LpWM: A Case for Sparse Representations in World Models

Yilun Kuang[1] · Yash Dagade[2] · Quentin Le Lidec[1] · Lucas Maes[3] · Randall Balestriero[4] · Yann LeCun[1]
[1]New York University, AMI Labs · [2]Duke University · [3]Mila, AMI Labs · [4]Brown University, AMI Labs

github.com/YilunKuang/lpworldmodel

Dense Gaussian for anti-collapse is a choice, not a requirement. Rectified Laplace yields sparse, simpler, more interpretable dynamics.

+57% planning success over dense LeWM on PushT, at intermediate predictor capacity

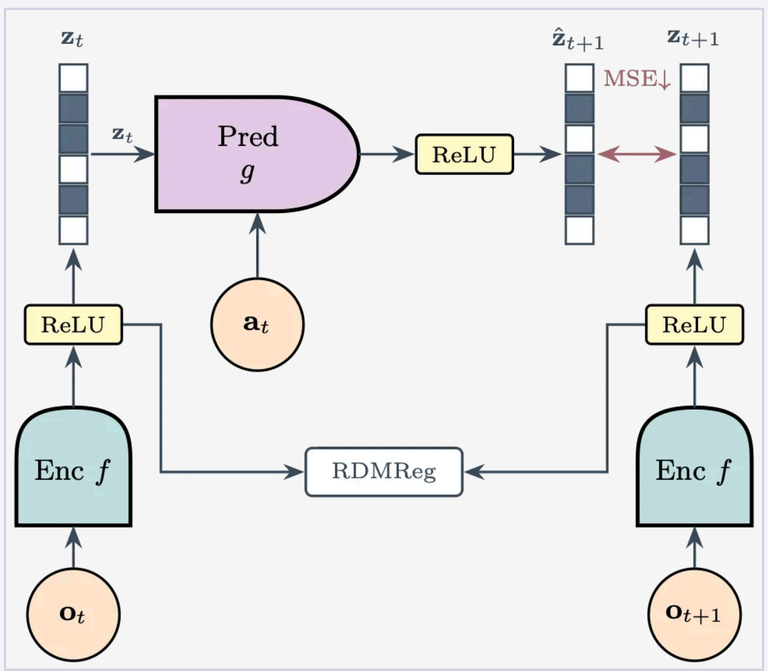
94–99% of the discrete dynamics regime decodable from the binary support alone

30–65% active coordinates in the learned code, across every predictor and width

0.05 → 0.61 support–contact correlation once a temporal-Jaccard prior is added

01 LpWorldModel with RDMReg

JEPAs prevent collapse with maximum-entropy targets like the isotropic Gaussian, so every coordinate is dense and nonzero. LpWM matches a **Rectified Generalized Gaussian** distribution instead.



Architecture. Encoder $z_t = f_\theta(o_t)$ maps observations to sparse latents via a terminal (Rep)ReLU. Predictor $\hat{z}_{t+1} = g_\psi(z_t, a_t)$ rolls them forward with rectifications.

RECTIFIED DISTRIBUTION MATCHING REGULARIZATION (RDMREG)

$$\min_{\theta,\psi} \|\hat{z}_{t+1} - z_{t+1}\|_2 + \lambda_{\text{RDMReg}} \cdot R(z_{t+1})$$

$$R(z) = \mathbb{E}_{c\sim\text{Unif}(\mathbb{S}^{d-1})}[\, W_2(\, \mathbb{P}_{c^\top z} \,\|\, \mathbb{P}_{c^\top y}\,)\,]$$

$$y \sim \prod_{i=1\ldots d} \text{ReLU}(\, \mathcal{GN}_p(\mu, \sigma)\,)$$

Random projection vectors c is uniform on the ℓ_2 unit sphere. At μ=0, σ=√½, p=1, the target is Rectified Laplace. At μ=0, σ=1, p=2 with no ReLU, it's the isotropic Gaussian used in LeWM.

WHY SPARSITY SHOULD HELP

By sufficiently fine quantization of a compact state space, Lipschitz nonlinear dynamics can be approximated arbitrarily well by dynamics that are exactly linear in a finite one-hot latent representation, with rollout error vanishing as the code dimension grows.

For $x_{t+1} = f(x_t, a_t)$ on compact $\mathcal{X}$ with $\|f(x,a) - f(y,a)\| \le L\|x-y\|$, for every $\varepsilon > 0$ there is a one-hot encoder $E_\varepsilon: \mathcal{X} \to \mathbb{R}^N$, decoder D_ε, and action-conditioned $P_\varepsilon(a) \in \{0,1\}^{N\times N}$ with

$$z_{t+1} = P_\varepsilon(a_t)\, z_t \quad \text{(exactly linear)}$$

$$\sup_{x,a} \|f(x,a) - D_\varepsilon(P_\varepsilon(a)\, E_\varepsilon(x))\| \le (L+1)\varepsilon$$

On $\mathcal{X} = [0,1]^d$ with $N = n^d$ cells, $\varepsilon_N = (\sqrt{d}/2)\cdot N^{-1/d}$, so the one-step error is $O(N^{-1/d})$ and over horizon H

$$\|x_H - \hat{x}_H\| \le \varepsilon_N \textstyle\sum_{k=0\ldots H} L^k \to 0 \quad \text{as } N \to \infty$$

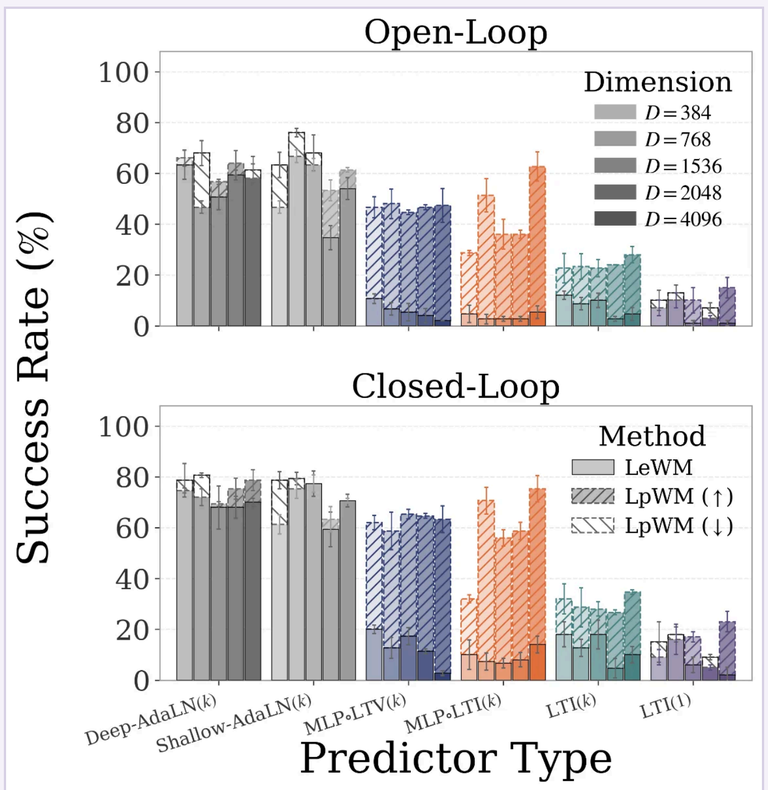
PREDICTOR COMPLEXITY LADDER

Predictor	Functional form
Deep-AdaLN(k)	$\hat{z}_{t+1} = \sigma(\text{AdaLN}^{(6)}(z_{t-k+1:t}, a_t))$
Shallow-AdaLN(k)	$\hat{z}_{t+1} = \sigma(\text{AdaLN}^{(1)}(z_{t-k+1:t}, a_t))$
MLP∘LTV(k)	$\hat{z}_{t+1} = \sigma(W\ \text{ReLU}(\Sigma\ A_i(z_t) z_{t-i} + B(z_t) a_t))$
MLP∘LTI(k)	$\hat{z}_{t+1} = \sigma(W\ \text{ReLU}(\Sigma\ A_i\ z_{t-i} + B\ a_t))$
LTI(k)	$\hat{z}_{t+1} = \sigma(\Sigma\ A_i\ z_{t-i} + B\ a_t)$
LTI(1)	$\hat{z}_{t+1} = \sigma(A\ z_t + B\ a_t)$

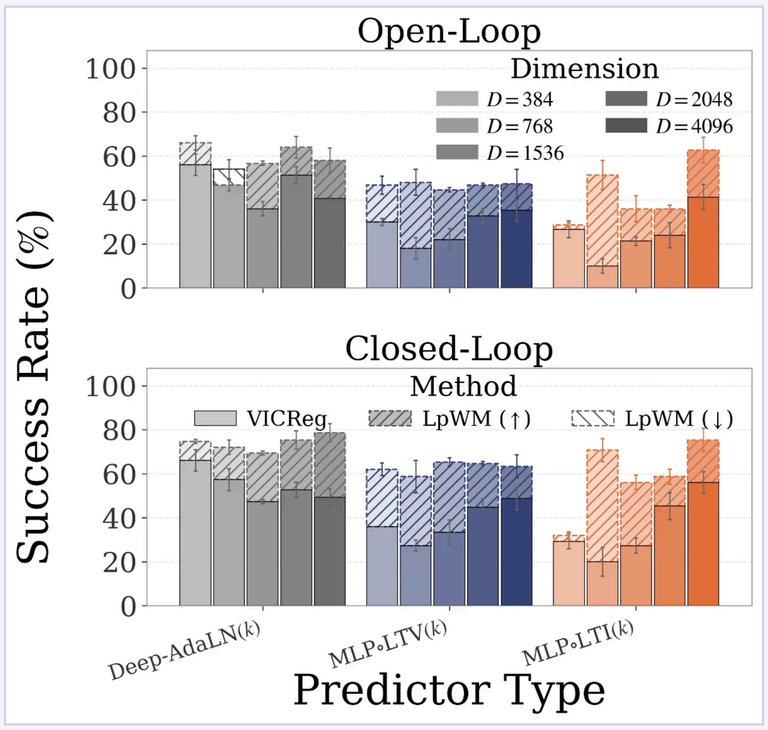
Shaded rows: the intermediate-capacity predictors. σ = identity for LeWM, RepReLU for LpWM, where $\text{RepReLU}(x) = \text{sg}(\text{ReLU}(x)) + \text{GeLU}(x) - \text{sg}(\text{GeLU}(x))$ with sg the stop-gradient — ReLU in forward, GeLU gradients backward. It's also fine to just use ReLU instead of RepReLU.

02 Sparsity simplifies dynamics

We compare LeWM and LpWM with controlled encoders and varying predictor types and feature dimensions. Both methods are well tuned using with maximum update parameterization (muP). On PushT, predictors that fail over dense codes succeed over sparse ones.



LpWM vs. LeWM, PushT. Arrows are success-rate differences. LpWM gains **24–57%** with MLP∘LTI(k), **36–45%** with MLP∘LTV(k), **11–23%** with LTI(k) — and draws level with the high-capacity DiT predictors, where complexity saturates. LTI(1) fails for both.



Beyond Gaussian targets. Against a VICReg-regularized dense JEPA, LpWM wins across predictor families — including Deep-AdaLN(k). We hypothesize that second-order constraints alone in VICReg leave the target distribution underspecified.

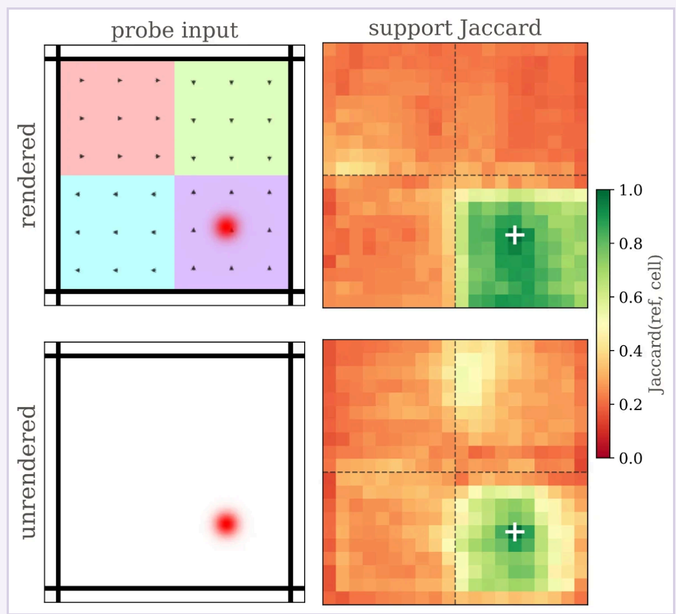
TWO CONDITIONS ON THE GAIN

The dynamics must be hard enough: on **Wall**, even LTI(1) nears 100% success for both methods, already too linear in latent space to separate sparse from dense.

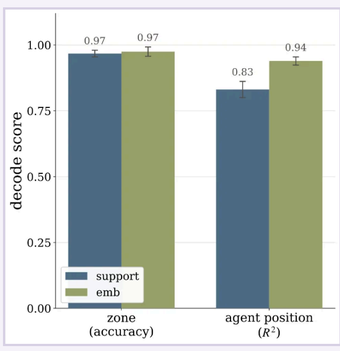
And the code must be **genuinely sparse**: 30–65% active coordinates at every predictor and width, against 100% for dense LeWM.

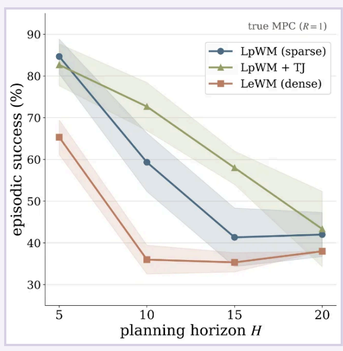
03 Sparse code is interpretable

Which coordinates are active encodes the **discrete dynamical regime.** How large they are encodes the **continuous state within** it.

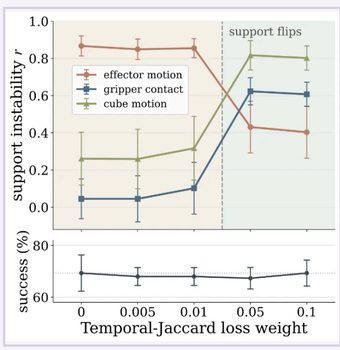


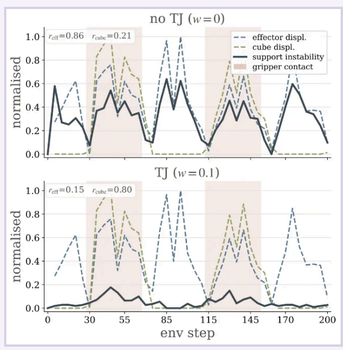
Support recovers the partition. Jaccard overlap with a reference cell stays high inside its force-field zone and drops across the boundary — even when the zones carry no visual cue (bottom left). The regime is read off the dynamics, not the pixels.





Probes and payoff. Left: the binary support decodes zone identity as well as the full embedding, while position decodes better from magnitudes. Right: LpWM beats LeWM at every planning horizon, and the gap widens with H.





Contact-rich case, OGBench-Cube. With vanilla LpWM, support instability is a motion detector: r ≈ 0.87 with effector motion, 0.05 with physical contact. A temporal-Jaccard slowness prior makes it track cube motion (0.26→0.80) and gripper contact (0.05→0.61) instead, at no cost in planning success. Sparsity alone may not ensure that support transitions correspond to semantically meaningful dynamical events, and without temporal structure the support can instead behave largely as a motion detector. This is a limitation and we intend to investigate more in future work.

TAKEAWAY

Sparse representations can reduce the predictor complexity for control while leading to more interpretable dynamics.

yilun.kuang@nyu.edu · yann.lecun@nyu.edu

Supported by AFOSR FA95502310139, NSF Award 1922658, AMI Labs and NYU HPC.

LPWORLDMODEL