Explicit Vision Signals as Auxiliary Supervision for World Model Fine-Tuning

Kevin Hopkins

kevinhop@usc.edu

April 2026

Abstract

Foundation models for robotics increasingly mediate between raw video and action selection through learned latent representations, raising an open question about whether engineered computer-vision signals (depth, point tracks, masks, geometric descriptors) should be injected as auxiliary supervision or instead absorbed implicitly through scale. We probe this question with a controlled fine-tuning ablation on LeWorldModel, a compact JEPA-style latent world model, over curated BridgeData V2 manipulation footage. Seven auxiliary supervision conditions are evaluated against a no-auxiliary baseline using identical optimizer, model, and data, varying only the auxiliary head. On a 13-episode curated pilot, depth-based supervision improves next-latent prediction MSE by 11.5–14.5% relative to baseline. On a 100-episode full-scale run, all auxiliary signals collapse to within $\pm 1.3\%$ of baseline and most regress slightly. We argue this sign flip reflects auxiliary heads acting as inductive bias that compresses the search space at low data scale and becomes redundant or interfering at higher scale. We further document a CLS-bottleneck failure mode for dense local signals (point tracks). The findings support a training-time view of explicit 3D vision in robot foundation models.

1 Introduction

The shape of training pipelines for robot foundation models is currently contested. One school argues that web-scale video pretraining will absorb the relevant inductive biases (geometry, contact, object permanence) implicitly, and that engineered computer-vision pipelines belong in the data-curation step rather than the loss function [10, 13]. A second school continues to inject explicit signals (subgoal images, depth, hand pose, object masks) as auxiliary objectives or conditioning channels and reports gains on real-world manipulation benchmarks [7, 9, 11]. A third position treats explicit 3D representations as evaluation tools or rendering layers separate from the policy entirely [12].

Empirical evidence on which kinds of explicit visual supervision actually help, under what data and architectural regimes, is scarce. The available signal at the foundation-model scale is filtered through proprietary training stacks and cannot be cleanly attributed to any single design choice. Smaller open testbeds offer the opposite tradeoff: limited generality but full attribution.

This paper contributes a small-scale empirical answer on a transparent open-source testbed. We extend the recently released LeWorldModel [4] with seven auxiliary supervision heads spanning depth (full-frame, masked, weighted), point tracks (uniform and foreground-biased), object masks, 3D centroids, and a 10-dimensional shape descriptor. All conditions share the same base model, dataset, and optimizer. We run the same conditions at two data scales and observe a clean sign flip: signals that improved next-latent prediction MSE by up to 14.5% on a curated 13-episode pilot move to within $\pm 1.3\%$ of baseline at 100 episodes.

We interpret this as evidence that explicit auxiliary supervision functions as inductive bias whose value is governed by data scale. We additionally diagnose a CLS-bottleneck limitation that prevents dense per-pixel or per-point targets from being reconstructed through the global latent.

2 Related Work

JEPA-family world models. Joint Embedding Predictive Architectures predict in a learned representation space rather than over pixels, motivated by the observation that high-frequency texture is wasteful capacity for downstream control [1, 2, 3]. LeWorldModel inherits this design: a frozen-architecture ViT-Tiny encoder produces a CLS token, and a transformer predictor learns next-latent dynamics conditioned on actions [4]. Recent JEPA variants extend the framework with object-centric masking [5] or vision-language joint embedding [6], but the underlying objective (predict next embedding under action) is shared.

Auxiliary CV heads in robot foundation models. A separate strand of work injects classical CV outputs directly into the training loss. Pi0 and its successors condition action heads on subgoal images and per-episode metadata, with documented compositional gains [7]. GR00T uses an explicit two-system architecture coupling a VLM reasoner to a diffusion action head, with auxiliary supervision over egocentric human video [9]. VideoManip recovers hand-pose, contact, and object meshes from web video and uses them as supervision signals for a diffusion policy [11]. DreamDojo argues for the opposite extreme, that 44,000 hours of egocentric pretraining is sufficient to subsume hand-engineered signals [10].

Scaling and the implicit-explicit boundary. Several recent results suggest that explicit auxiliary signals are most useful at low data scale and get absorbed by larger pretraining sets. Pi0's data-quality findings (top-20% subset outperforming algorithmic tweaks) and the broader pattern that subgoal-image conditioning loses relative strength as the data mix grows are consistent with this story [8]. The present work tests the same hypothesis in a controlled small-scale setting where the only variable is auxiliary supervision and dataset size.

3 Method

3.1 Dataset and curation

We use BridgeData V2 [14], a corpus of 7-DOF tabletop manipulation episodes recorded from a fixed overhead camera with paired action and gripper-state observations. Raw episodes are filtered with a weighted scoring function that prefers *put-X-on-Y* task descriptions, episode lengths between 15 and 40 frames, total point-track displacement above 60 pixels, object-mask coverage between 2% and 25% with low temporal variance, and depth dynamic range above 2.0. The top 300 episodes by score were retained, of which 100 were used for the full-scale set (2,920 frames total) and 13 hand-selected episodes (334 frames) form a curated pilot set. The pilot set was selected for visually clean manipulation and high-quality auxiliary signal extraction.

3.2 Base model

The base model is LeWorldModel-Cube [4], a JEPA-style world model with $\sim$15M parameters total. The encoder is ViT-Tiny ($\tilde{}$7.6M params, 12 layers, 192-dim hidden, 3 heads, 14$\times$14 patches at 224$\times$224 resolution) producing a 192-dimensional CLS token per frame. A 6-layer transformer predictor with AdaLN-zero modulation [18] consumes a 3-frame context window of CLS embeddings plus action tokens (7-DOF) and predicts the next CLS embedding. Predictor hidden dim 192, MLP dim 2048, 16 attention heads, dropout

0.1. The architecture remains unchanged across all conditions; auxiliary heads attach as additional decoders from the same CLS embedding.

3.3 Auxiliary supervision conditions

Seven conditions are evaluated against an A-baseline (no auxiliary head). All auxiliary signals are extracted offline and packed into HDF5 files alongside RGB pixels, actions, and observations.

- **B (Depth)**: Full-frame depth predicted as a 56×56 map via a small ConvTranspose decoder (∼400K extra parameters). Targets generated by Video-Depth-Anything (ViTS encoder) [15]. Loss is scale-shift-invariant L1.
- **C-family (Masked Depth)**: Same architecture as B but supervised only within an object mask region. C1 uses tight masks; C2 and C3 dilate the mask by 8 and 20 pixels respectively. C4 uses full-frame depth with a 5× multiplicative weight inside the object region. Masks are generated with SAM 3 [17] prompted with the text caption "object being grabbed by robotic arm."
- **D (Tracks)**: Predict 400 per-point 2D displacements from the CLS token through a flat MLP (∼135K extra parameters). Track targets are computed with CoTracker3 [16] in offline mode, with a 20×20 query grid. A foreground-biased variant places approximately 60% of query points inside the discovered object region, raising the average count of object-supported visible tracks per frame from ∼5 to ∼47.
- **E (Depth + Tracks)**: B and D auxiliary heads enabled simultaneously.
- **F (Centroid)**: Predict a single 3D centroid $[x, y, z]$ from the CLS token through a 3-layer MLP (∼37K extra parameters). Targets computed by backprojecting masked depth and averaging.
- **G (Shape Descriptor)**: Predict a 10-dimensional shape vector summarizing the object point cloud (centroid xyz, principal extents from PCA, spread ratios, depth range, log-count visibility proxy). Same MLP capacity as F.

3.4 Training setup

All conditions share identical optimizer and schedule: AdamW with $lr = 5 \times 10^{-5}$, weight decay 10^{-3}, batch size 16, 20 epochs, bf16 mixed precision, gradient clip 1.0, with a linear warmup followed by cosine annealing. Auxiliary loss weight is fixed at 0.1. The total loss is $\mathcal{L} = \mathcal{L}_{\text{pred}} + 0.09\,\mathcal{L}_{\text{sigreg}} + \sum_h 0.1\,\mathcal{L}_h$ where $\mathcal{L}_{\text{pred}}$ is MSE between predicted and target next-frame CLS embeddings, $\mathcal{L}_{\text{sigreg}}$ is the SIGReg regularizer from the LeWM base recipe, and $\mathcal{L}_h$ is the auxiliary head loss. Each condition is repeated across 3–10 random seeds (more seeds for early conditions, fewer for later additions). Hardware was a single consumer GPU.

4 Results

4.1 Pilot scale: depth-conditioned supervision improves next-latent prediction

On the 13-episode curated pilot, all auxiliary signals except F (centroid) deliver measurable improvement in next-latent prediction MSE. Depth-based conditions dominate: B (full-frame depth) at -11.5%, C4 (weighted full-frame depth) at -14.5%, and E (depth + tracks) at -14.0%. The track-only condition D shows a modest -4.9%. The compressed shape descriptor G achieves -6.1%. The centroid condition F is statistically indistinguishable from baseline.

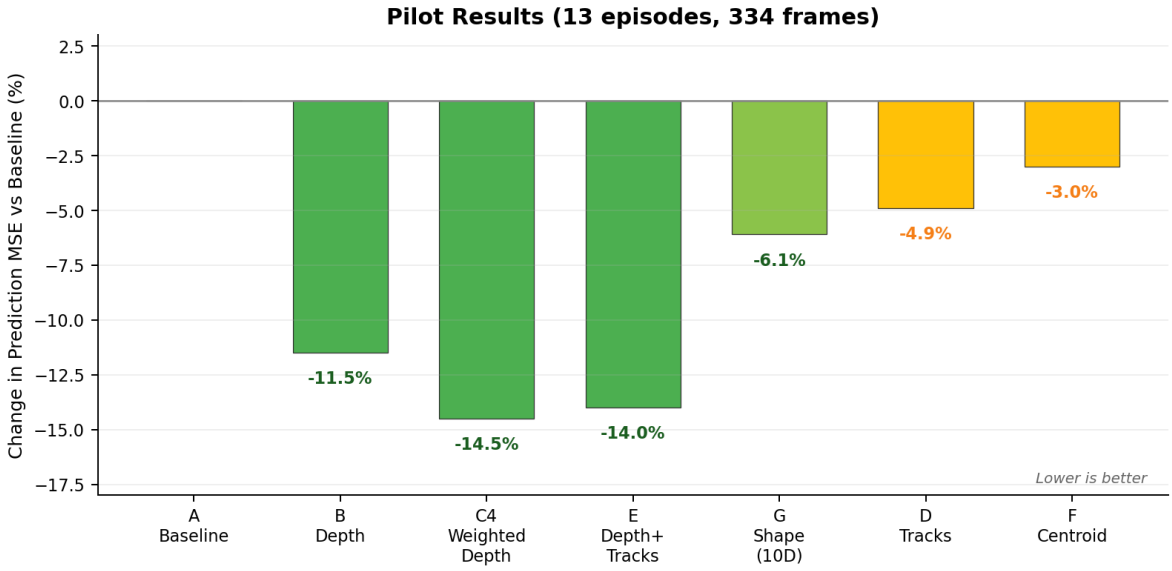

Figure 1: Per-condition change in next-latent prediction MSE relative to baseline on the 13-episode curated pilot. Lower (more negative) is better. Depth and depth-conditioned signals deliver the largest improvements; track-only and centroid-only signals show small or negligible effects.

These numbers have meaningful seed variance (overlapping error bars across most pairs of conditions), but the depth-conditioned cluster is consistently separated from the track-only and centroid cluster. We also observe that the addition of tracks to depth (E) does not meaningfully exceed depth alone (B), consistent with the conjecture that the track signal is not productively integrated through the CLS bottleneck (Section 4.4).

4.2 Mask family: tighter masking degrades depth supervision monotonically

A controlled sweep over masking strategies (Figure 2) reveals a clean monotonic relationship: as the depth supervision region shrinks, the auxiliary signal becomes less useful. Full-frame depth (B) achieves -10.5% and the weighted full-frame variant (C4) sits slightly better at -10.9%. Tight-mask-only (C1) drops to -7.5%, then C2 (8px dilation) at -5.8% and C3 (20px dilation) at -2.3%. The interpretation is that scale-shift-invariant depth normalization becomes unstable when computed over a small spatial support: with object masks covering only 2–5% of pixels, the per-sample scale and shift estimates become noisy, and the supervisory signal degrades. This is a methodologically useful negative result. If a depth-supervision pipeline uses scale-shift-invariant loss, it should not also apply tight spatial masking.

4.3 Full scale: the sign flips

The most consequential finding of the study is shown in Figure 3. The three auxiliary conditions repeated at full scale (B, D, E) all lose their pilot-scale benefit. Depth (B) moves from -11.5% to $+1.0\%$. Depth + tracks (E) moves from -14.0% to $+1.3\%$. Tracks alone (D) moves from -4.9% to $+0.2\%$. The absolute baseline MSE drops by approximately $4\times$ from pilot to full scale, indicating that the model has substantially more to learn from the larger dataset and is in fact learning it.

The sign flip is not driven by reduced statistical power. The full-scale runs use the same number of seeds (3) and the absolute MSE differences across seeds are smaller in absolute terms because the baseline error itself has dropped. The auxiliary conditions are reliably at or slightly above baseline, not noisily below it.

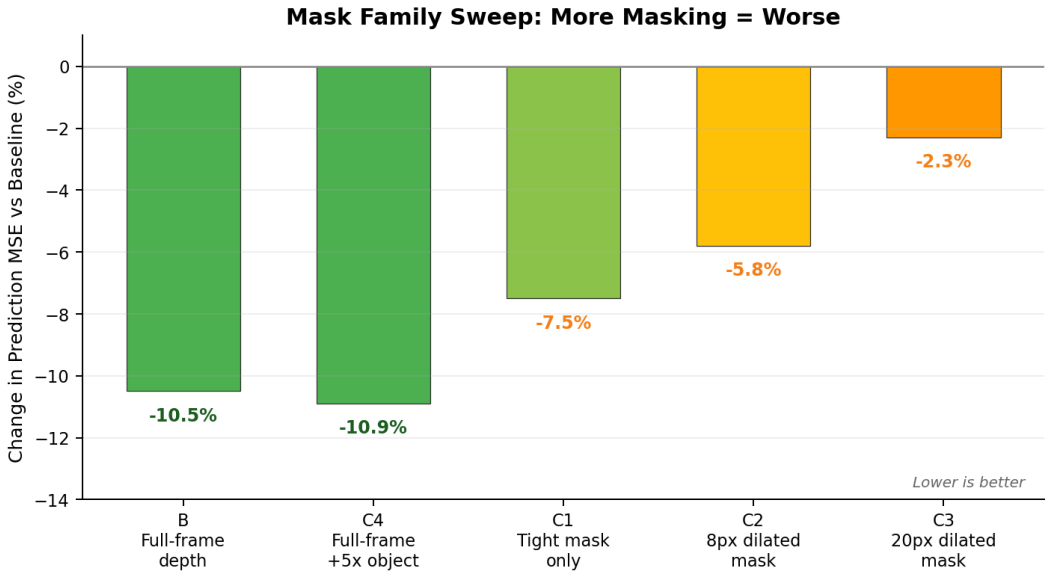

Figure 2: Mask family sweep. Restricting the depth supervision region monotonically degrades the auxiliary signal's value, from full-frame (-10.5%, -10.9% weighted) to tight-mask-only (-7.5%) to 8px-dilated (-5.8%) to 20px-dilated (-2.3%).

We interpret this pattern as evidence that auxiliary supervision in this setup acts as a scale-dependent inductive bias. On a small curated dataset, the auxiliary head provides a strong gradient signal that compresses the model's search space toward a useful prior (e.g., representations that encode depth structure). On a larger and more diverse dataset, the same prior is available implicitly through the next-latent prediction objective alone, and the auxiliary loss either becomes redundant or actively interferes through gradient conflict. The sign flip is clean enough that the data scale is the only varying factor.

4.4 The CLS bottleneck and dense local signals

The track condition (D) provides a useful diagnostic. The first track-supervision run used uniform 20×20 grid initialization which placed only $\sim$5 visible tracks on the manipulated object per frame. We hypothesized this was a coverage problem and built a foreground-biased variant that achieved $\sim$47 visible object tracks per frame. Visual audit confirmed the tracks landed on the object with high fidelity. The supervision still produced no improvement at full scale and only -4.9% at pilot scale, well behind depth.

The mismatch between signal quality and learning value points to an architectural failure mode rather than a data problem. The track head must regress 800 scalars (400 points $\times$ 2 coordinates) from a single 192-dimensional global CLS vector through a flat MLP. Each track is a local quantity that depends on a specific image region; the CLS token compresses the whole frame into a global summary; and the MLP has no spatial structure with which to recover the per-point information. Depth, which is dominated by global scene properties (near vs. far), can be recovered by such a head; tracks cannot.

This is the same bottleneck noted in concurrent work that achieves better results with track-supervised models by treating each track as its own token with local DINOv3 features and explicit occlusion handling [19]. The takeaway is that extraction quality and supervision strength alone do not determine the value of an auxiliary signal. The downstream architecture must be capable of using the signal.

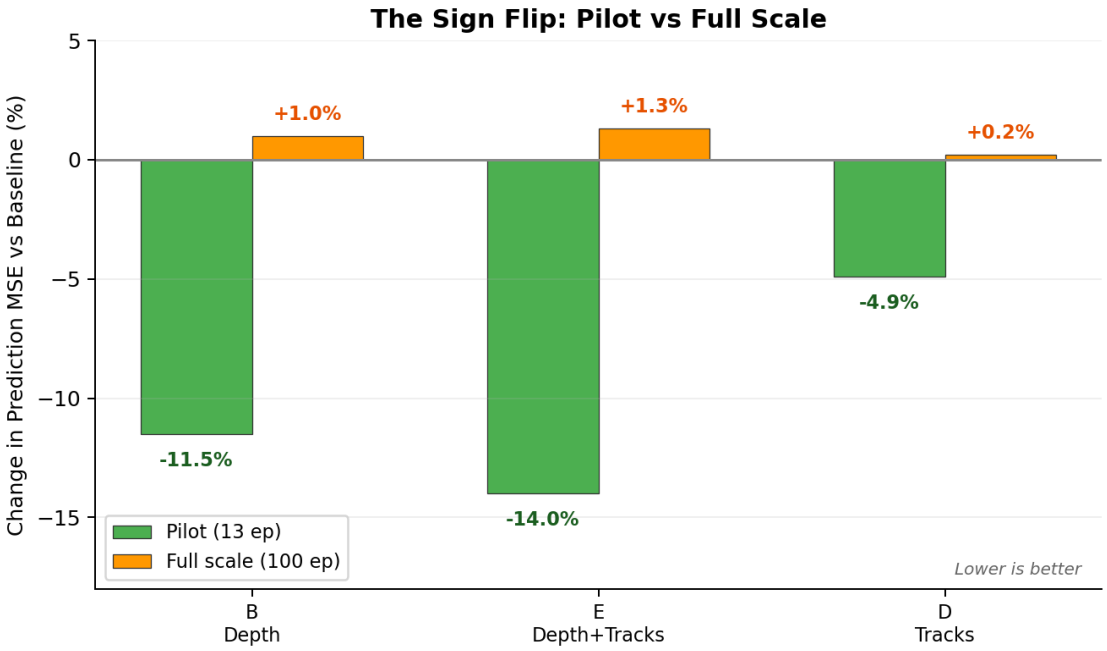

Figure 3: Pilot vs full-scale comparison for the three conditions run at both data scales. The pilot-scale advantages of depth (B), depth+tracks (E), and tracks (D) all collapse at 100 episodes and three of three even reverse direction.

4.5 Qualitative evaluation via retrieval

Because LeWorldModel predicts in latent space and does not ship with a pixel decoder, qualitative evaluation requires care. We trained a cross-attention decoder from CLS embeddings to RGB but achieved insufficient quality at the available data scale to use as primary visual evidence. Instead, we use a retrieval procedure: for each predicted CLS, we find the L2-nearest real frame in the corpus and display it (Figure 4). The retrieval is a proxy rather than a generation, but it preserves image fidelity and reflects the model's latent-space behavior more reliably than the underdetermined decoder.

5 Discussion

The sign flip is the central finding. Auxiliary CV signals that improved prediction by 11% to 14% on a 13-episode curated set lost all benefit when the same training recipe was applied to 100 episodes from the same corpus. We propose three readings of this result, in order of confidence.

First, auxiliary supervision functions as inductive bias whose value is governed by the gap between the prior the model can learn from data and the prior the auxiliary signal encodes. At small data scale, the gap is large and the auxiliary loss is informative. At larger data scale, the model learns the same structure implicitly and the auxiliary loss becomes either redundant (no gradient mass goes anywhere new) or interfering (gradient mass goes somewhere that conflicts with the next-latent objective). This account is consistent with concurrent findings in robot foundation-model literature, including Pi0's report that data-quality curation outperforms algorithmic tweaks by a factor of three to ten at large scale [8] and DreamDojo's argument that web-scale video pretraining absorbs hand-engineered signals [10].

Second, the value of an auxiliary signal is gated by whether the predictor architecture can use it. Dense local signals (per-point tracks) cannot be recovered from a single global latent through a flat decoder. Global

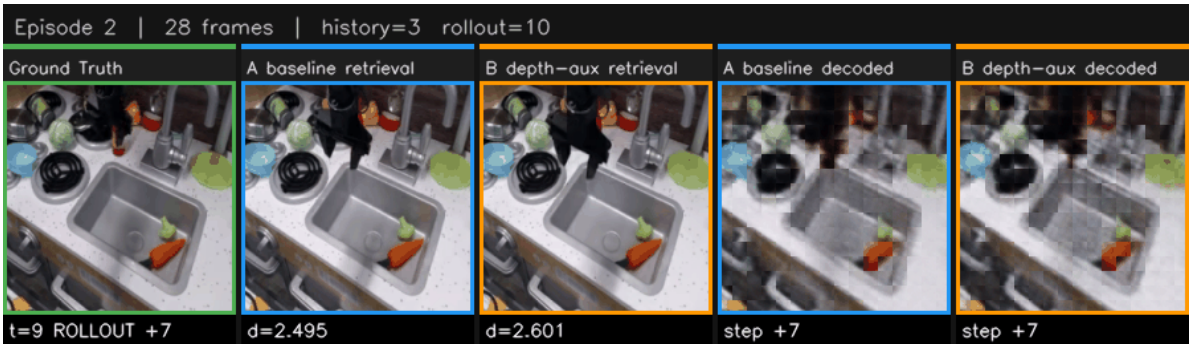

Figure 4: Single rollout frame from a side-by-side comparison (full animation in the supplementary materials). Columns left to right: ground truth, baseline retrieval, depth-aux retrieval, baseline decoder, depth-aux decoder. The retrieval columns nearest-neighbor the predicted latent into the corpus of real frames and report the L2 distance; the decoder columns reconstruct pixels through a learned cross-attention decoder. Retrieval preserves visual fidelity at the cost of being a proxy; the decoder is direct but suffers from the underdetermined inverse problem at the available training scale.

scene-level signals (full-frame depth) can. Track-supervised work that succeeds at scale uses architectures with per-track tokens and local visual features [19]. The CLS bottleneck is not unique to LeWorldModel and applies to any predictor that compresses to a single token before regressing dense targets.

Third, the methodological implication for foundation-model design is that explicit CV signals are most valuable at training time over curated data, and least valuable as inference-time decoded outputs at scale. This positions explicit 3D representations as instruments for shaping the latent rather than artifacts to be deployed. Concurrent industry work appears to have arrived at similar conclusions through different paths, including Pi0's metadata-conditioning approach [7] and the broader migration of 3D vision from policy-loop components to data-engineering tools.

6 Limitations

The findings are bounded by the testbed. The base model is small (15M parameters) and findings may not transfer to multi-billion-parameter video generators or foundation-scale VLAs. The dataset uses a fixed overhead camera, which forecloses true multi-view 3D experiments and may bias the results toward signals that are camera-invariant. We measure next-latent prediction MSE rather than downstream manipulation success, and the relationship between latent prediction quality and policy performance is not characterized here. Auxiliary heads are simple MLPs over the CLS token; richer prediction heads (per-patch attention, per-token dense decoders) could change the picture and partially defeat the bottleneck finding. The seed counts vary across conditions (3–10) and the full-scale set is run with three seeds, which limits statistical power for small effect sizes near baseline.

7 Conclusion

We ran a controlled fine-tuning ablation of seven explicit-vision auxiliary heads on a JEPA-style world model. Depth-conditioned supervision improved next-latent prediction by 11.5% to 14.5% on a 13-episode curated pilot, then collapsed or reversed at 100 episodes. We interpret this sign flip as auxiliary heads functioning as scale-dependent inductive bias, with diminishing returns once the base model can learn the same structure implicitly. The implication for foundation-model teams is that explicit 3D vision is most useful as a training-time instrument over curated data, not as an inference-time deployment target. The natural

next step is to repeat this ablation with a patch-aware predictor or per-token dense head that can route local supervision through the model, and on a multi-view dataset where 3D supervision can encode information unavailable from a single view.

Code and data

The fine-tuning toolkit, extraction pipelines, and evaluation harness are available at https://github.com/kevdozer1/leWN_finetune_toolkit. The companion blog post with extended discussion and additional figures is at https://kevdozer1.com/blog/2026/lewm-finetune/.

References

[1] Y. LeCun. A Path Towards Autonomous Machine Intelligence. OpenReview, 2022.

[2] M. Assran et al. V-JEPA: Latent Video Prediction for Visual Representation Learning. 2024.

[3] G. Zhou et al. DINO-WM: World Models on Pre-trained Visual Features Enable Zero-shot Planning. 2024.

[4] LeWorldModel: A Compact Latent World Model for Robot Policy Pretraining. https://le-wm.github.io/, 2026.

[5] Causal JEPA: Object-Centric Latent World Models with Counterfactual Masking. 2026.

[6] VL-JEPA: Vision-Language Joint Embedding Predictive Architectures. 2026.

[7] Physical Intelligence. $\pi_{0.5}$: A VLA Model with Subgoal-Image Conditioning. Technical report, 2026.

[8] Hugging Face Robotics Team. Fine-Tuning π_0 for Shirt Folding: Data Quality vs. Algorithmic Tweaks. https://x.com/DominiqueCAPaul/status/2042658951301382464, 2026.

[9] NVIDIA. GR00T N1.7: An Open Humanoid Foundation Model. Technical report, 2026.

[10] NVIDIA. DreamDojo: An Open Foundation World Model from Egocentric Human Video. Technical report, 2026.

[11] VideoManip: Reconstructing Hand-Object Trajectories from RGB Video for Robot Policy Learning. 2026.

[12] Re2Pix: Hierarchical Latent-then-Pixel Video World Models. 2026.

[13] V. Sitzmann. The Bitter Lesson of Computer Vision. https://www.vincentsitzmann.com/blog/bitter_lesson_of_cv/, 2025.

[14] H. Walke et al. BridgeData V2: A Dataset for Robot Learning at Scale. CoRL, 2023.

[15] W. Yang et al. Video Depth Anything: Consistent Depth Estimation for Super-Long Videos. 2024.

[16] N. Karaev et al. CoTracker3: Simpler and Better Point Tracking by Pseudo-Labelling. 2024.

[17] Meta AI. Segment Anything Model 3: Text-Driven Video Segmentation. 2025.

[18] W. Peebles and S. Xie. Scalable Diffusion Models with Transformers. ICCV, 2023.

[19] Motion Forecasting via Track Tokens with DINOv3 Features. https://motion-forecasting.github.io/, 2026.