DDP-WM: Disentangled Dynamics Prediction for Efficient World Models

Shicheng Yin*, Kaixuan Yin*, Weixing Chen, Yang Liu†, Guanbin Li, Liang Lin

Sun Yat-sen University | X-Era AI Lab

ICML 2026

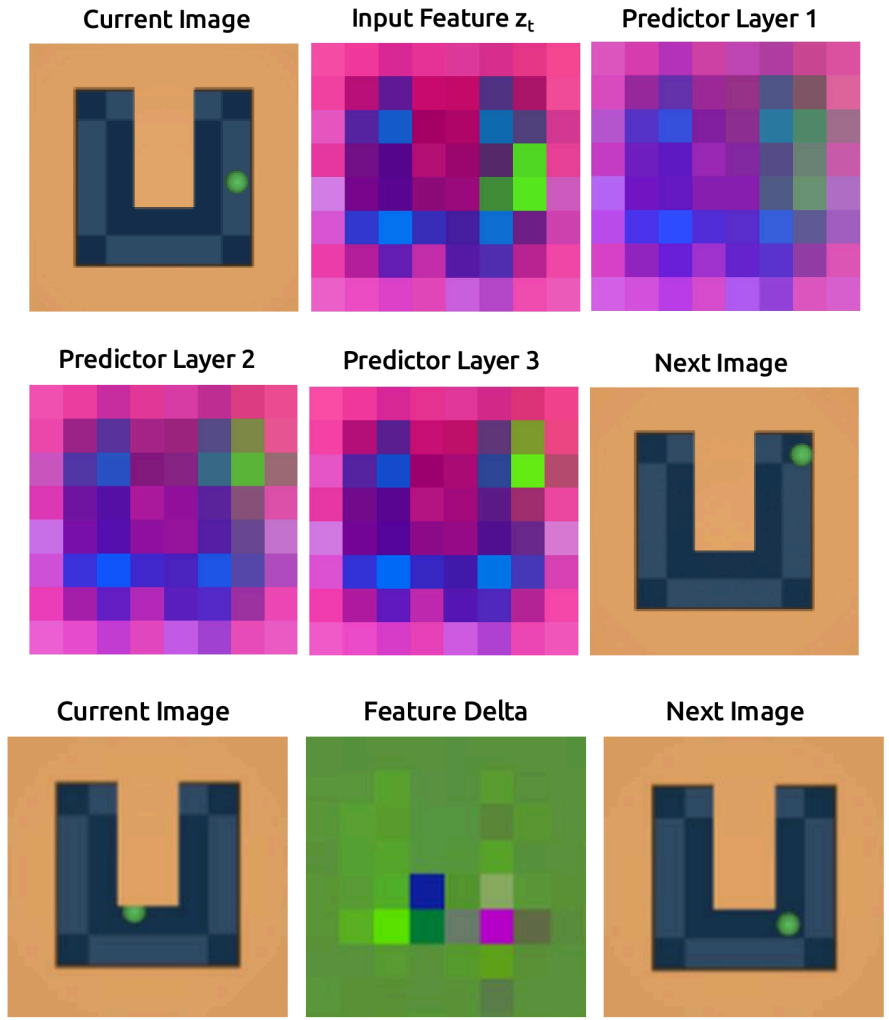
Motivation

World models allow robots to plan by imagining the future consequences of their actions. Current state-of-the-art approaches build these models using dense Transformers that process every visual token with equal computational cost, regardless of whether the corresponding region is actually changing. In most physical interactions, however, only a small fraction of the scene undergoes meaningful change — the rest is static background receiving redundant computation.

This inefficiency is severe: on a representative manipulation task (Push-T), the best existing dense model requires approximately two minutes per MPC planning decision on a single GPU, making real-time deployment infeasible.

We propose DDP-WM, which exploits this inherent sparsity to achieve a **9× inference speedup** while simultaneously improving planning success from **90% to 98%**.

Key Insight: Dynamics Sparsity is Preserved in Feature Space



Our starting observation is that physical dynamics are sparse: objects move, but tables and walls stay still. The critical question is whether this sparsity is preserved in the DINOv2 feature space we operate on.

(a) PCA projections of features at each predictor layer show that background regions remain nearly identical — the self-attention applied to them essentially computes an identity transformation.

(b) The inter-frame feature difference is dominated by green (near-zero change), with only a small localized region showing significant variation.

Conclusion: We can directly exploit this sparsity in feature space — focus computation on changing regions and skip the rest.

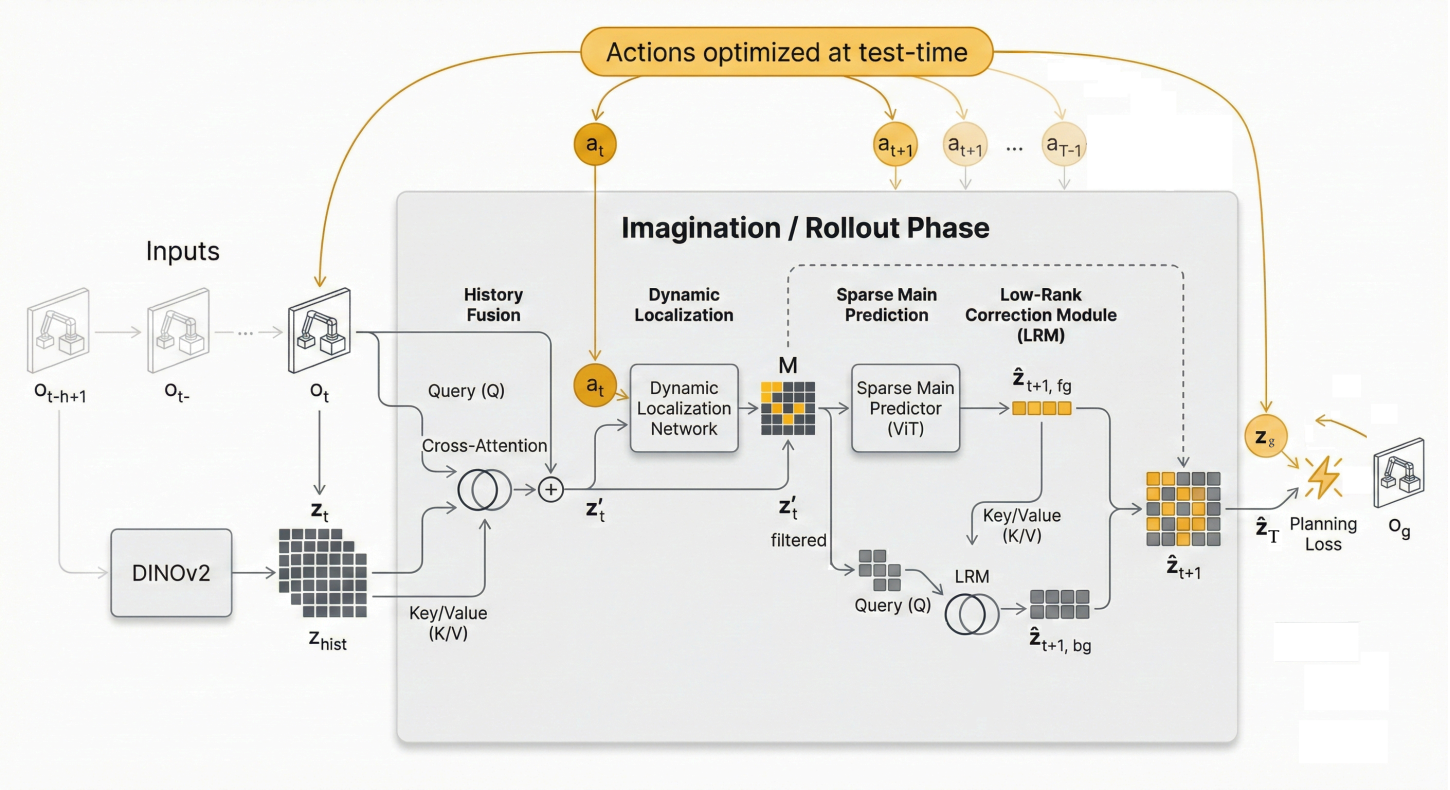
DDP-WM: Decoupled Dynamics Prediction Framework

1. History Fusion: Cross-attention injects temporal context from past frames.

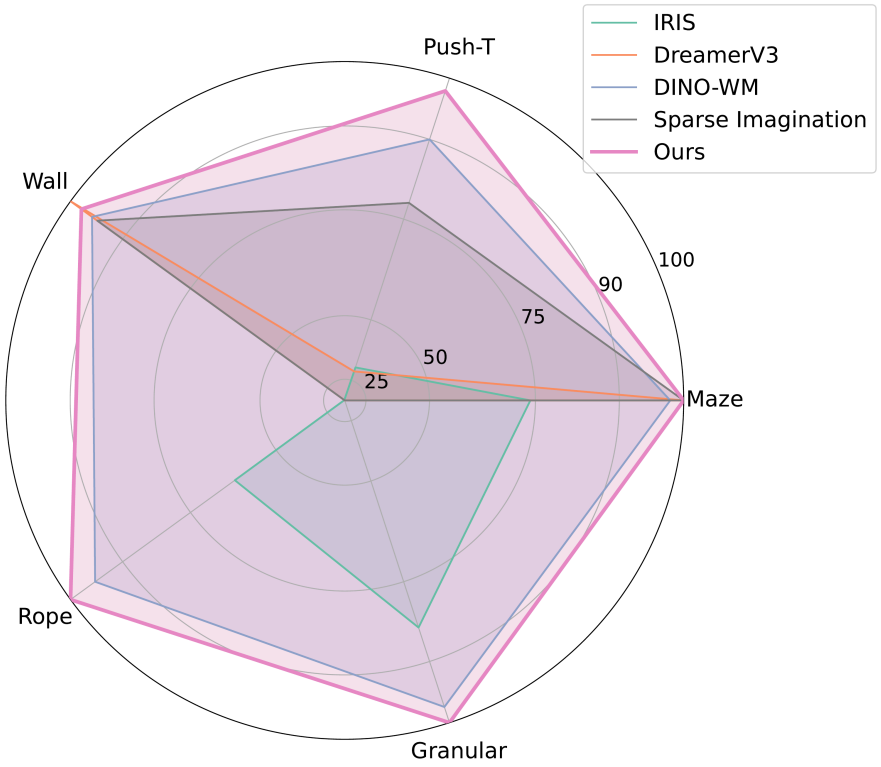
2. Dynamic Localization: Lightweight ViT predicts which regions will change. Produces sparse mask.

3. Sparse Primary Prediction: Powerful Transformer on masked tokens only (~10–20%). Quadratic cost drops 1–2 orders of magnitude.

4. Low-Rank Correction: Single cross-attention updates background by querying foreground. Minimal cost.



Results: Performance and Efficiency



DDP-WM achieves simultaneous improvements in both planning performance and computational efficiency across all five environments.

Metric (Push-T)	DINO-WM	Ours	Speedup
FLOPs	23G	2.5G	**9.2×**
MPC Latency	~120s	~16s	**7.5×**
Throughput	170 s/s	1563 s/s	**9.2×**

This is not a tradeoff — the decoupled architecture produces higher-quality predictions by eliminating noise from redundant background computation.

Real-World Validation: DROID Dataset

Evaluated on **DROID** real-world robot dataset with DINOv3 ViT-L/16 encoder.

Method	Action Score	Speedup
DINO-WM (ViT-S)	39.4 ± 2.1	—
V-JEPA 2-AC	42.9 ± 2.5	—
JEPA-WMs dense	46.5 ± 0.4	1×
DDP-WM (Ours)	**47.9 ± 0.6**	**4.26×**

The dynamic localization network **correctly classifies** illumination changes and background human movement as static — it does not confuse appearance variation with physical dynamics.

When truly global changes occur, masks expand automatically and the model gracefully degrades to dense prediction, ensuring robustness no worse than the baseline.

Conclusion

DDP-WM demonstrates that the inherent sparsity of physical dynamics can be directly exploited to build world models that are both faster and more accurate. By decomposing scene evolution into sparse primary dynamics and low-rank background updates, and allocating computation accordingly, we achieve a 9× inference speedup on Push-T while improving planning success from 90% to 98%. The approach generalizes to real-world data (DROID, 4.26× speedup) and handles diverse physics including rigid bodies, deformable objects, and multi-body systems.

The key architectural insight is that naive sparse prediction fails in closed-loop planning because it creates discontinuous optimization landscapes. Our Low-Rank Correction Module resolves this by maintaining feature-space consistency, providing the smooth cost surface that sampling-based planners require.

Code: https://hcplab-sysu.github.io/DDP-WM/

Thank you.