Cross-View World Models

Rishabh Sharma* · *Gijs Hogervorst · Wayne E. Mackey · David J. Heeger · Stefano Martiniani**

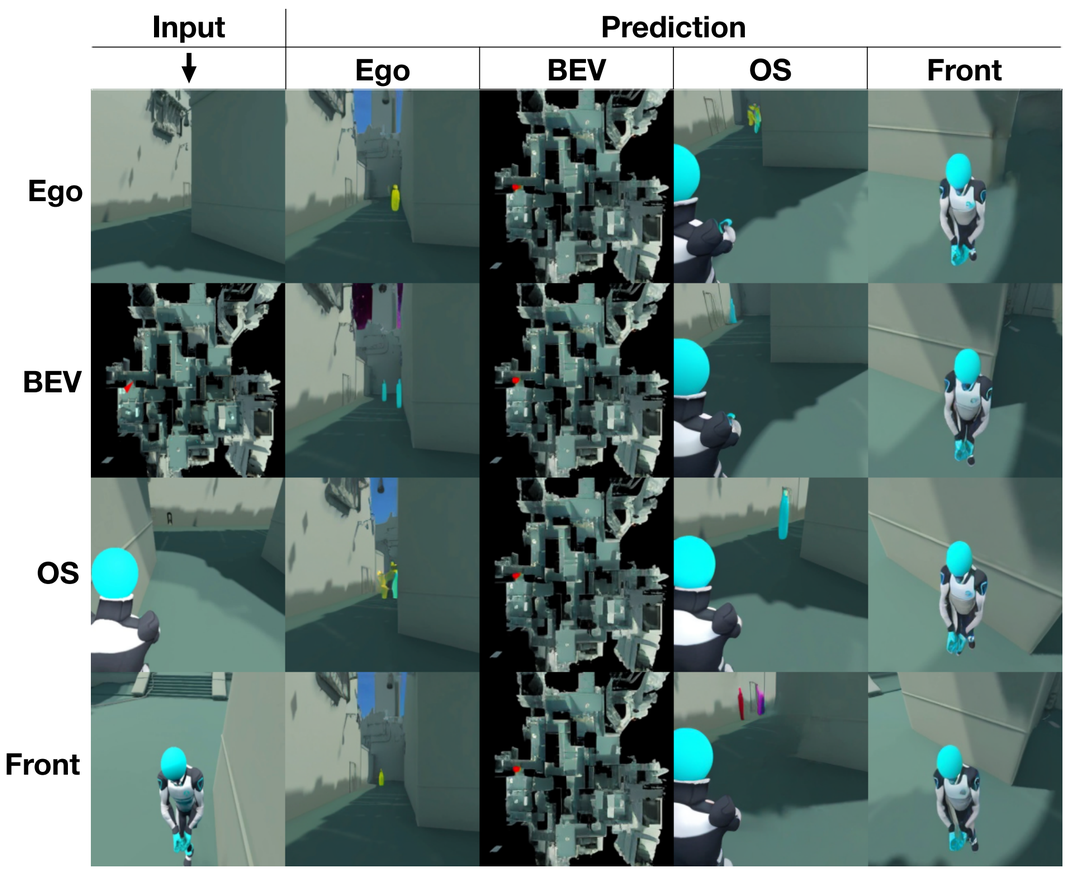
ICLR
International Conference On Learning Representations

NYU · SIMONS FOUNDATION · STATESPACE

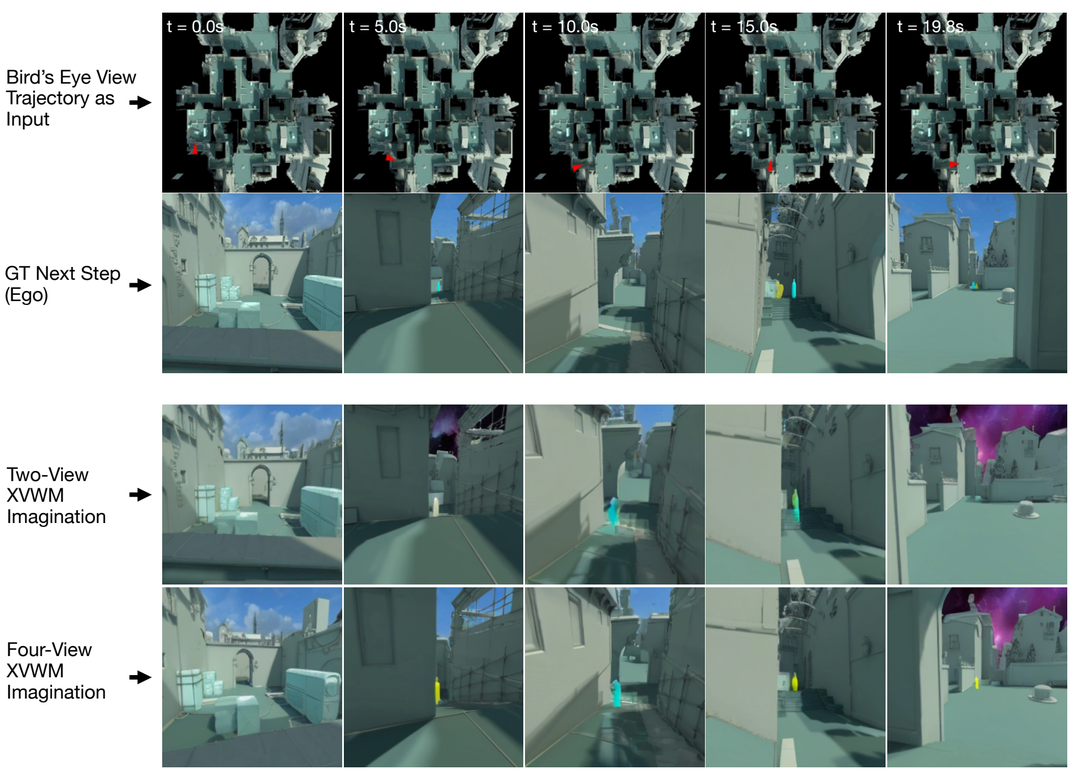
Paper · Connect

Motivation: Choosing the right frame of reference

- **Not every problem is easier in egocentric view.** Navigation favors a bird's-eye map; manipulation favors egocentric; perspective-taking favors another agent's view.
- **Most World models today imagine from only one viewpoint** — usually egocentric — regardless of the task.
- **Cross-View World Models (XVWM)** predict future frames from any viewpoint given context from one:
  - **Context (view_in) + action (Δx, Δy, Δψ) → future frame (view_out)**
  - **Trained with 50% cross-view sampling (view_in ≠ view_out)**
  - **Build on CDiT architecture from Navigation World Models (Bar et al., 2025)**
- **Data.** Synchronized multi-view gameplay recordings (AimLabs) with precise action telemetry — ego / BEV / over-the-shoulder / front-facing.
- **Geometric regularization.** Input and output may share no visual overlap, so the model must encode view-invariant 3D structure — not just match pixels.
- **Result** - Parallel imagination streams across viewpoints + a view-invariant spatial code with functional similarities to mammalian cognitive maps — an "internal GPS" that localizes and orients from visual cues alone.



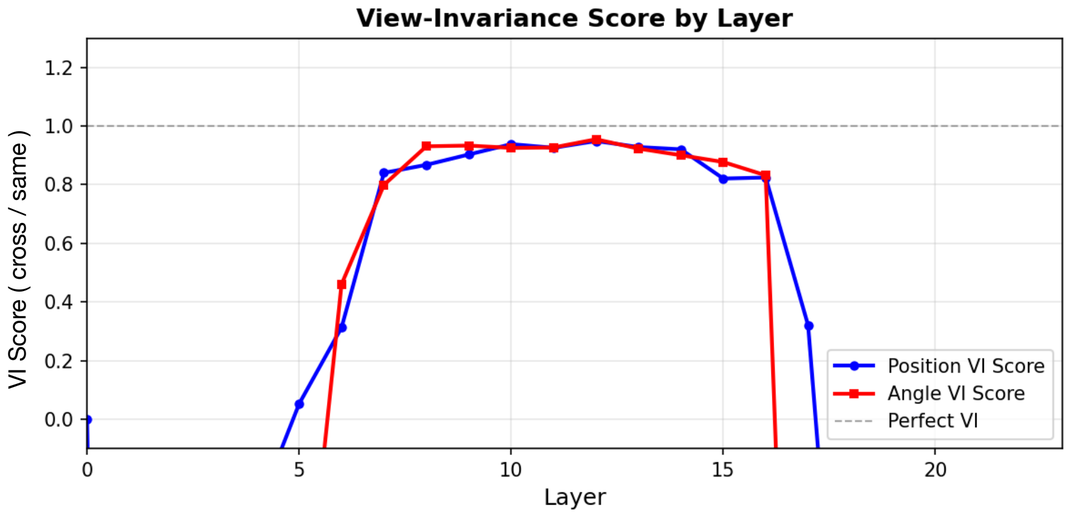
Spawn anywhere: place a marker on the map → full scene



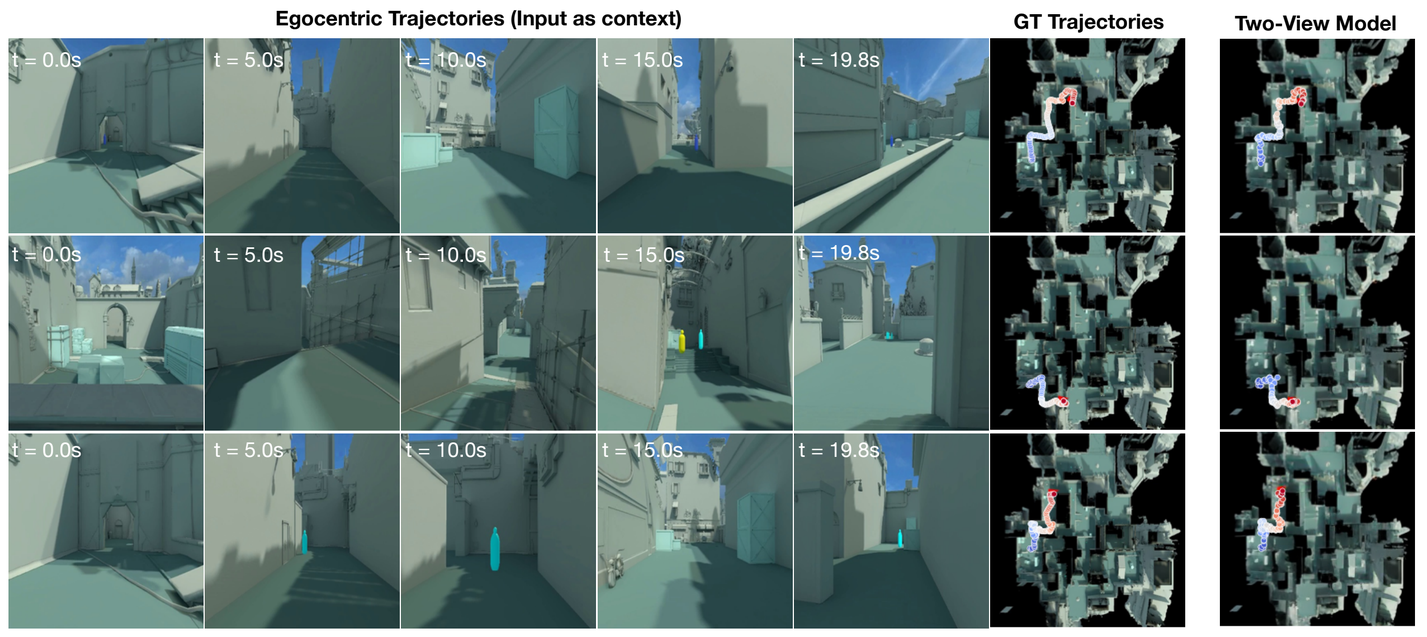
A 17 px red marker on a 224 × 224 BEV map (top) → XVWM ego rendering at t = 0, 5, 10, 15, 20 s (bottom). Action-conditioned predictions, 0.2 s into the future at each step. BEV and ego share no pixel-level overlap and sit at different spatio-temporal scales.

The model cannot match pixels; it must decode position + heading from <0.6% of input pixels, and then render the scene.

A view-invariant spatial code emerges in the middle layers

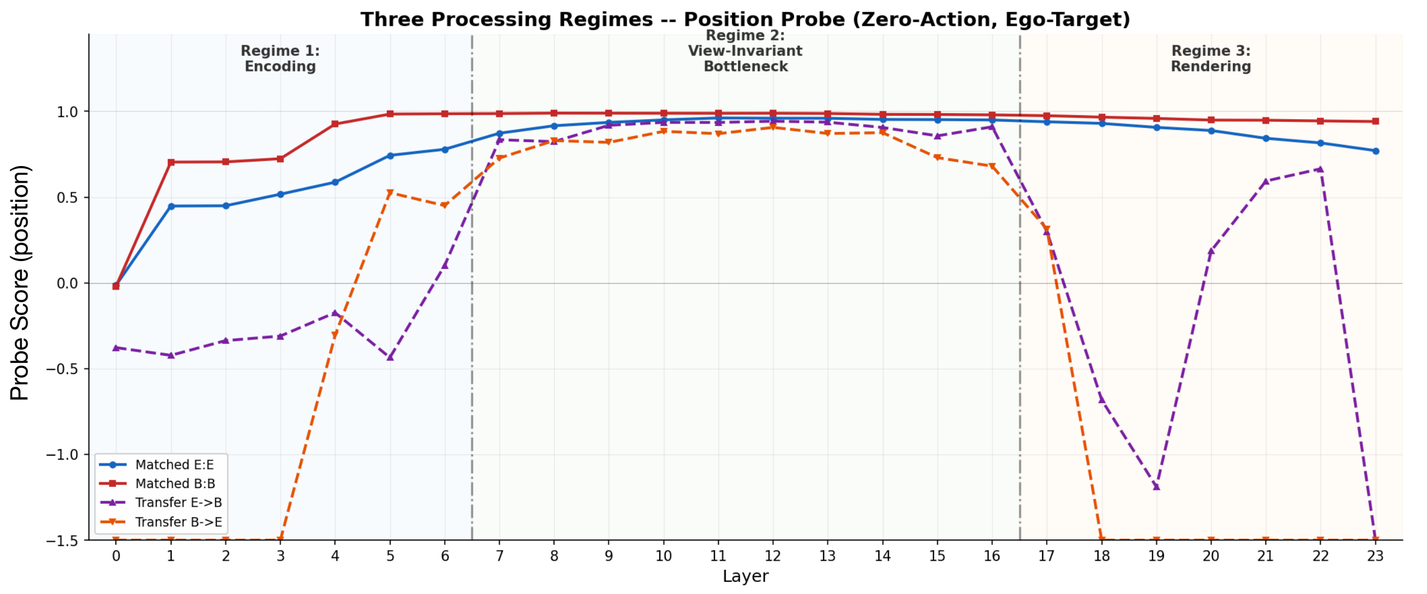


XVWM's emergent internal GPS



Three 20-second egocentric trajectories (filmstrip) and the corresponding BEV paths imagined by XVWM (Two-View model). Blue → red denotes time. Imagined paths track ground truth — the model localizes itself and preserves spatial ordering.

Three processing regimes: encoding, view-invariant bottleneck, rendering



Ridge probe for position, per layer. Solid: matched view (E→E, B→B). Dashed: cross-view transfer (E→B, B→E) — probe trained on one view, tested on the other.

Three regimes emerge:

- **Encode (L0–6)** — features are view-specific, transfer score goes negative.
- **View-invariant bottleneck (L7–16)** — transfer matches the same-view ceiling, i.e. a probe trained on one view reads out the other without retraining.
- **Render (L17–23)** — features re-specialize for the target view and transfer collapses again.

Angular velocity integration after the view-invariant code has formed

Heading is stored as a circular code — a ring in the hidden state. Injected Δψ (angular velocity) rigidly rotates it (gain ≈ 1.0), i.e. the model performs angular velocity integration — but only in the middle layers (L6–16), after the unified cross-view code has formed.

Scan the QRs for the full layer-by-layer animations!

Matched · Cross-view