WORLDMODELLENS: FROM ATTENTION TO ROLLOUTS, A FRAMEWORK FOR WORLD MODEL INTERPRETABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned world models now sit inside agents that plan, generate interactive environments, and control physical systems, but the interpretability toolkit that has grown up around language models does not transfer to them. World models condition on actions, maintain recurrent and often stochastic state, and are queried through multi-step rollouts; none of these has a natural home in libraries built around a single forward pass over tokens. We present WorldModelLens, an open-source library that exposes recurrent state-space models, token-based transformers, and joint-embedding predictive architectures through one capability-typed interface of four core methods. On top of it, a time-indexed hook layer supports caching, imagination, and intervention replay, so that probing, patching, attribution, and lens analyses are implemented once and run on any of nine supported backends. We use the library for an in-depth study of the I-JEPA ViT-H/14 predictor. Enforcing full key coverage when loading the checkpoint reveals that non-strict loading silently leaves all MLP weights and QKV biases at their random initialization. With the correct weights, Integrated Gradients recovers context that is sufficient for the prediction (Cohen's $d = -1.00$ against random, $N = 1{,}000$), whereas attention is only weakly aligned with it at every predictor depth. Zero, mean, and resample ablations agree that the prediction passes through the target-token positions. A latent lens places the point at which predicted content overtakes a position-only baseline at depth 3.84 of 12, several layers after the content becomes identifiable by retrieval. Finally, we find that attribution patching matches exhaustive patching at $179\times$ lower cost only when the gradient estimator is chosen according to the size of the intervened site.

1 INTRODUCTION

Model-based agents have long relied on a learned simulator of their environment (Ha & Schmidhuber, 2018). What has changed recently is the scale and reach of these simulators. DreamerV3 trains its policy almost entirely on imagined rollouts and does so across more than 150 tasks with a single configuration (Hafner et al., 2025); TD-MPC2 plans in the latent space of a joint-embedding model (Hansen et al., 2024); IRIS and DIAMOND model Atari games with a transformer over discrete tokens and with a diffusion model, respectively (Micheli et al., 2023; Alonso et al., 2024). At the other end of the spectrum, Genie and its successors generate controllable environments from a single prompt (Bruce et al., 2024; Parker-Holder et al., 2024; Parker-Holder & Fruchter, 2025), Cosmos and GAIA-2 are trained as simulators for robotics and driving (Agarwal et al., 2025; Russell et al., 2025), and V-JEPA 2 is used to plan robot manipulation directly in representation space (Assran et al., 2025). In all of these systems, decisions are made against the model's predictions, so an error in what the model believes about the future becomes an error in behavior.

Despite this, we know very little about how world models arrive at their predictions. In language models, the corresponding questions (which components carry a piece of information, how it is routed, and what happens when it is removed) have become tractable in large part because of shared tooling (Nanda & Bloom, 2022; Fiotto-Kaufman et al., 2024; Wu et al., 2024). Once activations can be named, cached, and overwritten in a uniform way, a technique such as activation patching (Vig

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2020; Meng et al., 2022) can be applied to a new checkpoint in an afternoon, and results obtained on different models can be compared (Wang et al., 2023; Conmy et al., 2023).

No comparable infrastructure exists for world models, and the gap is not only one of engineering effort. Language-model libraries are built around a sequence of tokens processed in one forward pass, with activations indexed by layer and position. That abstraction does not describe how world models are used. A Dreamer agent carries a deterministic recurrent state together with a stochastic latent; an intervention on one of them at step t affects every subsequent imagined step, and the quantity of interest is often the trajectory rather than a single output. Actions enter the computation as a separate input whose effect we may want to isolate. Joint-embedding predictors such as I-JEPA (Assran et al., 2023) have no decoder at all, so their predictions cannot be inspected by rendering them. As a result, existing interpretability work on world models and related agents has been carried out one architecture at a time: probing and editing board states in Othello and chess models (Li et al., 2023; Nanda et al., 2023; Karvonen, 2024), reverse-engineering planning in recurrent and model-free agents (Taufeeque et al., 2024; 2026; Bush et al., 2025; Jenner et al., 2024), probing video models for physical variables (Garrido et al., 2025; Joseph et al., 2026; Punzo et al., 2026), and probing the latents of IRIS, DIAMOND, and DreamerV3 (Zhang, 2026; Eaton et al., 2024; Aadithiyan, 2026). Each of these studies builds its own instrumentation, which makes it hard to reuse methods or to compare findings across model families.

This paper introduces WorldModelLens, a library intended to make interpretability of world models as routine as it has become for language models. Its central design decision is to separate what a model *is* from what an analysis *needs* (Figure 1). Every backend implements the same four methods and declares which optional heads (decoder, reward, continuation, actor, critic) it provides; analyses state which of these they require. A single hook layer indexes activations by name and time step, which is enough to express component-level, step-level, action-level, and rollout-level interventions with the same machinery.

We then use the library to study one model in depth. I-JEPA ViT-H/14 is a demanding test case: it has no decoder, its training targets come from a separate EMA encoder, and its 630.8M-parameter encoder makes exhaustive interventions expensive. Our contributions are threefold. First, we define a capability-typed interface and a time-indexed hook layer for world models, together with an intervention-replay procedure that propagates edits through rollouts, and implement them for nine backends (§3). Second, we provide analysis modules with controls built in, namely replacement ablations, validated attribution patching, and a latent lens with a positional baseline, backed by a statistics layer that is applied by default (§3.3). Third, we report a detailed mechanistic study of the I-JEPA ViT-H/14 predictor (§4). It uncovers a silent checkpoint-loading failure, shows that the prediction is routed through the target-token positions, gives a depth-resolved account of when target content becomes readable, and characterizes when attribution patching can stand in for exhaustive patching.

2 Related work

World models. Latent dynamics models learned from pixels date back at least to Ha & Schmidhuber (2018). Recurrent state-space models (RSSMs), used in PlaNet and the Dreamer series (Hafner et al., 2020; 2025), split the state into a deterministic recurrent component and a stochastic latent. Token-based models discretize observations and model them autoregressively (Micheli et al., 2023), while diffusion world models predict frames directly (Alonso et al., 2024; Valevski et al., 2025). Joint-embedding predictive architectures (Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025) and TD-MPC2 (Hansen et al., 2024) instead predict in a learned representation space without reconstructing observations. Large generative world models are increasingly proposed as training environments for embodied agents (Yang et al., 2024; Agarwal et al., 2025; Parker-Holder & Fruchter, 2025).

Interpreting world models and agents. Sequence models trained on board games develop internal board representations that can be probed and edited (Li et al., 2023; Nanda et al., 2023; Karvonen, 2024), and language models represent space and time linearly (Gurnee & Tegmark, 2024). Vafa et al. (2024) show that a generative model can predict well while its implicit world model is incoherent, which behavioral evaluation alone may miss. Mechanistic accounts of planning exist for a Sokoban

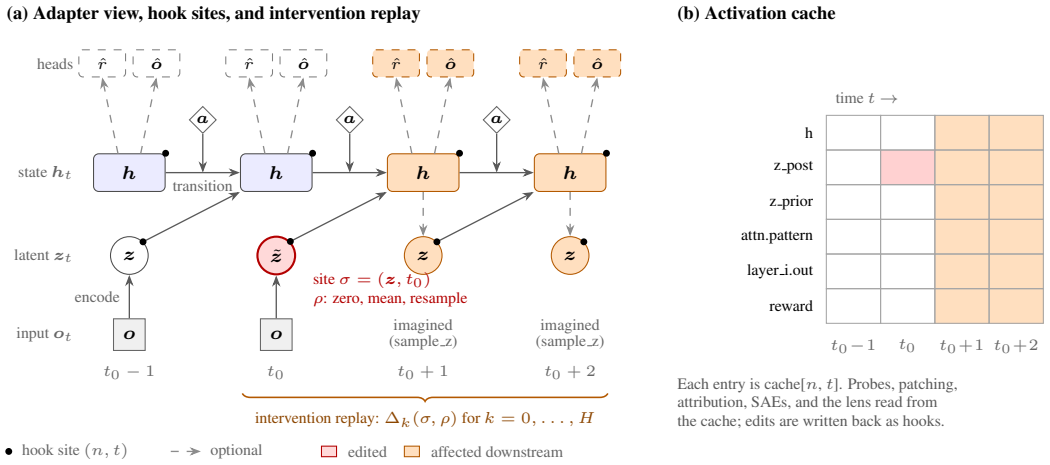

Figure 1: Architecture of WorldModelLens. (a) The adapter view of a world model unrolled over time: observations are encoded into latents $\boldsymbol{z}_t$, transition advances the state $\boldsymbol{h}_t$ under actions $\boldsymbol{a}_t$, optional heads read from the state, and after t_0 the model imagines forward with sample_z. Every state and latent is a hook site (n, t). Intervention replay overwrites one site with a zero, mean, or resampled value and measures the effect on all downstream steps over a horizon H. (b) The activation cache stores every site by name and time step; the analysis modules read from it and write edits back through the same hooks.

RNN (Taufeeque et al., 2024; 2026), for model-free agents (Bush et al., 2025), and for a chess network (Jenner et al., 2024). For video models, physical variables are decodable from intermediate layers (Garrido et al., 2025; Joseph et al., 2026; Punzo et al., 2026; Esmati et al., 2026), and prediction objectives yield more action-relevant latents than reconstruction (Yeom et al., 2026). Closest to our setting, Zhang (2026) probe and patch IRIS and DIAMOND, Eaton et al. (2024) find IRIS codebook entries hard to interpret, and Aadithiyan (2026) decode a confusion signal from DreamerV3 state. Surveys of world-model risk (Parmar, 2026) describe failure modes behaviorally; we are not aware of mechanistic work on JEPA predictors.

Interpretability tooling. TransformerLens (Nanda & Bloom, 2022), NNsight (Fiotto-Kaufman et al., 2024), pyvene (Wu et al., 2024), nnterp (Dumas, 2025), and Penzai (Johnson, 2024) provide activation access and interventions for sequence models; Prisma (Joseph et al., 2025) brings similar functionality to vision and video encoders, and SAELens (Bloom et al., 2024) supports sparse dictionary learning. Captum and Zennit (Kokhlikyan et al., 2020; Anders et al., 2021) implement input attribution. None of these represents actions, recurrent or stochastic state, or imagined rollouts. The interface in WorldModelLens was first outlined in a preprint (Challagundla et al., 2026); here we add the intervention-replay formalism, the controlled analysis modules, and the first large-scale study built on it.

Patching, attribution, and lenses. Activation patching is sensitive to the choice of corruption and metric (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024). Attribution patching replaces it with a first-order approximation (Nanda, 2023; Syed et al., 2023; Kramár et al., 2024), and path-integrated variants improve faithfulness in some regimes (Hanna et al., 2024). Whether attention weights constitute explanations remains disputed (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019; Abnar & Zuidema, 2020; Chefer et al., 2021). Deletion and insertion curves (Samek et al., 2017; Petsiuk et al., 2018) are the standard causal test for input attributions such as Integrated Gradients (Sundararajan et al., 2017). Our latent lens follows the logit lens and tuned lens (nostalgebraist, 2020; Belrose et al., 2023).

3 WORLDMODELLENS

3.1 A COMMON INTERFACE FOR WORLD MODELS

We consider a world model that, at step t, maintains a deterministic state $\boldsymbol{h}_t$ and a latent $\boldsymbol{z}_t$, receives an observation $\boldsymbol{o}_t$ and an optional action $\boldsymbol{a}_t$, and advances according to

$$\begin{aligned} (\boldsymbol{z}_t, \boldsymbol{z}_t^{\text{prior}}) = \text{encode}(\boldsymbol{o}_t, \boldsymbol{h}_{t-1}), \quad \boldsymbol{h}_{t+1} = \text{transition}(\boldsymbol{h}_t, \boldsymbol{z}_t, \boldsymbol{a}_t), \\ \boldsymbol{h}_0 = \text{initial_state}(B), \qquad \boldsymbol{z}_t \sim \text{sample_z}(\boldsymbol{\ell}_t, \tau). \end{aligned} \tag{1}$$

Here sample_z applies a Gumbel-softmax relaxation to the logits $\boldsymbol{\ell}_t$ when latents are categorical and reduces to the identity for continuous or deterministic ones. These four methods are mandatory. Each adapter also declares a capability set $\mathcal{K} \subseteq \{\text{decode}, \text{reward}, \text{continue}, \text{actor}, \text{critic}\}$ of optional heads, and each analysis declares the set $\mathcal{R}$ it requires; the library runs an analysis on a backend whenever $\mathcal{R} \subseteq \mathcal{K}$. Reward attribution thus runs on DreamerV3 and TD-MPC2 but is refused for I-JEPA, whereas probing and patching, which need only the core methods, run everywhere.

Equation 1 covers the families listed in Figure 1 and Appendix A with little strain. The repository also contains experimental adapters for video, diffusion, robotics, and driving world models (Figure 1), which we do not evaluate here. For an RSSM, $\boldsymbol{h}_t$ is the GRU state and $\boldsymbol{z}_t$ the stochastic latent. For IRIS, $\boldsymbol{z}_t$ is the block of codebook indices produced by the tokenizer and transition is the autoregressive transformer. For a joint-embedding predictor, encode produces patch embeddings of the visible context, transition is the predictor that maps them, together with positional queries, to predicted target embeddings, and no action is supplied. Because I-JEPA predicts in representation space, its adapter declares no optional heads.

3.2 TIME-INDEXED SITES AND INTERVENTION REPLAY

Every activation computed inside Equation 1 is registered under a standardized name n (for instance encoder.layer_6.hook_output or attn.hook_pattern). Unlike in language-model libraries, a *site* is the pair $\sigma = (n, t)$, where t indexes the environment or imagination step. run_with_cache executes a trajectory and returns the cache $\{\boldsymbol{a}_\sigma\}$ over all sites; run_with_hooks executes it while applying user functions at chosen sites; and imagine rolls transition forward from a given state without observations, drawing actions from the actor head when actor $\in \mathcal{K}$. Keeping the time index in the site name is what allows a single mechanism to express interventions at the level of a component (fix t, vary n), a step (fix t, intervene on $\boldsymbol{h}_t$ or $\boldsymbol{z}_t$ as a unit), an action (intervene on $\boldsymbol{a}_t$), or a rollout (vary t and measure the downstream trajectory).

The central operation is *intervention replay* (Algorithm 1). Given a recorded trajectory, a site $\sigma = (n, t_0)$, and a replacement rule ρ, the library overwrites $\boldsymbol{a}_\sigma$, recomputes every activation that depends on it for steps $t_0, \ldots, t_0 + H$, and reports the effect on a metric m at each step:

$$\Delta_k(\sigma, \rho) = m\big(\hat{\boldsymbol{y}}_{t_0+k}^{\sigma \leftarrow \rho}\big) - m\big(\hat{\boldsymbol{y}}_{t_0+k}\big), \qquad k = 0, \ldots, H. \tag{2}$$

For single-step models such as I-JEPA, $H = 0$ and replay reduces to ordinary activation patching from the intervened layer onward. The replacement rule is part of the specification. WorldModelLens supports *zero* replacement, *mean* replacement (the dataset mean at that site), and *resample* replacement (the activation at the same site for a different input). Running all three is our default, because zero ablation places the network off its activation distribution and can overstate importance (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024). When no hook is active, the instrumentation costs roughly 12% per step (Challagundla et al., 2026), which we found low enough to leave caching enabled during routine evaluation.

3.3 ANALYSES AND STATISTICS

The analysis modules build on the cache and on Algorithm 1. They include linear, ridge, and logistic probes; sparse autoencoders with ReLU, top-k, and gated activations; activation patching and causal tracing; surprise and uncertainty decomposition; and representational similarity, including linear CKA (Kornblith et al., 2019). For this work we added three modules, each prompted by a problem we ran into during the case study. The first is attribution patching (Nanda, 2023) in two variants, a one-step estimate at the corrupted activation and a path-integrated estimate (Hanna et al., 2024),

Algorithm 1 Intervention replay with replacement controls

Input: adapter W, trajectory $\tau = (\boldsymbol{o}_{0:T}, \boldsymbol{a}_{0:T})$, site $\sigma = (n, t_0)$, horizon H, metric m, rules $\mathcal{P} \subseteq \{\text{zero}, \text{mean}, \text{resample}\}$

$\{\boldsymbol{a}_{\sigma'}\}, \hat{\boldsymbol{y}}_{0:T} \leftarrow W.\text{run_with_cache}(\tau)$ ▷ clean pass

for $\rho \in \mathcal{P}$ **do**

  $\tilde{\boldsymbol{a}}_\sigma \leftarrow \text{Replace}(\boldsymbol{a}_\sigma, \rho)$ ▷ $\mathbf{0}$, $\mathbb{E}[\boldsymbol{a}_\sigma]$, or $\boldsymbol{a}_\sigma(\tau')$ for a donor τ'

  $\hat{\boldsymbol{y}}^{\rho}_{t_0:t_0+H} \leftarrow W.\text{run_with_hooks}(\tau,\ \{\sigma \mapsto \tilde{\boldsymbol{a}}_\sigma\})$ from layer n at step t_0

  $\Delta_k(\sigma, \rho) \leftarrow m(\hat{\boldsymbol{y}}^{\rho}_{t_0+k}) - m(\hat{\boldsymbol{y}}_{t_0+k})$ for $k = 0, \ldots, H$

end for

return $\{\Delta_k(\sigma, \rho)\}$ with bootstrap intervals over trajectories

together with a harness that compares either one with exact patching on a user-chosen subset of sites. The second is a latent lens that applies a model's own output head to intermediate states and reports it against a position-only baseline (§4.3). The third is a statistics layer that attaches paired t and Wilcoxon tests, Cohen's d (Cohen, 1988), percentile or cluster bootstrap intervals (Efron, 1979), and Benjamini-Hochberg or Holm correction (Benjamini & Hochberg, 1995; Holm, 1979) to every reported comparison.

3.4 Checkpoint integrity

Loading the public I-JEPA ViT-H/14 weights turned out to be less straightforward than expected. The released blocks include QKV biases (attn.qkv.bias), and several parameter names (mlp.fc1, mlp.fc2, predictor_blocks) differ from those in common ViT reimplementations. With load_state_dict(strict=False), which is what our own earlier loader used, the mismatched tensors are skipped without warning: every MLP and every QKV bias stays at its random initialization, and the model still runs and produces plausible-looking outputs. The WorldModelLens loader now remaps names, attaches the missing biases, and refuses to proceed unless every tensor is matched in both directions (389/389 for the context encoder, 389/389 for the target encoder, and 152/152 for the predictor). It also writes the coverage table into each results file. We suspect this failure is not specific to I-JEPA, since interpretability code frequently reimplements released architectures rather than importing the original training code.

4 Case study: the I-JEPA ViT-H/14 predictor

Setup. We use the official I-JEPA checkpoint trained on ImageNet-1K for 300 epochs (Assran et al., 2023). An image $\boldsymbol{x}$ of size 224×224 is divided into $N = 256$ patches. A context encoder f_θ (32 blocks, width 1,280, 630.8M parameters) embeds the visible patches $\mathcal{C}$. The predictor g_ϕ (12 blocks, width 384, 22.4M parameters) receives $[\boldsymbol{z}_\mathcal{C}; \boldsymbol{e}_\mathcal{Y}]$, where $\boldsymbol{e}_\mathcal{Y}$ are a learned mask token plus positional embeddings for the target positions $\mathcal{Y}$, and outputs $\hat{\boldsymbol{z}}_\mathcal{Y}$. During pretraining, targets are produced by an EMA copy $f_{\bar{\theta}}$ of the encoder (Grill et al., 2020), and the loss is $|\mathcal{Y}|^{-1} \sum_{j \in \mathcal{Y}} \|\hat{\boldsymbol{z}}_j - f_{\bar{\theta}}(\boldsymbol{x})_j\|_2^2$. Unless stated otherwise, we evaluate on 1,000 ImageNet-1K validation images (Russakovsky et al., 2015), 20 per class from 50 classes, drawn with a fixed seed. For intervention experiments we fix the target to the central 4×4 block ($|\mathcal{Y}| = 16$, with the remaining 240 patches as context), so that token positions are aligned across images. The attribution experiments use a visible-context budget of 20% ($|\mathcal{C}| = 51$).

4.1 Context attribution: gradients and attention disagree

For a target token j, define the prediction error $E_j(\boldsymbol{z}_\mathcal{C}) = \frac{1}{2}\|g_\phi([\boldsymbol{z}_\mathcal{C}; \boldsymbol{e}_j])_j - \boldsymbol{z}_j^{\text{ref}}\|_2^2$. We score each context token i with Integrated Gradients (Sundararajan et al., 2017),

$$A_{\text{IG}}(i \to j) = \big(\boldsymbol{z}_{\mathcal{C},i} - \boldsymbol{z}^{(0)}_{\mathcal{C},i}\big)^\top \int_0^1 \frac{\partial E_j\big(\boldsymbol{z}^{(0)}_\mathcal{C} + \alpha(\boldsymbol{z}_\mathcal{C} - \boldsymbol{z}^{(0)}_\mathcal{C})\big)}{\partial \boldsymbol{z}_{\mathcal{C},i}}\, d\alpha, \tag{3}$$

using a zero baseline $\boldsymbol{z}^{(0)}$ and a 50-step Riemann sum. We compare against attention attribution, defined as the mean attention from target queries to context keys in the last two predictor layers, and against random scores. Each ranking is evaluated with a deletion curve, in which the top-K context

Table 1: Faithfulness of context attributions for I-JEPA ViT-H/14 ($N = 1{,}000$). Areas under the deletion and insertion curves of prediction MSE, with 95% bootstrap intervals. d is the paired Cohen's d; all I-JEPA comparisons have Benjamini-Hochberg-adjusted $p < 10^{-7}$. The MAE row uses pixel MSE and $N = 20$, so only its effect sizes are comparable to the I-JEPA rows.

	Deletion AUC ↑	Insertion AUC ↓	d_{del}	d_{ins}
Random	0.739 [0.716, 0.761]	1.355 [1.332, 1.379]		
Attention	**0.957** [0.933, 0.980]	1.278 [1.254, 1.303]		
IG	0.924 [0.899, 0.948]	**1.255** [1.231, 1.281]	$+1.03$	-1.00
IG vs. attention			-0.18	-0.25
MAE, IG vs. random ($N = 20$)	2.580 vs. 2.571	2.544 vs. 2.583	$+0.56$	-1.15

tokens are removed and a faithful ranking should increase the error quickly, and an insertion curve, in which tokens are added to an empty context in ranked order and a faithful ranking should decrease the error quickly (Samek et al., 2017; Petsiuk et al., 2018). We use $K \in \{1, 3, 5, 10, 20\}$ and summarize each curve by its area.

Both methods are far better than chance (Table 1). For IG against random, $t = 32.4$ on deletion and $t = -31.8$ on insertion, and the Wilcoxon test gives $p < 10^{-126}$ in both cases. The two methods do not select the same tokens, however. Removing the tokens that attention ranks highest causes more damage than removing IG's choices ($d = -0.18$), whereas IG's choices restore the target more quickly when inserted ($d = -0.25$). One reading is that attention concentrates on tokens that the predictor routes information through, while IG identifies a smaller set that is sufficient to reconstruct the target. On a pixel-reconstruction MAE (He et al., 2022), the insertion effect of IG is of similar size ($d = -1.15$) and the deletion effect is weaker ($d = +0.56$, $p = 0.02$); with only 20 images we regard this comparison as preliminary.

We also measured, for every predictor layer, how well the attention scores at that layer agree with IG (Appendix C). The per-image Spearman correlation lies between 0.19 and 0.27 at all twelve layers, and the top-6 Jaccard overlap between 0.14 and 0.18. Agreement is lowest at the final layer ($\rho = 0.186$, with 20.8% of images showing a negative correlation), but the trend across depth is weak and not monotonic. We therefore do not read these results as attention "failing" in deep layers; attention is a poor proxy for gradient-based importance throughout the predictor.

4.2 Replacement controls: the prediction is carried by target tokens

We applied Algorithm 1 with $H = 0$ to the predictor residual stream, replacing either the 16 target-token positions or a block of context positions at one layer and recomputing the predictor from that layer onward. Averaged over the 12 layers, replacing the target tokens increases MSE by a factor of 2.07 under mean replacement, 3.00 under zero replacement, and 3.36 under resampling (Figure 2a). Applying the same three replacements to the 48 context positions in the last quarter of the sequence changes MSE by at most a factor of 1.006, and a linear probe on the prediction stays at 85.5 to 85.8% accuracy (85.5% without intervention). The ordering mean $<$ zero $<$ resample holds on almost every image: mean is less damaging than zero for 99.8% of images, and resampling is more damaging than zero for 79.6%. Because the same interventions are harmless at some positions and damaging at others, the effect at the target positions cannot be attributed to the intervention merely pushing activations off-distribution. At layer 11, every replacement mode reduces the linear probe to chance (2%), which is expected since the target-token states at that depth are the output.

Context does matter, but mainly early. At layer 0, resampling the second quarter of the context sequence (Q1) almost triples the error ($2.95\times$), more than resampling the last quarter (Q3, $1.21\times$); Q2 is intermediate ($1.75\times$) and Q0 has almost no effect ($1.01\times$; Figure 2b). If the context tokens are ordered by raster position, Q1 corresponds to grid rows 4 to 8 and contains most of the positions adjacent to the central target block, so this ordering is consistent with early influence depending on distance to the target. A direct test with an adjacency-based partition remains to be done. By layer 11, replacing any context quarter leaves the output unchanged (MSE ratio 1.000). On a smaller subset of 132 to 180 images, zeroing a single target token raises MSE from about 0.228 to 0.251 to 0.256 regardless of which of the 16 positions is chosen, and zeroing any one of four 96-channel blocks

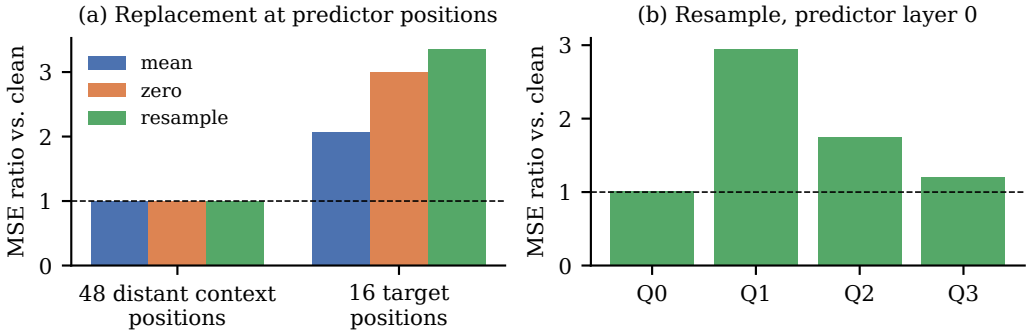

Figure 2: Replacement controls in the I-JEPA predictor. (a) Replacing the 16 target-token positions raises prediction MSE by a factor of 2.1 to 3.4 under all three rules, while the same replacements at 48 context positions far from the target have no effect (mean over 12 layers, $N = 1{,}000$). (b) At predictor layer 0, resampling the context quarter Q1, which contains most of the positions adjacent to the target block, has the largest effect.

raises it to 0.45 to 0.51. The prediction is thus localized across token positions but spread evenly over channels. Resampling from donors of the same class does less damage than resampling from distant classes (Appendix D), which suggests that the semantic content of the replacement matters and not only whether it is a plausible activation.

4.3 Latent lens: identification precedes alignment

The predictor ends with an output head $\hat{\boldsymbol{z}} = W_{\text{out}}\,\text{LN}(\boldsymbol{h})$ that maps its 384-dimensional residual stream into the encoder's space. We apply the same head to the residual stream after each block, so that the reading at the last block coincides with the model's output. At layer 0 a target token consists only of the mask token and a positional embedding and is identical across images; the implementation checks that its spread across images is exactly zero. We report two quantities. The first is the cosine similarity between the lens reading and the true EMA-target embedding, compared with a *positional prototype* that predicts the mean target embedding at that position without looking at the image. The second is top-1 retrieval of the correct image among 240 candidates at the same patch position; since position is fixed within this gallery, a positional prior cannot exceed chance (0.42%). Results use 240 images and 32 masks, giving 331,680 scored tokens per layer, with 10,000 bootstrap replicates resampled over images and, separately, over images and masks. The same replicates are used at every layer so that comparisons across layers are paired.

The cosine similarity increases from 0.092 at layer 0 to 0.724 at the output (Figure 3a), while the positional prototype, which ignores the image, achieves 0.336. For layers 0 to 3 the lens reading is significantly *worse* than the prototype (Holm-adjusted $p \leq 2.6 \times 10^{-3}$ under both resampling schemes). Interpolating within each bootstrap replicate, the two curves cross at depth 3.84, with a 95% interval of [3.71, 3.98] when resampling images and [3.62, 4.04] when resampling images and masks; every replicate crosses. Retrieval tells a different story: at layer 3 the correct image is already ranked first in 16.8% of cases, 40 times chance (Figure 3b). The predictor therefore separates the correct target from alternatives before its prediction is closer to the target than a positional guess would be, a distinction that a retrieval-based evaluation alone would not reveal. All twelve layer-to-layer changes in retrieval accuracy remain significant after Holm correction, including a small decrease from layer 8 to 9 (-1.0 percentage points). Most of the increase happens between layers 3 and 5 ($+28.7$ and $+25.2$ points), with a further step from 9 to 10 ($+10.5$ points), and no single step accounts for more than 31% of the total.

4.4 When can attribution patching replace exhaustive patching?

Exhaustive patching of all 836 head and block sites in the encoder and predictor takes 87.3 s per image pair on the 4 GB laptop GPU we used. We therefore evaluated two gradient estimates. The

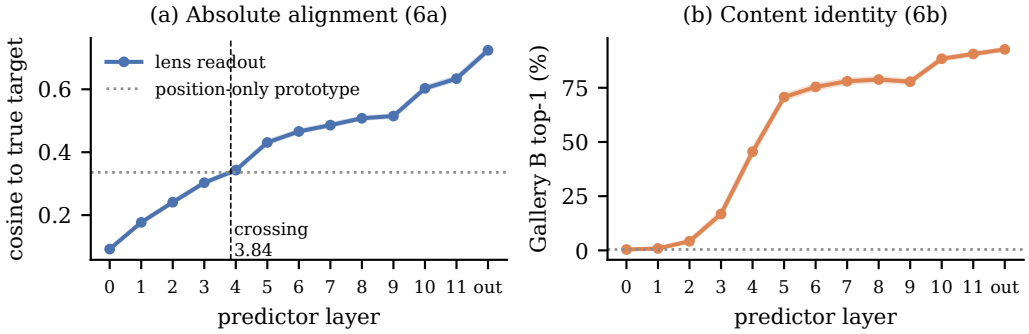

Figure 3: Latent lens on the I-JEPA ViT-H/14 predictor with 95% bootstrap bands. (a) Cosine similarity to the true target remains below the positional prototype (dotted) until an interpolated depth of 3.84. (b) Retrieval of the correct image at a fixed position is already 40 times chance at layer 3.

Table 2: Attribution patching against exact activation patching on I-JEPA ViT-H/14 (4 validated image pairs). Time is per image pair for all 836 sites.

Estimator	Time	Speedup	All sites r / ρ	Heads r / ρ	Residual stream r
Exact patching	87.3 s	1×			
AP	0.49 s	179×	0.53 / 0.61	**0.70 / 0.67**	
AP-IG (8 steps)	3.33 s	26×	**0.93 / 0.77**	0.49 / 0.47	0.24

corrupted input encodes a different image's context under the same mask, so positions correspond one to one, and effects are expressed as the fraction of the gap between the clean and corrupted runs, measured in the sign-flipped I-JEPA loss, that a site recovers; the mean gap over 12 pairs is 0.684. Plain attribution patching (AP) estimates the effect of site $\boldsymbol{a}$ as $(\boldsymbol{a}_{\text{clean}} - \boldsymbol{a}_{\text{corr}})^\top \nabla_{\boldsymbol{a}} L|_{\text{corr}}$ (Nanda, 2023). AP-IG averages the gradient over eight points on the path from the corrupted to the clean activation (Hanna et al., 2024). We compared both estimates with 784 exact patching measurements; as a sanity check, patching enc.resid_pre.0 recovers exactly 1.000 of the gap in every pair.

Neither estimator is uniformly better (Table 2; Appendix E). For large sites, namely residual-stream positions and complete block outputs (whose true effects average 4.9% of the gap and reach 67.7%), a single first-order step is inaccurate. Plain AP places residual-stream sites anywhere between -0.7 and $+1.0$ when their true effect lies between $+0.55$ and $+0.79$, and averaging along the path reduces this spread. For individual heads, whose effects are small (1.1% on average, at most 10.5%), the linearization at the corrupted activation is the appropriate one, and plain AP outperforms AP-IG ($r = 0.70$ versus 0.49). Pooled correlations obscure this: AP-IG achieves $r = 0.93$ across all sites but only 0.24 within the encoder residual stream. Our recommendation is therefore to use plain AP when ranking heads, AP-IG for block-level sites, and exact patching, at 104 ms per site, to confirm any shortlist. Applying each estimator where it is reliable, the encoder shows a large contribution from the layer-0 MLP (0.354 of the gap) and head attributions concentrated late in the network: the top 8 of 512 heads account for 17.7% of the total absolute attribution, and the mass per head is largest in layers 30 and 31.

4.5 Error geometry and the choice of reference encoder

Two further experiments check assumptions that analyses of I-JEPA usually leave implicit. First, the error covariance of the predictor is highly anisotropic: its largest eigenvalue is roughly 7,600 times the 1,024th, and its participation ratio is 137 out of 1,280 dimensions. It is therefore reasonable to ask whether MSE is a misleading summary. Using a held-out split, we found that the Mahalanobis distance and MSE rank images almost identically (Spearman 0.987) and rank the 50 classes similarly ($\rho = 0.957$, with 9 of the 10 hardest classes in common). Agreement decreases when more low-

variance components are included (0.794 with 1,024 components), consistent with estimation noise in the smallest sample eigenvalues being amplified by whitening (Appendix F). For class-level comparisons, MSE is adequate.

Second, the targets come from the EMA encoder, while most downstream uses rely on the online encoder. Linear CKA between the two, computed on the same images, stays above 0.997 up to block 15, declines to 0.70 to 0.76 over blocks 24 to 30, reaches 0.556 at block 31, and recovers to 0.936 at the final block. The choice of reference therefore matters mostly for analyses of late layers. In a small test with three diffusion-inpainted completions of the same masked region (30 images), I-JEPA's target embeddings of the alternative completions are much closer to each other than those of a Torchvision SWAG ViT-H/14 reference model (paired difference -0.29, 95% CI $[-0.37, -0.22]$, negative in 29 of 30 cases). The predictor, however, does not select the most context-consistent completion more often than chance (36.7% against 33.3%; Appendix G).

5 Discussion

Several of the results above would have been easy to get wrong. With the default non-strict loader, every number in §4 would have described a partially random network. Without mean and resample controls, the target-token effect in §4.2 could reasonably have been dismissed as an artifact of zero ablation. Without the positional prototype, the lens would have credited the first four predictor layers with image content that, in absolute terms, they do not yet carry. We built these checks into WorldModelLens as defaults because none of them is specific to I-JEPA.

The case study exercises only the step-level part of the interface, since I-JEPA has neither actions nor rollouts. The questions we consider most pressing for the field lie elsewhere: identifying which latent directions mediate the effect of an action in Dreamer, TD-MPC2, or latent-action models such as Genie; attributing the drift of an imagined trajectory to a particular step and component through intervention replay; training sparse dictionaries (Cunningham et al., 2023; Stevens et al., 2025) on the deterministic and stochastic parts of RSSM state and relating their features to reward and value heads; and testing whether internal signals anticipate prediction errors, as Aadithiyan (2026) suggest for DreamerV3. All of these can be expressed with the operations in §3.2. Progress on them would also benefit from shared checkpoints and tasks, so that findings can be compared across model families.

Limitations. Our large-scale evaluation covers a single JEPA checkpoint and a small MAE comparison. The action and rollout interfaces are implemented for the reinforcement-learning backends but have not yet been evaluated at the scale of this study, so the claim that analyses transfer across families rests on the interface and backend coverage rather than on results reported here. Attribution patching was validated on 4 image pairs (12 scored) and the MAE comparison uses 20 samples. The lens (240 images) and attribution-patching (13 images) experiments use small ImageNet subsets that may overlap with the pretraining data. Mask geometry differs between the attribution and intervention experiments. The lens is observational, and early-layer readings through the final head are lower bounds on the information present. Finally, MAE and I-JEPA differ in decoder design, masking ratio, and target normalization as well as in their objective.

6 Conclusion

We presented WorldModelLens, a library that brings the hook-and-cache approach of language-model interpretability to world models through a capability-typed interface and time-indexed intervention replay. Applied to the I-JEPA ViT-H/14 predictor, it exposed a silent weight-loading failure, showed that the prediction is carried by the target-token positions under three different ablation rules, located the depth at which predicted content surpasses a positional baseline, and clarified when attribution patching can substitute for exhaustive patching. The library and all experiment code are released to support further work on how world models represent the futures they predict.

AI use statement

Generative AI tools were used to help draft and edit the text of this manuscript and to write the script that produces Figures 2 and 3 from recorded results. They were not used to generate data,

formulate research questions, or produce experimental results; all reported numbers come from our experiment logs. The authors checked every number against the results files, verified that the plotting code reproduces the logged values, and take full responsibility for the content of the paper.

Ethics statement

This work analyzes publicly released models on the public ImageNet-1K dataset and involves no human subjects. The ambiguity experiment uses a latent diffusion model to inpaint masked regions of ImageNet images for evaluation only. We do not foresee direct harmful uses of an interpretability library.

Reproducibility statement

Anonymized code for WorldModelLens, including the coverage-checked loader (§3.4), all experiment scripts, image manifests, and random seeds, is included in the supplementary material. Model and data details are given in §4 and the statistical procedures in Appendix B. The lens results can be regenerated from a cached trace without a GPU, and the attribution-patching results were reproduced to three decimal places across reruns.

References

Donald Aadithiyan. Reading the room: Implicit confusion encoding in recurrent world model states. *arXiv preprint arXiv:2608.21582*, 2026.

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Advances in Neural Information Processing Systems*, 2024.

Christopher J. Anders, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Software for dataset-wide XAI: From local explanations to global insights with Zennit, CoRelAy, and ViRelAy. *arXiv preprint arXiv:2106.13200*, 2021.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995.

Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. SAELens. GitHub repository, 2024. URL https://github.com/decoderesearch/SAELens.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning*, 2024.

Thomas Bush, Stephen Chung, Usman Anwar, Adrià Garriga-Alonso, and David Krueger. Interpreting emergent planning in model-free reinforcement learning. In *International Conference on Learning Representations*, 2025.

Bhavith Chandra Challagundla, Sanskar Pandey, Param Thakkar, Rishikesh Mallagundla, Yugandhar Reddy Gogireddy, Wenhao Lu, et al. One lens, many worlds: A capability-typed interface for world-model interpretability. *arXiv preprint arXiv:2606.09936*, 2026.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, second edition, 1988.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, 2023.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Clément Dumas. nnterp: A standardized interface for mechanistic interpretability of transformers. In *Mechanistic Interpretability Workshop at Advances in Neural Information Processing Systems*, 2025.

Kenneth Eaton, Jonathan Balloch, Julia Kim, and Mark Riedl. The interpretability of codebooks in model-based reinforcement learning is limited. *arXiv preprint arXiv:2407.19532*, 2024.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979.

Parsa Esmati, Somjit Nath, Katja Hofmann, Derek Nowrouzezahrai, Samira Ebrahimi Kahou, and Majid Mirmehdi. The invisible hand of physics: When video diffusion models know more than they show. *arXiv preprint arXiv:2606.05328*, 2026.

Jaden Fiotto-Kaufman, Alexander R. Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Michael Ripa, Adam Belfki, Nikhil Prakash, Sumeet Multani, Carla Brodley, Arjun Guha, Jonathan Bell, Byron Wallace, and David Bau. NNsight and NDIF: Democratizing access to foundation model internals. *arXiv preprint arXiv:2407.14561*, 2024.

Quentin Garrido, Nicolas Ballas, Mahmoud Assran, Adrien Bardes, Laurent Najman, Michael Rabbat, Emmanuel Dupoux, and Yann LeCun. Intuitive physics understanding emerges from self-supervised pretraining on natural videos. *arXiv preprint arXiv:2502.11831*, 2025.

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

Wes Gurnee and Max Tegmark. Language models represent space and time. In *International Conference on Learning Representations*, 2024.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

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

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding and evaluating circuits. In *Conference on Language Modeling*, 2024.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

Erik Jenner, Shreyas Kapur, Vasil Georgiev, Cameron Allen, Scott Emmons, and Stuart Russell. Evidence of learned look-ahead in a chess-playing neural network. In *Advances in Neural Information Processing Systems*, 2024.

Daniel D. Johnson. Penzai + Treescope: A toolkit for interpreting, visualizing, and editing models as data. In *Mechanistic Interpretability Workshop at the International Conference on Machine Learning*, 2024.

Sonia Joseph, Praneet Suresh, Lorenz Hufe, Edward Stevinson, Robert Graham, Yash Vadi, Danilo Bzdok, Sebastian Lapuschkin, Lee Sharkey, and Blake Aaron Richards. Prisma: An open source toolkit for mechanistic interpretability in vision and video. *arXiv preprint arXiv:2504.19475*, 2025.

Sonia Joseph, Quentin Garrido, Randall Balestriero, Matthew Kowal, Thomas Fel, Shahab Bakhtiari, Blake Richards, and Michael Rabbat. Interpreting physics in video world models. In *International Conference on Machine Learning*, 2026.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *Conference on Language Modeling*, 2024.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, et al. Captum: A unified and generic model interpretability library for PyTorch. *arXiv preprint arXiv:2009.07896*, 2020.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components. *arXiv preprint arXiv:2403.00745*, 2024.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 2018.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *International Conference on Learning Representations*, 2023.

Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2:49–55, 1936.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023.

Neel Nanda. Attribution patching: Activation patching at industrial scale. Blog post, 2023. URL https://www.neelnanda.io/mechanistic-interpretability/attribution-patching.

Neel Nanda and Joseph Bloom. TransformerLens. GitHub repository, 2022. URL https://github.com/TransformerLensOrg/TransformerLens.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2023.

nostalgebraist. Interpreting GPT: The logit lens. LessWrong, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Jack Parker-Holder and Shlomi Fruchter. Genie 3: A new frontier for world models. Google DeepMind Blog, 2025. URL https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/.

Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, et al. Genie 2: A large-scale foundation world model. Google DeepMind Blog, 2024. URL https://deepmind.google/blog/genie-2-a-large-scale-foundation-world-model/.

Manoj Parmar. Safety, security, and cognitive risks in world models. *arXiv preprint arXiv:2604.01346*, 2026.

Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized input sampling for explanation of black-box models. In *British Machine Vision Conference*, 2018.

Samuele Punzo, Niccolò Caselli, Ippokratis Pantelidis, Francesco Massafra, Salvatore Lo Sardo, and Mohammadreza Salehi. How do video foundation models encode intuitive physics? Probing across pretraining paradigms. *arXiv preprint arXiv:2606.09646*, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. GAIA-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2017.

Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens van der Maaten. Revisiting weakly supervised pre-training of visual perception models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Interpretable and testable vision features via sparse autoencoders. *arXiv preprint arXiv:2502.06755*, 2025.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, 2017.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. *arXiv preprint arXiv:2310.10348*, 2023.

Mohammad Taufeeque, Philip Quirke, Maximilian Li, Chris Cundy, Aaron David Tucker, Adam Gleave, and Adrià Garriga-Alonso. Planning in a recurrent neural network that plays Sokoban. In *Mechanistic Interpretability Workshop at the International Conference on Machine Learning*, 2024.

Mohammad Taufeeque, Aaron David Tucker, Adam Gleave, and Adrià Garriga-Alonso. Path channels and plan extension kernels: A mechanistic description of planning in a Sokoban RNN. In *International Conference on Learning Representations*, 2026.

Keyon Vafa, Justin Y. Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. In *Advances in Neural Information Processing Systems*, 2024.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, 2025.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, 2020.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah Goodman, Christopher Manning, and Christopher Potts. pyvene: A library for understanding and improving PyTorch models via interventions. In *Conference of the North American Chapter of the Association for Computational Linguistics (System Demonstrations)*, 2024.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024.

Jewon Yeom, Hanseul Kim, Jeongjae Park, Sungmok Jung, Jaejin Lee, and Taesup Kim. What makes video world model latents action-relevant: Prediction over reconstruction. *arXiv preprint arXiv:2606.07687*, 2026.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024.

Xinyu Zhang. What do world models learn in RL? Probing latent representations in learned environment simulators. *arXiv preprint arXiv:2603.21546*, 2026.

A SUPPORTED BACKENDS

Table 3 lists the backends evaluated in this work and the optional heads each declares; Figure 1 places them within the overall architecture.

Table 3: Backends implemented in WorldModelLens and the optional heads each one declares in addition to the four core methods.

Backend	Latent state and dynamics	Optional heads
PlaNet	Continuous RSSM	decode, reward
DreamerV1	Continuous RSSM	decode, reward, actor, critic
DreamerV2	Categorical RSSM	decode, reward, continue, actor, critic
DreamerV3	Categorical RSSM	decode, reward, continue, actor, critic
IRIS	Transformer over codebook tokens	decode, reward, continue
TD-MPC2	Joint embedding	reward, actor, critic
I-JEPA	Joint embedding	none
Decision Transformer	Transformer	actor
Ha and Schmidhuber	VAE with MDN-RNN	decode, actor

B STATISTICAL PROCEDURES

Paired comparisons use two-sided paired t-tests and Wilcoxon signed-rank tests on per-image differences δ_i. We report $d = \bar{\delta}/s_\delta$ and follow the usual convention of reading $|d| \geq 0.2$, 0.5, and 0.8 as small, medium, and large. Confidence intervals are percentile bootstrap intervals with 1,000 resamples for the attribution experiments and 10,000 for the latent lens. For the lens we use cluster bootstraps over images and over images and masks, reusing the same resampled indices at every layer. Families of tests are corrected with the Benjamini-Hochberg procedure (attribution and layer sweeps) or the Holm procedure (lens increments); for the lens we additionally report a simultaneous sup-t band ($c = 2.85$ for 12 increments). With 12 tests and $B = 10{,}000$, the smallest Holm-adjusted bootstrap p-value that can be obtained is $12 \cdot 2/(B+1) \approx 2.4 \times 10^{-3}$, and all lens increments attain it.

C ATTENTION FAITHFULNESS BY LAYER

Table 4: Agreement between predictor attention and IG attributions at each layer ($N = 1{,}000$). Inversion is the fraction of images with $\rho < 0$; failure is the fraction with top-6 Jaccard overlap at most 0.3. The Wilcoxon test compares each layer's ρ with layer 0.

Layer	Spearman ρ [95% CI]	Top-6 Jaccard	Inversion	Failure	Wilcoxon p
0	0.267 [0.254, 0.280]	0.181	10.0%	75.8%	reference
1	0.275 [0.260, 0.289]	0.183	12.7%	74.6%	6.0×10^{-2}
2	0.229 [0.216, 0.242]	0.162	16.7%	79.8%	1.5×10^{-15}
3	0.266 [0.253, 0.279]	0.169	12.1%	78.9%	7.6×10^{-1}
4	0.236 [0.224, 0.248]	0.163	11.1%	80.2%	1.8×10^{-13}
5	0.219 [0.207, 0.232]	0.152	15.5%	81.8%	7.5×10^{-25}
6	0.226 [0.215, 0.237]	0.162	11.9%	81.1%	3.5×10^{-20}
7	0.242 [0.231, 0.254]	0.160	11.1%	81.6%	2.7×10^{-8}
8	0.215 [0.205, 0.226]	0.144	11.6%	87.4%	3.8×10^{-23}
9	0.249 [0.236, 0.261]	0.163	11.6%	82.2%	1.2×10^{-3}
10	0.215 [0.201, 0.227]	0.159	16.8%	82.8%	8.9×10^{-13}
11	0.186 [0.171, 0.199]	0.166	20.8%	80.3%	3.5×10^{-23}

Inversions are weakly associated with low-texture images: the point-biserial correlation with Laplacian variance is $r = -0.09$ ($p = 0.004$), which explains less than 1% of the variance. For MAE, attention and IG are nearly unrelated ($\rho = 0.06$, 45% inversions, top-K Jaccard 0.24 at decoder block 7; $N = 20$).

D Further replacement experiments

Donor class. On a subset of 132 images from 9 classes, we resampled the target tokens at predictor layer 6 using a donor image from the same class, from the nearest other class, or from the most distant class, with class distance measured by embedding similarity. The damage grows with donor distance: MSE is 0.683, 0.702, and 0.735, and probe accuracy is 100%, 89%, and 85%. Replacing the whole state instead of the target tokens magnifies the difference (probe accuracy 93%, 48%, and 22%). Since every resampled activation is one the network could produce, the difference reflects semantic compatibility rather than activation plausibility.

Whole-state replacement. Replacing the entire residual stream at one predictor layer raises MSE from 0.235 to between 0.85 and 1.38 (averages over the 12 layers for the three rules) and reduces the probe to chance for all rules. The same happens at each of the 32 encoder blocks, where mean replacement is almost as damaging as zeroing (MSE 1.211 versus 1.131), which indicates that encoder states are highly image-specific. The Mahalanobis distance of the resulting predictions follows the expected order (zero 48.7, mean 33.5, resample 19.4). Whole-state replacement removes all information at once and cannot resolve structure across depth; the attribution-patching results in §4.4 can.

E Attribution patching validation

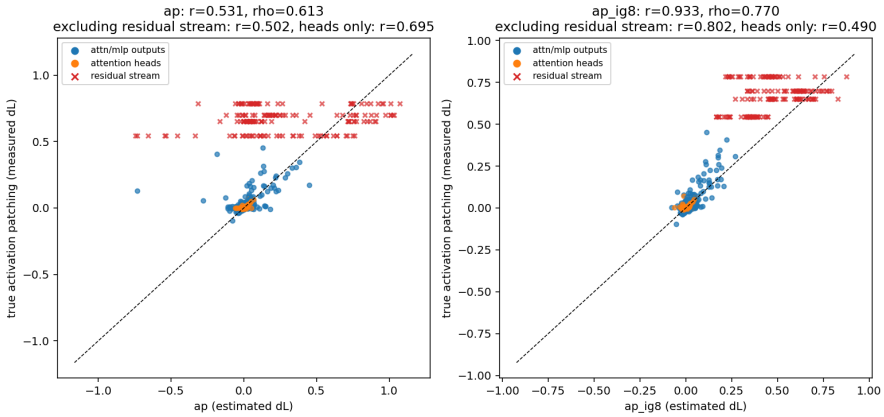

Figure 4: Estimated against exact patching effects for plain AP (left) and AP-IG (right). Red: residual stream; blue: attention and MLP block outputs; orange: attention heads.

Per site family, AP-IG correlates with exact patching at $r = 0.95$ for predictor attention outputs, 0.92 for predictor MLP outputs, 0.78 for encoder attention outputs, 0.76 for encoder MLP outputs, 0.49 for encoder heads, and 0.24 for the encoder residual stream. The heads used for validation are, by design, the 32 highest-scoring and 32 randomly chosen heads in each pair. For enc.resid_pre.0 the estimate (1.10) exceeds the largest possible true value of 1.0, as one would expect from a linear extrapolation of a saturating quantity. In the encoder residual stream, attributed importance decreases from layer 0 to layer 29 and increases again at layers 30 and 31. The top 32 heads carry 47.9% of the total absolute attribution, with a secondary concentration at layers 18 to 21. The most consistent heads across pairs include L31.8, L31.5, L31.3, L31.2, L31.0, L30.10, and L31.4, with ratios of mean to standard deviation of 2.80, 2.76, 2.33, 2.15, 1.73, 1.58, and 1.56.

F Prediction-error geometry

The 1,000 images are split into 500 calibration and 500 evaluation images, stratified by class. On the 8,000 patch-level error vectors of the calibration set, we fit a 256-component PCA of the prediction error against the EMA target, add a ridge of 10^{-4} times the mean eigenvalue, and score

evaluation errors by Mahalanobis distance (Mahalanobis, 1936; Lee et al., 2018). Subtracting the mean calibration error makes no difference (Pearson 0.9998), so the predictor has no large systematic bias. Between MSE and Mahalanobis distance, class ranks move by 3.2 positions on average and by at most 10. We treat class-specific patterns in the low-variance components as exploratory, because the smallest eigenvalues of a sample covariance are biased downward. Visual similarity does little to predict difficulty: the MSE difference between an image and its DINOv2 (Oquab et al., 2023) nearest neighbour is 0.91 times that between random pairs.

G EMA DIVERGENCE AND AMBIGUOUS COMPLETIONS

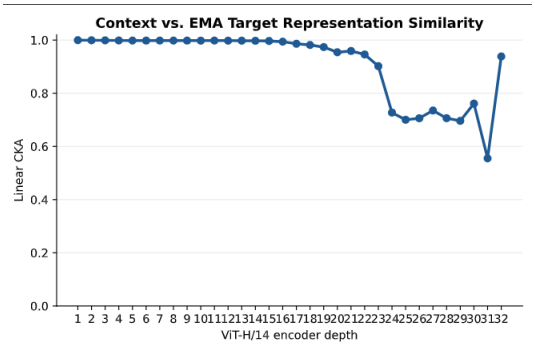

Figure 5: Linear CKA between the I-JEPA online encoder and its EMA target encoder at each block, computed on the same unmasked images ($N = 1{,}000$).

For 30 ImageNet images from different classes, we fixed the visible context and generated three plausible completions of a contiguous, manually reviewed mask of 4 to 9 patches with a latent diffusion inpainting model (Rombach et al., 2022) (30 steps, guidance scale 7.5, seed 42). For each model, we computed the mean pairwise cosine distance between the embeddings of the three completions in the masked region and divided it by the model's mean distance between random image pairs. This normalized spread is 0.058 for I-JEPA and 0.348 for the Torchvision SWAG ViT-H/14 reference (Singh et al., 2022) (paired difference -0.290, 95% CI $[-0.365, -0.219]$). The two models were not trained on the same data, so the difference cannot be attributed to the I-JEPA objective alone. The predictor chose the completion that annotators had marked as most consistent with the context in 11 of 30 cases (36.7%, 95% CI [20.0%, 53.1%]), which is not distinguishable from the chance rate of 33.3%.