problem into three simpler, independent sub-tasks. This makes the training process more stable and straightforward to reproduce. First, we train the Dynamic Localization Network on the complete offline dataset. If historical frames are used, the weights of the Historical Information Fusion Module are jointly trained during this stage. Subsequently, using the localization network, we train the Primary Dynamics Predictor. Its loss function is computed only on the foreground regions, aiming to minimize the MSE between the predicted foreground and ground-truth foreground features. Finally, with the first two modules, we train the Low-Rank Correction Module(LRM). Its loss function is computed only on the background regions, aiming to minimize the MSE between the compensated background and ground-truth background features.

I. Additional Ablation Results

I.1. Sensitivity of Threshold τ

The threshold $\tau = 0.5$ is the standard binary classification default; all main results were obtained without any environment-specific tuning. Table 11 shows that performance remains stable within $\tau \in [0.25, 0.75]$, requiring no per-task fine-tuning.

Table 11. Sensitivity analysis of the threshold τ on the Wall task.

τ	Success Rate	Relative Speedup
0.25	98%	Slightly below default
0.50 (default)	98%	$9.1\times$
0.75	90%	Slightly above default

I.2. Long-Horizon Prediction Stability

To address concerns about error accumulation, we extend the open-loop prediction comparison to 20 steps on Push-T (Table 12). DDP-WM's error decreases over longer horizons as the object decelerates after initial contact, while DINO-WM

diverges due to accumulated context crosstalk.

Table 12. Long-horizon open-loop prediction pixel error on Push-T (↓).

Steps	DDP-WM	DINO-WM
5	361	524
10	301	657
20	255	642

J. Qualitative Analysis of Open-Loop Rollouts

This section provides a qualitative comparison of long-term open-loop predictions (5-step rollouts) between our DDP-WM and the dense baseline, DINO-WM, across three representative tasks: Push-T, Granular, and Rope. For a fair comparison, the same pixel decoder, originally co-trained with the DINO-WM baseline, is used to reconstruct images from the latent features of all models.

A visible and consistent pattern emerges across the tasks: predictions from the dense model, DINO-WM, can sometimes degrade over time, becoming blurry, distorted, and losing critical physical details. In contrast, our DDP-WM is more inclined to generate sharp, physically coherent, and high-fidelity predictions. We argue that this difference in prediction quality helps to explain the success of our method.

J.1. Push-T Task

Figure 7 shows some samples on the Push-T task. In DINO-WM's predictions, visual artifacts can be observed. For instance, the pushed T-block might exhibit feathering at its edges or soft-body-like distortions, which can compromise its true rigid-body characteristics. In contrast, our DDP-WM consistently maintains the block's sharp boundaries and correct rotational pose.

J.2. Granular Manipulation Task

In the more challenging Granular task (Figure 8), both models can sometimes misinterpret the concept of a multi-body system, incorrectly predicting the discrete particles as a continuous, amorphous blue "gel" or "cloud." However, our DDP-WM successfully preserves the discrete nature of the system in most cases, making predictions about the collective dynamics of the particles that are highly similar to the ground truth.

J.3. Rope Manipulation Task

For the prediction of the deformable body (rope), as shown in Figure 9, our DDP-WM consistently predicts the rope as a clear, continuous curve, accurately modeling its bending and twisting deformations.

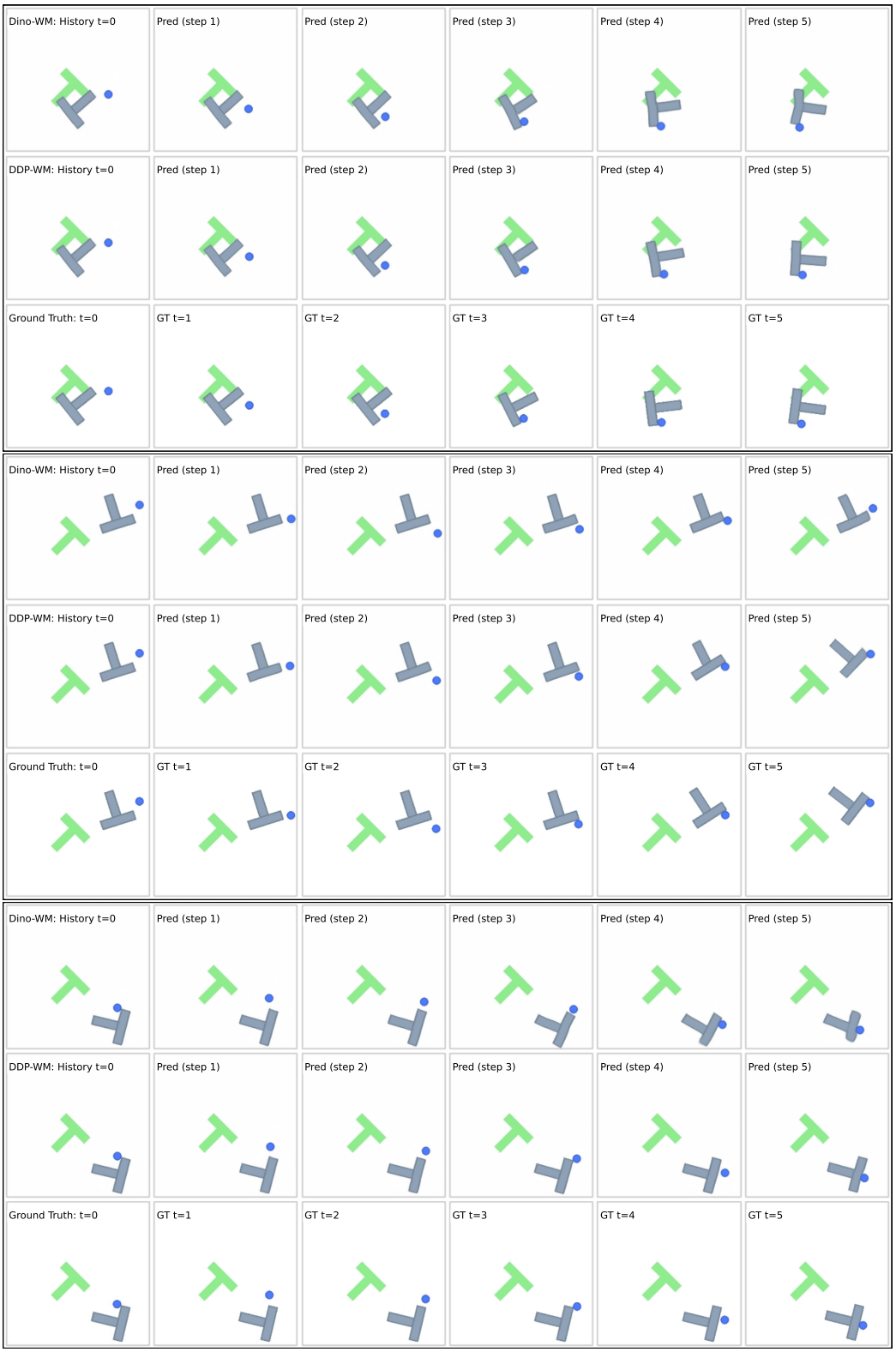


Figure 7. Open-loop rollouts on the Push-T task. From top to bottom: Sample 1, Sample 2, and Sample 3. For each sample, the rows show the results from DINO-WM (top), DDP-WM (Ours, middle), and the ground truth (bottom).

Figure 8. Open-loop rollouts on the Granular task. From top to bottom: Sample 1, Sample 2, and Sample 3. Similar to the previous figure, each sample shows results from DINO-WM (top row), DDP-WM (Ours, middle row), and the ground truth (bottom row).

Figure 9. Open-loop rollouts on the Rope task. From top to bottom: Sample 1, Sample 2, and Sample 3. For each sample, the rows depict the rollouts from DINO-WM (top), DDP-WM (Ours, middle), and the ground truth (bottom).