CROSS-VIEW WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models enable agents to plan by imagining future states, but existing approaches operate from a single viewpoint, typically egocentric, even when other perspectives would make planning easier. We introduce Cross-View World Models (XVWM), trained with a cross-view prediction objective: given frames from one viewpoint, predict the future state from the same or a different viewpoint after an action is taken. Because the input and output views may share little or no visual overlap, the model must learn view-invariant representations of the environment's 3D structure and the agent's place in it. Trained on synchronized multi-view gameplay data from Aimlabs, the model localizes itself on an overhead map from egocentric frames alone and renders egocentric scenes from only a small marker on that map. Probing the diffusion transformer layer by layer pinpoints where these representations emerge and how actions transform them. In early layers, the encoding of the agent's position and heading depends on the input view; in the middle layers it converges onto a view-invariant code, where a linear probe trained on egocentric inputs decodes pose from bird's-eye inputs without retraining, and vice versa. Within this code, heading lies on a ring distributed across many units. When we inject a yaw action, the ring rotates rigidly by the commanded angle, while the heading encoded in the earlier, view-dependent layers is left untouched: the model applies the turn to its view-invariant code, not to the view-specific encoding that precedes it. Together, these results give a mechanistic account of the model's "cognitive map": how it tracks where it is and which way it is facing in a known environment. Such mechanistic understanding of how world models represent an agent's spatial state, and how actions transform it, is essential for reliable embodied agents.

1 INTRODUCTION

Good world models (Ha & Schmidhuber, 2018; LeCun, 2022) are faithful internal representations of an environment's dynamics, cheaper to query than acting in the real world. This allows agents to iterate on policy internally—that is, to *plan*—before taking actions, learning from imagined actions and their outcomes. Having an internal world model is paramount for robust, grounded agents, especially those that must interact with the physical world.

Despite tremendous progress in action-conditioned video generation for world modeling (Alonso et al., 2024; Bar et al., 2024; Decart et al., 2024; Valevski et al., 2024), current approaches predict from a single perspective, almost always egocentric. While such models may implicitly learn aspects of 3D structure, their training objective does not require it. The model can minimize its loss through patterns that hold only within that particular viewpoint.

Humans, too, see the world from an egocentric viewpoint. Yet we can shift to novel perspectives when we plan. For navigation, we can imagine a map of our environment and traverse it "from above." Problems often become trivial after reframing them in the right coordinate system. Vision is a malleable mode of thinking, and perspective shifts are integral to it.

When world models serve as internal planning engines, the egocentric viewpoint is not always the most efficient frame of reference. Consider navigation: current world models (Bar et al., 2024) plan by generating ensembles of autoregressive rollouts, requiring thousands of diffusion passes even in known environments, only to check how close each trajectory came to the goal. This is an inefficient abstraction. Mammalian brains solved navigation differently, using specialized cells

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.

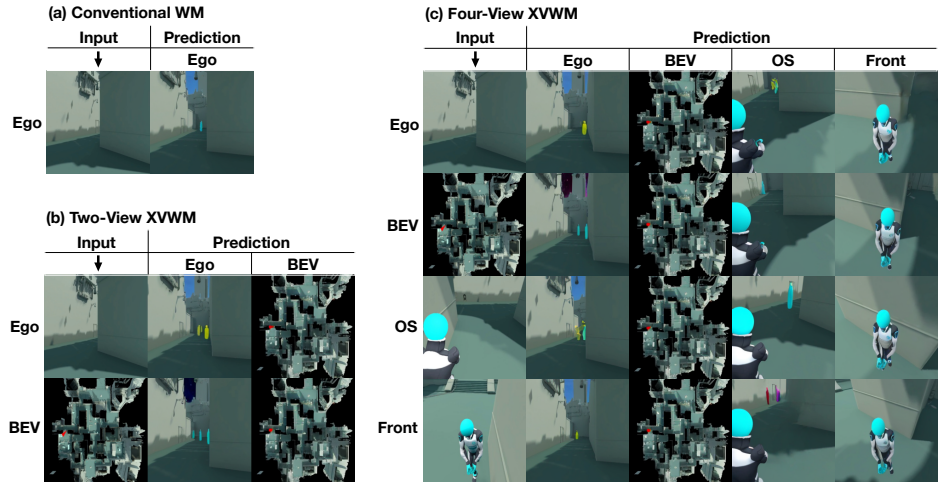


Figure 1: **Gaining multiple imagination streams.** (a) A conventional world model predicts future states from the same viewpoint. (b) An XVWM trained on egocentric and bird's-eye views learns bidirectional cross-view prediction: given either viewpoint as input, it can also imagine the future state from the other. These results yield a functional analog of biological cognitive maps: the model can locate and orient itself in its environment from visual cues alone. (c) Training on four viewpoints yields any-to-any prediction across all 16 input–output view pairs. Given context from a single viewpoint, XVWM can predict action-conditioned future states from all available perspectives. All predictions shown are 4 seconds into the future with identical state and action conditioning. Only the last of the 4 context frames is shown as input.

in the hippocampus that form a cognitive map and allow the animal to locate and track itself in space. Map-based navigation is more robust than the route-based planning that current world models perform.

We introduce Cross-View World Models (**XVWM**), trained with a cross-view prediction objective: given frames from one viewpoint as context along with an action, predict the future from any viewpoint in a fixed set (Fig. 1). This produces multiple parallel and consistent imagination streams, so that given the same context and action, the model can predict the consequences of those actions from any desired viewpoint. See our model in action[1]. Enforcing cross-view consistency acts as geometric regularization. This is the first use of cross-view prediction as a self-supervised objective for training world model dynamics. Unlike joint multi-view prediction, where the model observes all viewpoints simultaneously and need only maintain consistency between them, cross-view prediction requires transforming from one perspective to another. The input and output views may share little or no visual overlap, so pixel-level patterns do not transfer. To predict accurately, the model must learn view-invariant representations of the environment's 3D structure, capturing the true physical state underlying all perspectives.

We train on synchronized multi-view gameplay data from Aimlabs, a first-person shooter aim-training platform providing precisely aligned multi-camera recordings with high-frequency action labels. When one of the views is a bird's-eye view (BEV) perspective with a marker indicating the agent's location and orientation, cross-view training yields spatial grounding, and the model learns to locate itself in its environment from visual cues alone. We also find that cross-view prediction provides positive transfer to same-view prediction, when the chosen views provide complementary information. For instance, egocentric and BEV are complementary, as they capture details at very different scales.

Beyond planning, perspective-taking has benefits in multi-agent environments. Both predators stalking prey and prey evading predators benefit from modeling what others can see. Reasoning about

[1] https://streamable.com/d62q71

visibility and anticipating others' observations are plausible precursors to the theory of mind observed in humans and other primates (Baron-Cohen & Cross, 1992; de C. Hamilton et al., 2009; Flavell et al., 1981; Yang et al., 2015), and these are capacities that advanced AI agents will need to navigate the social world.

Our main contributions are as follows:

- We introduce cross-view prediction as a self-supervised objective for training world models, producing multiple parallel, switchable imagination streams from a single context–action pair.
- We show that cross-view training with a bird's-eye view yields spatial grounding in a known environment: sub-marker localization on the map from egocentric frames alone, detailed egocentric rendering from a $\sim$17px map marker, and positive transfer to same-view prediction.
- Probing the diffusion transformer layer by layer, we find a view-invariant linear code for the agent's allocentric position and heading in the middle layers: probes transfer zero-shot between viewpoints, heading lies on a distributed ring, and none of this structure exists in an untrained network.
- Injecting a yaw action rotates the ring rigidly by the commanded angle where the code is view-invariant, while leaving the early, view-specific heading encoding untouched: the model applies the turn to its view-invariant code, an action-equivariance that was never imposed.

2 Related work

World models for planning. World models learn environment dynamics to enable planning through imagination. The Dreamer family (Hafner et al., 2019; 2020; 2023) uses recurrent state-space models to learn latent dynamics and has achieved strong results across diverse domains. More recently, approaches based on action-conditioned video generation have emerged. Genie (Bruce et al., 2024) introduced generative interactive environments trained from unlabelled video, followed by models such as DIAMOND (Alonso et al., 2024), GameNGen (Valevski et al., 2024), and NWM (Bar et al., 2024). While these models generate high-fidelity future frames, they predict from a single, typically egocentric, perspective. Our work removes this restriction, allowing a single model to imagine from any viewpoint in its training set. We build on the conditional diffusion transformer architecture of NWM (Bar et al., 2024) and adapt it for cross-view prediction.

Multi-view representation learning. Several works leverage multiple viewpoints to learn better representations. Contrastive Multiview Coding (Tian et al., 2020) and related approaches (Caron et al., 2021; Chen et al., 2020; Grill et al., 2020; Zhang et al., 2025) use a contrastive objective to learn embeddings that capture information shared across views. Time-Contrastive Networks (Sermanet et al., 2018) use synchronized video for self-supervised imitation. Computer vision approaches have explored cross-view synthesis (Kulhánek et al., 2022; Regmi & Borji, 2018), though without action conditioning. The closest multi-view world models, MV-MWM (Seo et al., 2023), ReViWo (Pang et al., 2025) and Ctrl-World (Guo et al., 2025), are compared with XVWM in Appendix D.

Multi-view 3D representations. Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) and 3D Gaussian Splatting (Kerbl et al., 2023) learn to synthesize novel views from multi-view images. Recent works have integrated dynamics into these representations (Lu et al., 2025; Zhang et al., 2024), though they typically require explicit 3D supervision or incur high rendering costs. In autonomous driving, BEVFormer (Li et al., 2022) and Lift-Splat-Shoot (Philion & Fidler, 2020) fuse multiple views into bird's-eye-view representations using explicit geometric projection, requiring known camera intrinsics and extrinsics. XVWM learns geometric relationships implicitly through the prediction objective, without camera calibration.

Interpreting world models. Linear probes have recovered board states from sequence models trained on Othello (Li et al., 2023; Nanda et al., 2023), coordinates and dates from language models (Gurnee & Tegmark, 2024), and circular codes for periodic quantities such as days and months (Engels et al., 2025); such multidimensional features are increasingly studied as manifolds

rather than single directions (Bhalla et al., 2026), and Beckmann et al. (2026) give a mechanistic account of spatial world modelling in a navigation transformer. Closest to our analysis, Joseph et al. (2026) probe frozen video encoders and find a mid-depth zone in which physical quantities, among them a circular code for the direction of motion, become linearly decodable. Decodability alone can mislead: a probe can be accurate for a variable that the network never uses (Fadnavis et al., 2026; Hewitt & Liang, 2019). We therefore complement probing with an intervention on the action input, which tests whether the probed code responds to actions.

Spatial representations in neuroscience. We draw inspiration from the mammalian brain's ability to build rich spatial representations. Place cells and grid cells in the hippocampal-entorhinal system form cognitive maps that allow animals to locate and track themselves in space (Hafting et al., 2005; O'Keefe & Dostrovsky, 1971). We show that cross-view training with a bird's-eye view can yield analogous spatial grounding, where the model infers its global location and orientation from local visual cues alone.

3 RESULTS

Figure 1 shows the capabilities of XVWMs at a glance: given an input view and action, the model predicts future states from multiple viewpoints, producing parallel imagination streams. Note the consistency across columns: each output depicts the same underlying scene from a different perspective, suggesting the model has learned a coherent representation of the 3D environment.

We train CDiT-L world models on $4{,}186$ one-minute Aimlabs episodes recorded from four synchronized viewpoints (egocentric, bird's-eye, over-the-shoulder and front), subsampled to 5 fps and conditioned on the action $(\Delta x, \Delta y, \Delta\phi)$; a learnable view embedding selects the output view. We compare a Single-View (egocentric) baseline, a Two-View (egocentric + BEV) model and a Four-View model, all trained with identical compute (Appendix A). In brief (Appendices B and C): the Two-View model matches the same-view perceptual quality of the fully trained baseline on DreamSim while seeing a quarter of the ego→ego pairs, and outperforms the exposure-matched baseline on both LPIPS and DreamSim; the Four-View model, whose additional views are redundant or uninformative, is worse. Given a BEV trajectory as context, both models render detailed egocentric scenes from the $\sim$17px marker alone.

Table 1: Localization accuracy in bird's-eye view (BEV).

XVWM	Input View	Output View	Median Error (px)	Success @5px	Success @10px
Two-View	Ego	Bird's Eye	2.0	93.3%	98.6%
Two-View	Bird's Eye	Bird's Eye	1.7	96.7%	99.3%
Four-View	Ego	Bird's Eye	2.9	77.9%	95.2%
Four-View	Bird's Eye	Bird's Eye	2.6	81.8%	95.0%
Four-View	Over-shoulder	Bird's Eye	3.7	64.4%	88.0%
Four-View	Front	Bird's Eye	3.8	62.7%	86.4%

3.1 LOCALIZATION AND TRAJECTORY CONSISTENCY

As discussed in Appendix B, perceptual metrics like LPIPS and DreamSim are of limited use for BEV evaluation: the static map dominates the score, saturating both metrics. To isolate the signal of interest, we instead quantify marker localization—the predicted position of the agent marker compared to ground truth. We measure the median distance in pixels between the centroid of the ground-truth marker and the predicted marker across 1,257 test samples, for all input views targeting BEV (Table 1). For reference, the marker is $\sim$17px on the long side on 224×224px images, and all predictions reported in the table are for 4 seconds into the future. Across all models and input views, we observe sub-marker-size median error, indicating strong spatial grounding. The Two-View model definitively outperforms the Four-View model, consistent with both, having seen $2\times$ more BEV-targeted pairs during training, and being only exposed to high quality views, unlike the Four-View model. Success@k denotes the percentage of predictions with error below k pixels.

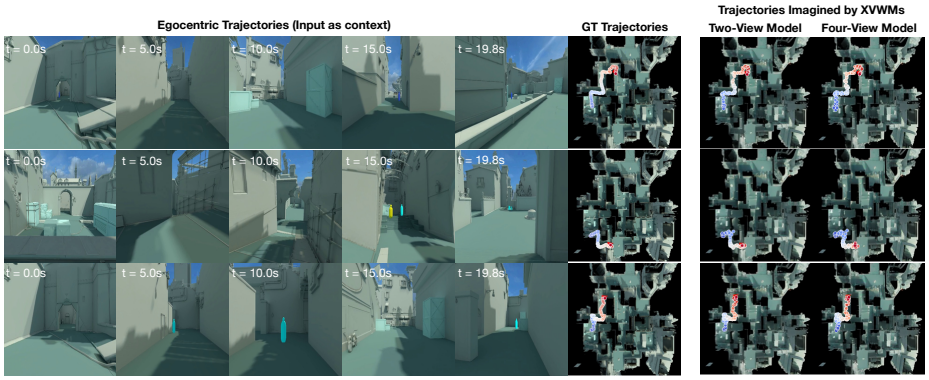


Figure 2: **XVWM's internal geolocalization.** Localization in an environment is but one component of the mammalian cognitive map. Localization alone is insufficient for navigation, one also needs to encode consistent spatial ordering. Here, we probe this property by following three egocentric trajectories and imagining the corresponding path from BEV. Comparing to ground-truth trajectories, we observe that movements in the egocentric view translate to consistent movements on the BEV map. The blue-to-red gradient denotes time increasing from 0.0s to 20.0s. At each point, the model predicts the next frame, i.e., it predicts 0.2s into the future. The model possesses an "internal GPS" that allows it to locate and orient itself in its environment from visual cues alone.

Going beyond localization, a useful cognitive map must also capture spatial relationships between locations. Localization alone, even at sub-marker accuracy, is not sufficient. Fig. 2 compares three egocentric trajectories taken by the agent against the BEV trajectories imagined by the model. The agreement with ground truth is striking, confirming that the model maintains a functioning internal "GPS." Example videos[2,3] show the agent following an egocentric path while imagining the corresponding path from BEV. Accurately detecting orientation for a marker of this size is inherently noisy; however, the orientational accuracy can be gauged from the demo video[1]. Orientation plays an even larger role in the inverse task of spawning, which we discuss in Appendix C.

4 Probing the View-Invariant Representation

The preceding sections measured what the Two-View model can render from either view. We now ask what it represents. If predicting one view from another forces the network to build a view-invariant description of the scene and of the agent's place in it, that description should contain the agent's pose, its allocentric position (x, y) and heading θ, in a form that can be read out linearly, that is the same whether the context frames come from the ego view or the bird's-eye view, and that moves when the agent is commanded to turn. We test this with linear probes (Sec. 4.2) and with an intervention on the action input (Sec. 4.3).

4.1 Setup

For each pose we run the model on the four context frames of one view and record, after each of the 24 transformer blocks (indexed 0–23), the mean of the 196 target-view tokens, a single 1024-dimensional vector per block. States are recorded at the first denoising step ($t = 999$), where the target latent is pure noise, so they depend only on the context frames and the action. The target view is always the ego view; the context view is either the ego view (*ego-context* states) or the bird's-eye view (*BEV-context* states). With a zero action we record 1000 poses from 416 episodes under both context views.

A probe is a ridge regression ($\alpha = 1$) from the state of one block to $(\cos\theta, \sin\theta)$ or to (x, y). We report $R^2 = 1 - \mathrm{SS}_{\mathrm{res}}/\mathrm{SS}_{\mathrm{tot}}$ on held-out poses (1 is perfect, 0 is no better than predicting the mean,

[2] https://streamable.com/t1o1ym
[3] https://streamable.com/llv6z5

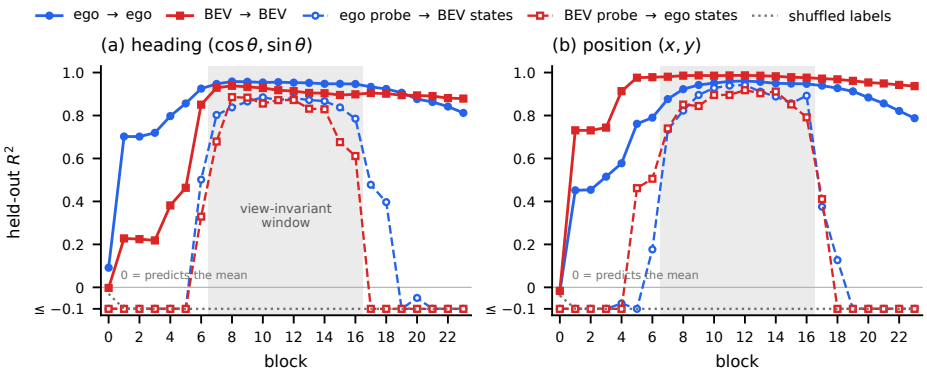


Figure 3: **A linear readout of the agent's pose becomes view-invariant in blocks 7–16.** Held-out R^2 of ridge probes that decode (a) heading, as $(\cos\theta, \sin\theta)$, and (b) allocentric position (x, y) from the hidden state after each block, for the same poses seen with an ego-view (blue) or bird's-eye (red) context. $R^2 = 1 - \mathrm{SS}_{\mathrm{res}}/\mathrm{SS}_{\mathrm{tot}}$ on held-out poses: 1 is perfect, 0 is no better than predicting the mean, negative is worse; values below -0.1 are drawn at -0.1. Solid: probe fit and tested on the same view. Dashed: probe fit on one view and applied, without refitting, to the states of the other view. Dotted: shuffled-label control. Cross-validation is grouped by episode. In blocks 1–5 pose is decodable from each view, yet probes transfer poorly or not at all across views; from block 7 a probe fit on either view decodes the other (ego-fit probe on BEV states: heading 0.79–0.88, position 0.73–0.94), and transfer collapses again from block 17.

negative is worse) under five-fold cross-validation grouped by episode, so that no test pose shares an episode with a training pose. To measure cross-view transfer we take the probe fit on ego-context states and apply it, without refitting, to the BEV-context states of the same held-out poses, and vice versa. Probes fit on shuffled labels serve as a control.

For the intervention we take 200 further poses from 173 episodes and add a yaw command $\Delta\psi \in \{0, \pm 15°, \pm 22.5°, \pm 30°, \pm 45°\}$ to the action input, with zero displacement and a one-frame prediction offset, again under both context views. These states are read with a *frozen* heading probe for each block, fit on the 560 zero-action poses whose episodes are disjoint from the 173 intervention episodes, so the probe never sees a rotated state or any pose that appears in the plots. The decoded heading is $\hat{\theta} = \mathrm{atan2}$ of the probe output, the rotation of a pose is $\hat{\theta}(\Delta\psi) - \hat{\theta}(0)$, and the *gain* of a block is the slope of the circular mean of this rotation against $\Delta\psi$ over the nine commands.

Two limits should be kept in mind. Figures 4 and 5a plot states in the probe's own two-dimensional readout plane, where a ring of held-out poses is the same fact as a high R^2; these plots illustrate the linear code, they do not add evidence for it. And the intervention changes the action input and reads the representation: it establishes that the probed code responds to the action, an equivariance a static, action-blind readout would not show, without yet establishing that the model uses this code to produce its images.

4.2 A SHARED LINEAR CODE FOR POSITION AND HEADING

Figure 3 shows the held-out R^2 of the heading and position probes at every block. In blocks 1–5 the pose can already be decoded from each view on its own: ego-context states give a heading R^2 of 0.70–0.86, and BEV-context states give a position R^2 of 0.73–0.98, as one would expect from a frame that shows the agent's marker on the map. But a probe fit on one view fails on the other. Cross-view R^2 is negative for heading in both directions and for position in all but one case (0.46 for the BEV-fit probe on ego-context states at block 5). The early code is specific to the view. From block 7 the same probe decodes both views. An ego-fit probe reads heading from BEV-context states with R^2 0.79–0.88 and position with 0.73–0.94 through block 16; a BEV-fit probe reads heading from ego-context states with 0.61–0.89 and position with 0.74–0.92. At block 12 the transferred probes keep, on average over the two directions, 94% (heading) and 96% (position) of the same-view score, and the median heading error is $4°$ within a view and $7°$ across views. From block 17 the cross-

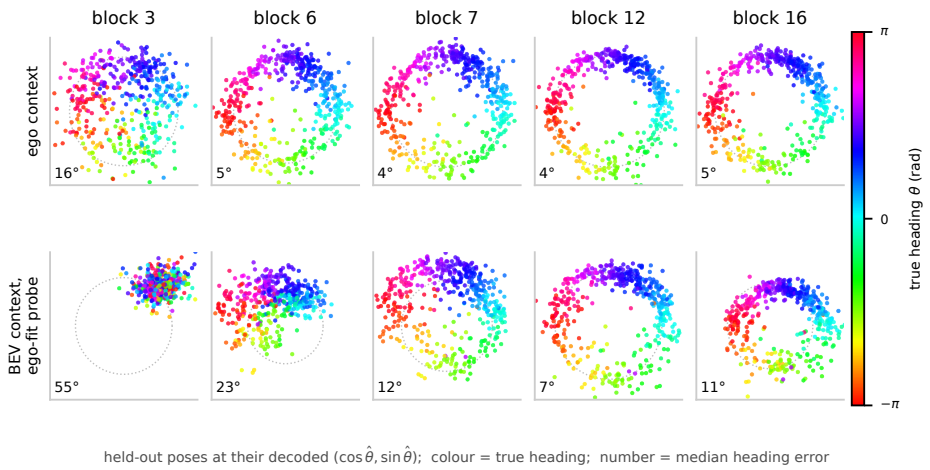


Figure 4: **The heading code is a ring shared by both views.** Held-out poses plotted at the heading decoded by the ego-fit probe, $(\cos\hat{\theta}, \sin\hat{\theta})$ (unit circle: perfect decoding), coloured by their true heading. Top: ego-view context. Bottom: bird's-eye context, read with the same ego-fit probe, which never saw a BEV state. In block 3 the ego states already form a ring ordered by heading (median error $16°$) while the BEV states do not ($55°$); from block 7 both views fall on the same ring in the same order ($4°$ and $12°$ at block 7; $4°$ and $7°$ at block 12). Numbers: median heading error.

view score collapses while the same-view scores stay at 0.79–0.97; the representation appears to re-specialise towards the ego output the network has to render. Shuffled-label probes stay at zero, and an untrained network of the same architecture shows neither a usable code nor any transfer (not shown). We refer to blocks 7–16 as the *view-invariant window*: there the network holds a single linear code for the agent's position and heading, independent of the view it was given.

This code is spread over many channels. The weights of the frozen heading probe have a participation ratio of 376 out of 1024 channels at block 7 and 448 at block 12; refitting the probe without its two most-weighted channels leaves the held-out R^2 unchanged (0.942 against 0.941 at block 7, 0.949 against 0.949 at block 12); and probes restricted to a random quarter of the channels still reach 0.90–0.93 across the window. Two channels with unusually large variance do account for most of the *variance* of the readout in blocks 6–9 (75% at block 7, below 1% from block 11 on), which is why an unsupervised PCA of the states shows a heading ring only in those blocks, but the code does not depend on them.

Since the heading probe predicts $(\cos\theta, \sin\theta)$, its two outputs define a plane in which perfectly decoded poses lie on the unit circle. Figure 4 plots held-out poses in this plane for the ego-fit probe. Ego-context states form a ring ordered by heading from the early blocks on, with a median error of $16°$ at block 3 and 4–$5°$ from block 7. BEV-context states, read with the same ego-fit probe, are unordered at block 3 ($55°$), partly ordered at block 6 ($23°$), and from block 7 lie on the same ring in the same order ($12°$ at block 7, $7°$ at block 12, $11°$ at block 16). The two context views thus share not only a decodable heading but the same two-dimensional readout of it. As noted above, the roundness of the rings restates the R^2 of Fig. 3; what Fig. 4 adds is that the ego and BEV rings coincide, which is what the intervention below relies on.

4.3 A YAW ACTION ROTATES THE HEADING CODE

If this ring is the model's heading code, an action that turns the agent should move states along it. Figure 5 shows what a yaw command does to the frozen heading readout. In blocks 0–5 nothing moves. The gain is 0.00–0.02, even though heading is decodable in blocks 1–5: the frozen probe reads the unperturbed states with a median error of 9–$16°$, and after a $45°$ command it still reports the original heading, $45°$ away from the commanded one. Block 6 is transitional. Its gain is 0.38, and the cloud deforms rather than turns; the best affine map from the $\Delta\psi = 0$ readout to the $\pm 45°$ readout is an $18°$ rotation combined with a $2.2\times$ anisotropic stretch. From block 7 the ring turns

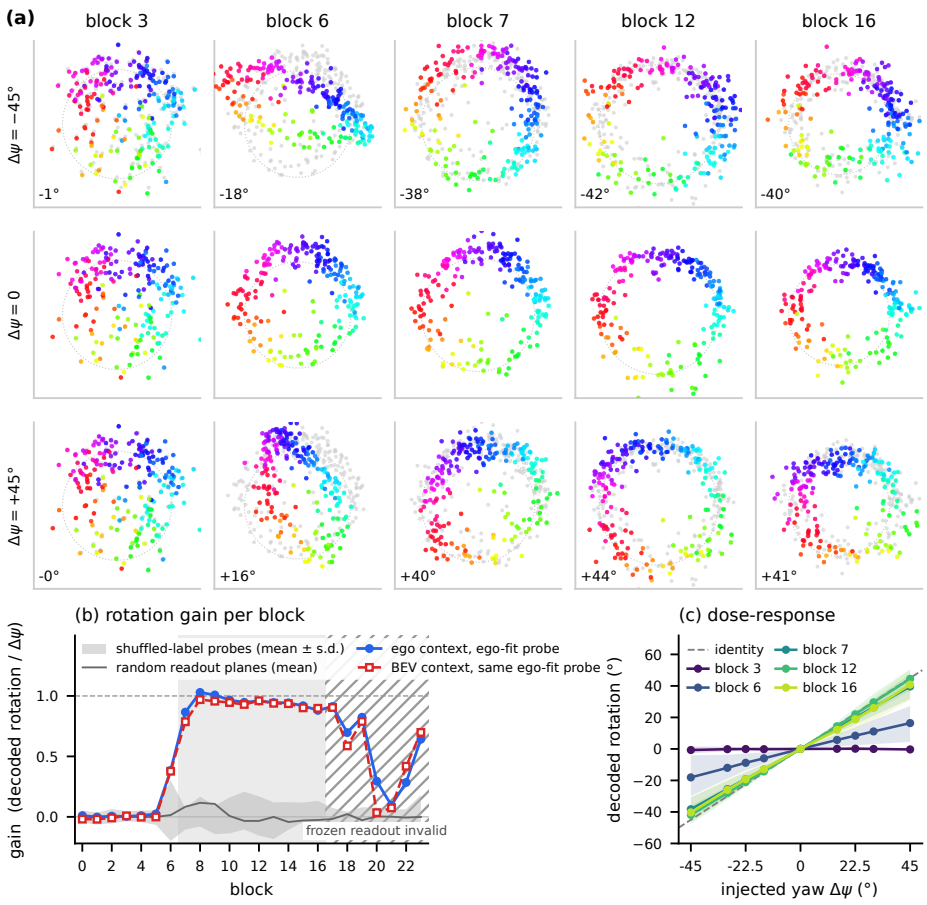


Figure 5: **A yaw action rotates the heading readout by the commanded angle, but only where the code is view-invariant.** A yaw command $\Delta\psi$ is added to the action input for 200 poses and the resulting states are read with a frozen probe fit on zero-action poses from other episodes. (a) Readout plane at $\Delta\psi = -45°$, 0, $+45°$ (rows) for five blocks (columns), colour = true heading at $\Delta\psi = 0$; grey: the $\Delta\psi = 0$ ring; number: mean decoded rotation. Block 3: no change. Block 6: a deformation rather than a rotation (best affine map: $18°$ rotation with a $2.2\times$ anisotropic stretch). Blocks 7–16: the ring turns as a whole (41–$47°$ for a $45°$ command, anisotropy ≤ 1.2). (b) Gain, the decoded rotation per unit $\Delta\psi$, per block, for ego-context states (blue) and for BEV-context states read with the same ego-fit probe (red); grey: shuffled-label probes and random readout planes; hatched: blocks where the frozen probe no longer decodes the unperturbed states ($R^2 < 0.8$ at $\Delta\psi = 0$; any threshold between 0.75 and 0.89 selects the same blocks). (c) Decoded rotation against the commanded angle for the nine values of $\Delta\psi$ (circular mean and interquartile range over poses); dashed: identity. Supplementary Movie 1 shows the intervention in every block for ego-context states, and Supplementary Movie 2 for BEV-context states plotted in the same ego-fit readout plane.

as a whole. The gain is 0.87 at block 7 and 0.88–1.03 in blocks 8–16, the affine map is a rotation by 41–$47°$ with anisotropy 1.03–1.18, and the error of predicting each pose's readout by rotating its $\Delta\psi = 0$ readout by $\Delta\psi$ (6–$10°$) is no larger than the decoding error itself (5–$9°$). Within the resolution of the readout the rotation is rigid, and it is linear and symmetric over the whole $\pm 45°$ range (Fig. 5c). BEV-context states behave the same way: read with the same ego-fit probe, they rotate with gain 0.79 at block 7 and 0.90–0.97 in blocks 8–16 (Fig. 5b, red). Twenty probes fit on shuffled labels and twenty random readout planes give gains within ± 0.12 of zero at every block, so the rotation belongs to the heading code and not to an arbitrary two-dimensional projection of the state. The rotation thus appears at the depth where the two views come to share a code, and not before: the early blocks encode heading, but the action does not act on that encoding. Supplementary

Movie 1 shows the intervention in every block for ego-context states; Supplementary Movie 2 shows the cross-view case, BEV-context states plotted in the same ego-fit readout plane.

From block 17 the frozen ego-fit probe no longer decodes even the unperturbed states (R^2 at $\Delta\psi = 0$ falls from 0.89 at block 16 to between -0.77 and 0.77 in blocks 17–23), so its gain is not interpretable there; these blocks are hatched in Fig. 5b. We attribute this to a mismatch between the condition the probe was fit in (zero prediction offset, where the ego target is the last context frame) and the intervention condition (one-frame offset), which the ego stream resolves differently in its last blocks. A BEV-fit probe applied to BEV-context states is not affected: it remains valid (R^2 0.81–0.88 at $\Delta\psi = 0$) and keeps rotating with gain 0.79–0.88 through block 23.

Figures 3–5 thus show that the network builds and maintains a view-invariant code for the agent's position and heading from whichever view it is given, and that a yaw action rotates the heading within that code by the commanded angle.

5 Conclusion and Future Directions

Cross-view prediction is an effective self-supervised signal for world models. Training on complementary views transfers to same-view prediction and yields capabilities a same-view model lacks: sub-marker localization from egocentric frames, consistent imagined trajectories over 20-second rollouts, and detailed egocentric scenes generated from a $\sim$17px marker that specifies only position and orientation.

Probing the network shows how these capabilities are carried. The agent's position and heading are linearly decodable from the early blocks, but in a form specific to the input view; in blocks 7–16 both views converge onto one linear code, in which heading lies on a ring distributed over hundreds of channels. A yaw command rotates this ring by the commanded angle, from either context view, and leaves the early view-specific encoding untouched. The network thus maintains a view-invariant code for its own pose and applies the action to that code rather than to the encoding that precedes it, an action-equivariance the training objective never imposed.

This gives the cognitive-map analogy that motivated this work a mechanistic footing, though it remains incomplete. We have shown that the heading code is decodable and action-equivariant. While we probe the model in a known environment, the transfer of the learned structure to unseen environments, and the persistence of memory beyond the context window, which biological maps achieve by being built incrementally through experience, remain open questions.

Finally, imagining how the world appears from a viewpoint one does not occupy is a plausible precursor of perspective-taking and theory of mind; whether cross-view world models can support it in multi-agent settings remains an open question.

AI use statement

We used LLM-based assistants to help write and debug analysis code, to search for and summarize related literature, and to draft and edit parts of the text and figures. All results and text were checked by the authors.

Reproducibility statement

The dataset, architecture, and training schedule are given in Appendix A; the probing and yaw-injection protocols, data splits, and controls in the probing section. All interpretability analyses run on CPU from stored hidden states; analysis code will be released with the camera-ready version.

References

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 *Thirty-eighth Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.12399.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models, 2024. URL https://arxiv.org/abs/2412.03572.

Simon Baron-Cohen and Pippa Cross. Reading the eyes: Evidence for the role of perception in the development of a theory of mind. *Mind & Language*, 7(1-2):172–186, 1992. doi: https://doi.org/10.1111/j.1468-0017.1992.tb00203.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0017.1992.tb00203.x.

Pierre Beckmann, Matthieu Queloz, and Andre Freitas. World modeling in transformers, 2026. URL https://arxiv.org/abs/2609.21748.

Usha Bhalla, Thomas Fel, Can Rager, Sheridan Feucht, Tal Haklay, Daniel Wurgaft, Siddharth Boppana, Matthew Kowal, Vasudev Shyam, Jack Merullo, Atticus Geiger, and Ekdeep Singh Lubana. Do sparse autoencoders capture concept manifolds?, 2026. URL https://arxiv.org/abs/2604.28119.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments, 2021. URL https://arxiv.org/abs/2006.09882.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. URL https://arxiv.org/abs/2002.05709.

Antonia F. de C. Hamilton, Rachel Brindley, and Uta Frith. Visual perspective taking impairment in children with autistic spectrum disorder. *Cognition*, 113(1):37–44, 2009. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2009.07.007. URL https://www.sciencedirect.com/science/article/pii/S0010027709001668.

Decart, Julian Quevedo, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. 2024. URL https://oasis-model.github.io/.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

Shreyas Fadnavis, Praitayini Kanakaraj, and Felix Wyss. When and how long? the readout-mediator angle in temporal reasoning, 2026. URL https://arxiv.org/abs/2605.29126.

John H. Flavell, Barbara A. Everett, Karen Croft, and Eleanor R. Flavell. Young children's knowledge about visual perception: Further evidence for the level 1–level 2 distinction. *Developmental Psychology*, 17(1):99–103, 1981. doi: 10.1037/0012-1649.17.1.99.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data, 2023. URL https://arxiv.org/abs/2306.09344.

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, 2020. URL https://arxiv.org/abs/2006.07733.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation, 2025. URL https://arxiv.org/abs/2510.10125.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jE8xbmvFin.

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. *arXiv preprint arXiv:1912.01603*, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, August 2005. ISSN 1476-4687. doi: 10.1038/nature03721. URL https://doi.org/10.1038/nature03721.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks, 2019. URL https://arxiv.org/abs/1909.03368.

Sonia Joseph, Quentin Garrido, Randall Balestriero, Matthew Kowal, Thomas Fel, Shahab Bakhtiari, Blake Aaron Richards, and Michael Rabbat. Interpreting physics in video world models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=aijGVmEG9Y.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Jonáš Kulhánek, Erik Derner, Torsten Sattler, and Robert Babuška. Viewformer: Nerf-free neural rendering from few images using transformers. In *European Conference on Computer Vision*, pp. 198–216. Springer, 2022.

Yann LeCun. A path towards autonomous machine intelligence. *OpenReview*, 2022.

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 *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DeG07_TcZvT.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers.(2022). *URL https://arxiv. org/abs/2203.17270*, 2022.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9263–9274, 2025.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. URL https://arxiv.org/abs/2003.08934.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2. URL https://aclanthology.org/2023.blackboxnlp-1.2/.

J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175, 1971. ISSN 0006-8993. doi: https://doi.org/10.1016/0006-8993(71)90358-1. URL https://www.sciencedirect.com/science/article/pii/0006899371903581.

Jing-Cheng Pang, Nan Tang, Kaiyuan Li, Yuting Tang, Xin-Qiang Cai, Zhen-Yu Zhang, Gang Niu, Masashi Sugiyama, and Yang Yu. Learning view-invariant world models for visual robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vJwjWyt4Ed.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision*, pp. 194–210. Springer, 2020.

Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3501–3510, 2018.

Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In *International Conference on Machine Learning*, pp. 30613–30632. PMLR, 2023.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pp. 776–794. Springer, 2020.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines, 2024. URL https://arxiv.org/abs/2408.14837.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization, 2019. URL https://arxiv.org/abs/1911.07013.

Daniel Y.-J. Yang, Gabriela Rosenblau, Cara Keifer, and Kevin A. Pelphrey. An integrative neural model of social perception, action observation, and theory of mind. *Neuroscience & Biobehavioral Reviews*, 51:263–275, 2015. ISSN 0149-7634. doi: https://doi.org/10.1016/j.neubiorev.2015.01.020. URL https://www.sciencedirect.com/science/article/pii/S0149763415000317.

Guanming Zhang, David Heeger, and Stefano Martiniani. Contrastive self-supervised learning as neural manifold packing. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=eTgXolhWCH.

Mingtong Zhang, Kaifeng Zhang, and Yunzhu Li. Dynamic 3d gaussian tracking for graph-based neural dynamics modeling. *arXiv preprint arXiv:2410.18912*, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

A Dataset and model details

Dataset. We use $4{,}186$ one-minute gameplay (first-person shooter) episodes collected at 60 frames per second and subsampled to 5 frames per second. Images are 224×224px. Each frame is paired with the action taken at that timestep. Every episode is recorded from four perspectives: egocentric[4], bird's-eye view (BEV)[5], over-the-shoulder (OS)[6], and front-facing (Front)[7] (see Fig. 1 for stills from each). OS and Front follow the agent from a fixed distance, whereas BEV is a fixed overhead view that marks the agent's position and facing direction with a red elongated triangle. Owing to the tactical nature of the game, colored targets occasionally appear and change color upon being hit. The sky varies across episodes, ranging from bright daytime to nighttime and stylized themes such as space. Notably, the BEV provides no sky context, which we later exploit as a natural test of generalization. In this work, we condition on translational movement and change in yaw $(\Delta x, \Delta y, \Delta\phi)$, and do not aim to capture shooting dynamics. We use a 90:10 train-test split.

Architecture: Our method is architecture-agnostic: any action-conditioned video generation model can in principle be extended with viewpoint conditioning. We choose to build on the Conditional Diffusion Transformer (CDiT) from Navigation World Models (NWM) (Bar et al., 2024), which was shown to outperform many of its predecessors. We add a learnable view embedding table that maps each view ID to a dense vector of the same dimension as the model's hidden state. This view embedding is added to the existing conditioning signal – comprising the diffusion timestep, relative time, and action embeddings – which is injected into each transformer block via adaptive layer normalization (Peebles & Xie, 2023; Xu et al., 2019). The model takes 4 consecutive frames from any single viewpoint as context (amounting to a 0.8 s context window at 5 FPS). Following NWM, the target frame can lie in either the future or the past relative to the context: during training, the prediction time horizon extends to 64 frames (12.8 s) into the future or into the past, while at test time it only predicts up to 25 frames (4 s) into the future.

Training and Compute: We train three models:

Single-View baseline: Trained exclusively on egocentric data with standard same-view prediction. This serves as our baseline.

Two-View XVWM (cognitive-map-inspired): Trained on egocentric and bird's-eye views. At each training step, the input view is sampled uniformly at random. The output view is then sampled independently, with a cross-view probability of 0.5: half the time the target matches the input view, half the time it differs. This yields all four input–output combinations (ego→ego, ego→BEV, BEV→ego, BEV→BEV) with equal probability 0.25.

Four-View XVWM (generalist): Trained on all four views. At each training step, the input view is sampled uniformly at random. With probability 0.5, the output view matches the input (same-view prediction); otherwise, the output view is sampled uniformly from the remaining three views (cross-view prediction). This yields same-view pairs at 12.5% each and cross-view pairs at approximately 4.17% each.

The cross-view probability is a design choice which can be further optimized. Same-view prediction is necessary for autoregressive rollouts, but one can weight perspectives according to how informative they are or how much the downstream task depends on a particular viewpoint.

All three models are trained on $4\times$ H100/H200 GPUs with an effective batch size of 64 for 73 epochs, using the CDiT-L architecture.

B Perceptual similarity and transfer

Our results suggest that the models not only learn to transform between views, but also learn to transform across scales. Fast movements in the egocentric view must translate to slow movements of the marker on the BEV map, and vice versa. Similarly, tiny angular changes seen from above can translate to the entire field of view shifting drastically at eye level. We also investigate the importance

[4] https://streamable.com/nutp36
[5] https://streamable.com/ahaxwo
[6] https://streamable.com/bkecg2
[7] https://streamable.com/e09f2w

of complementary, information-rich views for positive transfer, and show how redundant or visually impoverished views can degrade performance. Together, these experiments probe how cross-view training yields a coherent internal representation of 3D space.

We begin by systematically probing the same-view and cross-view generation quality for all three models, using LPIPS (Zhang et al., 2018) and DreamSim (Fu et al., 2023) as perceptual metrics. We find that static elements—namely the fixed map in BEV, and the always-present agent body in OS and Front—saturate these scores to different extents, making comparison across views unreliable (see Fig. 8). We thus limit perceptual quality analysis to ego as the target view for both same-view and cross-view predictions. The LPIPS score for the Single-View model is comparable to that reported for NWM (CDiT-L) (Bar et al., 2024) trained on RECON, and serves as our baseline.

Starting with same-view predictions: Fig. 6(a) reveals that even after seeing dramatically fewer ego→ego pairs during training, the Two-View XVWM matches the perceptual quality of the fully trained baseline on DreamSim, strongly suggesting positive transfer from cross-view to same-view prediction. The Four-View model, on the other hand, suffers some degradation. Given the dramatically fewer samples seen by both models, it is important to systematically explore transfer by normalizing for training exposure.

We investigate this by comparing the multi-view models against the Single-View baseline at matched training exposure (Fig. 6(b,c)). The Two-View model (egocentric + BEV) consistently outperforms the exposure-matched baseline on both metrics, confirming positive transfer from cross-view training. We attribute this to the complementary information content of the two views: egocentric provides rich local context at eye level, while BEV provides a global map with precise agent location and orientation. Together, they offer more learning signal than either alone.

The Four-View model, which adds over-the-shoulder and front-facing views, performs consistently worse than the Two-View baseline despite similar training budgets. This may still be a worthwhile trade-off given the two additional imagination streams, but the degradation warrants explanation. We attribute it to differences in view informativeness. The over-the-shoulder view overlaps substantially with the egocentric perspective and tilts slightly toward the ground, providing limited additional signal. The front-facing view is more problematic: it faces the agent and angles downward, devoting much of the frame to the monotonic ground plane. In tactical scenarios where the agent moves along walls, this view collapses to a near-uniform surface, stripping away the environmental context needed to disambiguate one location from another.

These results suggest that information content of the views matters more than just view count. Implementing XVWMs with complementary views—such as egocentric and BEV—is key to positive transfer. Adding redundant or low-information views can dilute the learning signal and even degrade performance.

Our analysis is consistent when considering cross-view performance. The Two-View model again decisively outperforms the Four-View model (Fig. 7), with over-the-shoulder and front being the primary underperformers. However, to properly normalize for training exposure, one would need base models trained on each view independently, analogous to what we did for same-view prediction. An analogous plot depicting LPIPS scores can be found in Fig. 9, which follows the same trend.

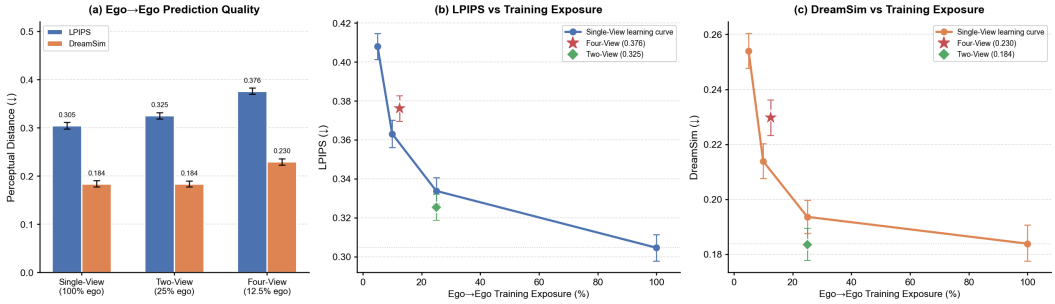


Figure 6: **Same-view prediction quality and transfer. (a)** Ego→ego perceptual similarity (LPIPS and DreamSim, lower is better) across the three models. The Single-View baseline sees 100% egocentric pairs, the Two-View model 25%, and the Four-View model 12.5%. All models are trained with identical compute (same steps, epochs, and batch size). **(b, c)** Ego→ego quality plotted against training exposure. Blue/orange curves show the Single-View model at intermediate checkpoints; starred and diamond markers show the fully trained Four-View and Two-View XVWMs at their effective exposures. Metrics are computed for a 4-second prediction target, averaged over 1,257 test samples (3 predictions per test episode). Error bars show bootstrap 95% confidence intervals.

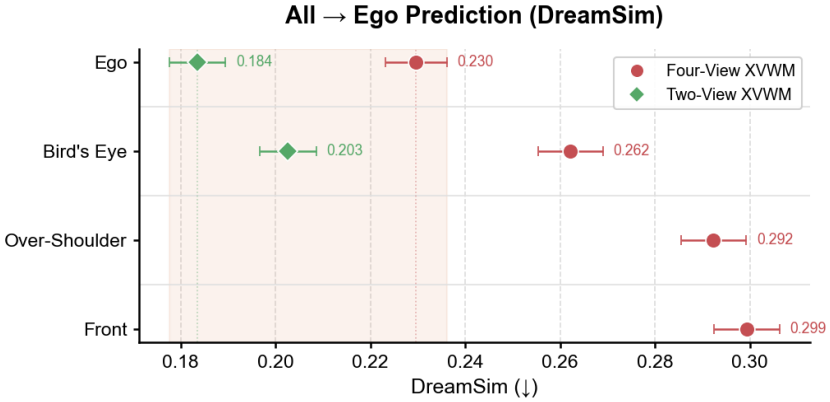


Figure 7: **Cross-view prediction quality (all→ego) within and across XVWM models.** The Two-View model, trained on complementary, high-signal views, outperforms the Four-View model. This is partly explained by the Two-View model's higher exposure (25% across all pairs) compared to the Four-View model (12.5% and 4.17% for same-view and cross-view pairs respectively). However, even within the Four-View model, OS and Front perform considerably worse than BEV at similar exposure levels. Metrics are computed for a 4-second prediction target, averaged over 1,257 test samples (3 predictions per test episode). Error bars depict bootstrap 95% confidence intervals.

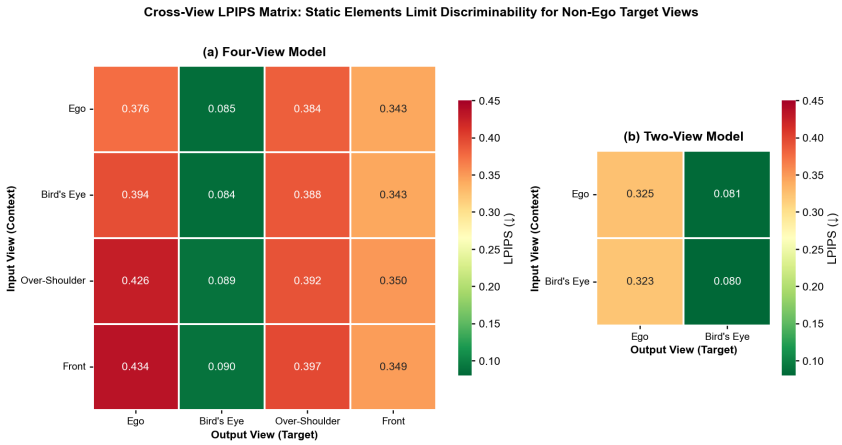


Figure 8: **Cross-view LPIPS matrix.** The bird's-eye column shows uniformly low LPIPS regardless of input view, as the static background map dominates the score rather than meaningful agent localization. Similarly, over-the-shoulder and front columns contain static elements (the agent's body), reducing metric sensitivity. We therefore reserve perceptual metrics for ego-targeted predictions (both same-view and cross-view), where static elements do not dominate. BEV prediction quality is instead quantified using arrow position accuracy, trajectory consistency over time, and bidirectional spawn tests (map→ego). Metrics are computed for a 4-second prediction target, averaged over 1,257 test samples.

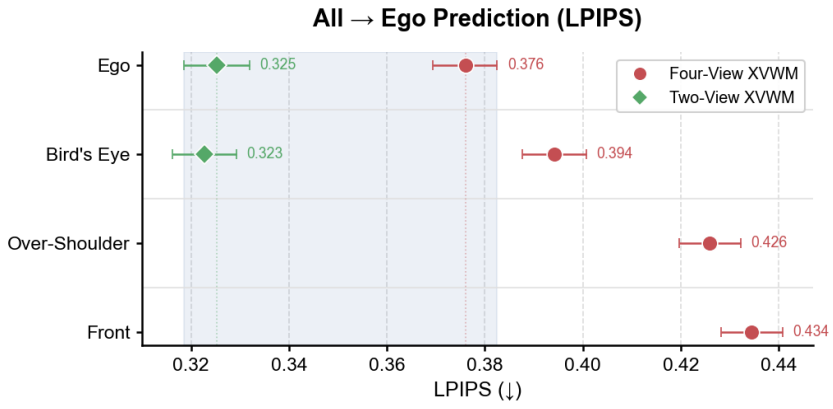


Figure 9: **Cross-view prediction quality (all→ego), LPIPS.** Same analysis as Figure 7 using LPIPS instead of DreamSim. The same trend holds: the Two-View model outperforms the Four-View model, with OS and Front as the primary underperformers.

C SPAWN ANYWHERE

Perhaps the most striking evidence that cross-view training yields spatial understanding lies not in localization (ego→BEV), but in its inverse. Given any marker position on the map and any action, the model generates how the 3D scene will appear from that vantage point at the next instant: a rich visual prediction from a ∼17px marker. Even a slight change in marker orientation completely rotates the field of view in the egocentric frame, forcing the model to attend closely to both the marker's position and heading. The model must also predict an *action-conditioned future*, precisely calibrating the slow marker dynamics observed in the BEV context to the much faster visual changes in egocentric. This is only possible if the model has internalized the spatial structure of the environment. Fig. 10 demonstrates this capability.

BEV→ego is a considerably harder task than ego→BEV. In the latter, the model could in principle predict a static map while ignoring the marker altogether, though as we have shown, it does not. In the BEV→ego direction, no such shortcut exists: the vast difference in information density between input and output, combined with the fact that small marker movements translate to dramatic viewpoint shifts, forces the model to attend to the marker and build correspondingly rich spatial representations. The tiny marker acts as an information bottleneck, and the asymmetry between sparse input and detailed output can only be bridged by genuine 3D understanding.

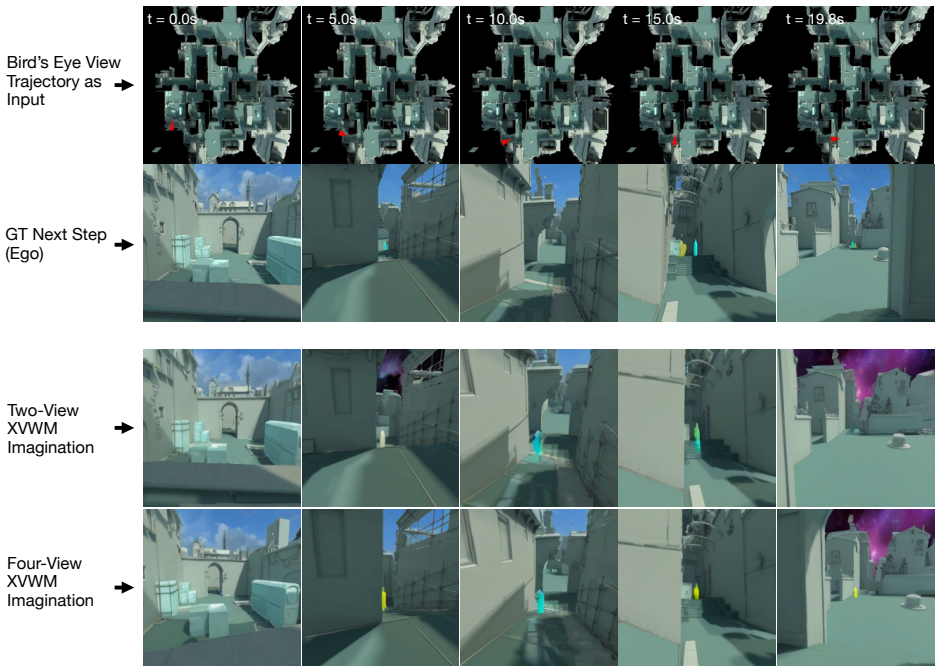


Figure 10: **Spawn anywhere in a known environment.** A dramatic consequence of the emergent bi-directionality from cross-view training is the ability to spawn at arbitrary locations. Here, we take a Bird's-eye view trajectory as input (top row) and predict the action-conditioned next step from an egocentric viewpoint. Both models imagine detailed egocentric views along the trajectory, decoding all the necessary information from the location and orientation of the tiny ∼17px marker alone. This suggests strong emergent understanding of the 3D environment. Notably, the sky varies stochastically when input context is BEV, as expected: the training environment included multiple sky layouts, and BEV provides no information about which sky should appear. This suggests the model has learned spatial structure rather than memorizing input-output pairs. At each instant, the model predicts the next frame, that is 0.2s into the future.

D EXTENDED RELATED WORK

This appendix gives the detailed comparison with the closest multi-view world models summarised in Sec. 2.

Closely related is Multi-View Masked World Models (MV-MWM) (Seo et al., 2023), which trains a multi-view masked autoencoder to reconstruct randomly masked viewpoints, then learns a world model on frozen representations from the encoder. The key distinction is architectural: MV-MWM uses cross-view signals to train the encoder via reconstruction at the current timestep, while the dynamics model operates on frozen features without any cross-view objective. In XVWM, cross-view prediction trains the dynamics model directly: the model must predict the future state in a different viewpoint than the input. This forces the transition function itself—not just the encoder—to learn view-invariant representations of scene structure and dynamics. ReViWo (Pang et al., 2025) explicitly disentangles view-invariant and view-dependent representations via separate encoders and a contrastive loss that forces identical representations across viewpoints. This bottlenecks the view-invariant representation to information present in the least informative view: a bird's-eye marker and a rich egocentric scene share only position and orientation, so only these can be encoded view-invariantly. The world model then operates purely in this abstract space, evolving position and orientation rather than visual state, and reconstruction of the environment relies on a feedforward decoder conditioned on the view-dependent representation. XVWM takes a different approach: we use a standard pretrained encoder and impose no factorization via contrastive learning. A latent diffusion model jointly evolves state and transforms viewpoint, trained end-to-end on a dynamical cross-view objective that uses temporal context and predicts into the future. This allows the model to retain asymmetric information content across views, benefit from iterative refinement during generation, and produce imagination streams without requiring a reference image from the target viewpoint.

In robotic manipulation, Ctrl-World (Guo et al., 2025) takes all camera streams as input and generates all streams jointly, enforcing multi-view consistency through simultaneous prediction. In contrast, XVWM operates from a *single* input view and generates any desired output view. This cross-view transformation task is strictly harder: the model cannot rely on correlations between input and output views, since they depict the scene from entirely different perspectives. To solve it, the model must internalize the 3D structure of the environment and re-render from the target viewpoint. This makes cross-view prediction a form of geometric regularization, forcing view-invariant representations in a way that joint multi-view prediction does not.