SPHEREJEPA: LEARNING SPHERICAL VIDEO REPRESENTATIONS FROM PERSPECTIVE VIDEO VIA 360° SELF-SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce SphereJEPA, a joint-embedding predictive architecture that learns to predict features for every direction around the camera from ordinary perspective video. Its context and target come from different projections of the same scene. The context encoder reads the perspective video, an EMA target encoder reads the panorama, and both write onto tokens indexed by direction on a gravity-aligned sphere. Unlike a standard JEPA, whose predictor is discarded after pre-training, SphereJEPA keeps its predictor, because its output is the representation of what lies out of view. Training crops are rendered from the panorama, so the data needs no posed or calibrated capture, and the model never receives camera parameters. Inference is a single forward pass through a 113M-parameter encoder and predictor. The predicted features identify unseen scenes and their orientation, and agree with what the camera later observes. With frozen features and an analytical readout, SphereJEPA roughly halves the TAPVid360 mean angular error of every method not trained on the benchmark, without using point labels.

1 INTRODUCTION

Humans maintain a stable, world-centred understanding of our surroundings. We continue to represent objects that have left our field of view and even infer scene elements that have not been observed using contextual cues. This is what enables us to reach for an object behind us or to sit down on a chair without looking at it. A video representation intended for navigation or tracking must similarly preserve more than the contents of the current frame. When the camera turns or translates, an object moves to another image location and may leave the field of view entirely, although it remains relevant to the task. Perspective-video models normally index features by image position. They can learn motion statistics, but they provide no explicit feature position for a direction outside the image and no stable directional index through a camera rotation.

One approach to tackle this problem is to generate the missing pixels, i.e. to outpaint complete panoramic 360° video from perspective input. Such an explicit representation requires a single solution to be drawn from the space of possible completions. This has recently been successfully modelled using conditional 360° video diffusion (Luo et al., 2025; Wu et al., 2026). However, explicit pixel prediction is expensive, does not capture uncertainty and allocates capacity to appearance details that many downstream tasks do not need. For example, for many tasks it is enough to know where the ground plane is rather than its precise texture.

Joint-embedding predictive architectures (JEPAs) (LeCun, 2022; Bardes et al., 2024) instead predict target *features* in representation space, sidestepping the need for explicit prediction such as is used in masked autoencoder training. This permits the target to retain scene content useful to downstream models without requiring an exact prediction of stochastic texture. We apply that idea to the complete directional field around a camera, asking whether a latent prediction objective can also change the *spatial support* of a perspective-video representation from the observed image plane to all directions on a sphere.. Whereas JEPA uses random masking of feature tokens, we propose structured masking that encourages learning panoramic scene features from whatever portion of the world was observed by a perspective camera moving through space. 360° video offers a direct training signal, because each perspective view is a crop of a panorama that records the rest of the scene at the same instant.

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.

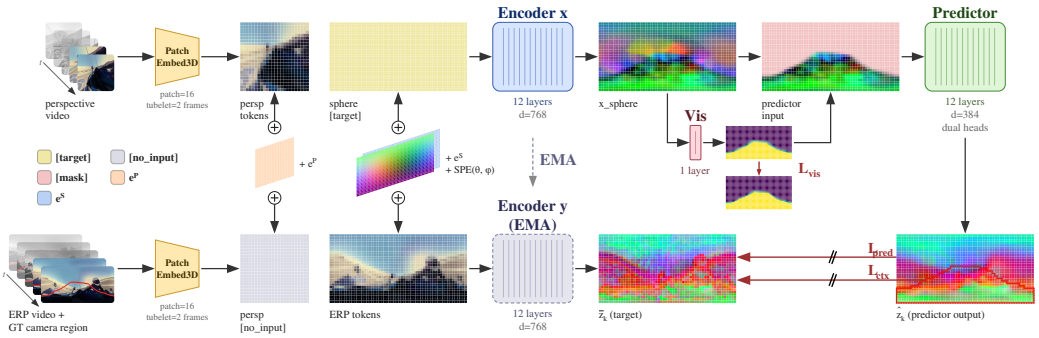

Figure 1: **SphereJEPA Architecture.** The context encoder maps perspective patches onto sphere query tokens, while the EMA target encoder maps the full ERP onto the same directional grid. The visibility head identifies observed directions, and the predictor completes the spherical feature field. Modality embeddings distinguish the two token types.

Large-scale 360° video datasets have recently become available (Wallingford et al., 2024) and remain underused for representation learning. Each equirectangular projection (ERP) frame observes every direction from one optical centre, providing a complete spherical observation from which we can render partial perspective views.

We introduce SphereJEPA (Figure 1), which receives only perspective RGB video at inference and produces a gravity-aligned spherical feature field. During training, the perspective input is rendered from a 360° video, while an exponential-moving-average (EMA) encoder of the complete ERP video supplies a feature target at every direction. Sphere queries and perspective patches interact inside a shared transformer. The model must therefore infer where observed content belongs on the sphere without receiving camera parameters, then predict features at the unobserved directions.

We evaluate three consequences of this construction. First, predicted visibility recovers the rendered in-view region, testing whether perspective content was bound to coherent directions. Second, out-of-view predictions identify both the scene and the correct spherical azimuth after removing branch- and direction-specific feature means. We pair this same-model retrieval diagnostic with an independent predict-then-observe protocol in which a second forward pass directly views directions hidden from the first. Third, frozen spherical features support an analytical TAPVid360 rotation-chain readout without point-label supervision. Our contributions are:

- a perspective-input JEPA whose output features are indexed over a gravity-aligned sphere;
- controlled evidence that its out-of-view features retain scene and directional information on unseen source videos, together with matched correspondence-supervision ablations; and
- a label-free directional tracking readout that uses the spherical index in a downstream task.

2 Related Work

Video Representation Learning and JEPAs. Video self-supervision has used temporal order, playback pace and narration as training signals (Misra et al., 2016; Wang et al., 2020; Miech et al., 2020). Masked autoencoders instead reconstruct removed video tokens (Tong et al., 2022; Feichtenhofer et al., 2022; Wang et al., 2023), while DINOv2 and DINOv3 provide strong dense image features (Oquab et al., 2024; Siméoni et al., 2025). These methods remain indexed by the perspective image grid, with no feature location outside the current field of view. Predictive coding and target-network methods showed that representations could be predicted without reconstructing pixels or using explicit negatives (van den Oord et al., 2018; Grill et al., 2020). JEPA makes this latent prediction the central objective (LeCun, 2022), spanning masked images, motion and video in I-JEPA, MC-JEPA and V-JEPA (Assran et al., 2023; Bardes et al., 2023; 2024); V-JEPA 2 and 2.1 further develop video prediction and dense readout (Assran et al., 2025; Mur-Labadia et al., 2026). SphereJEPA likewise predicts features rather than pixels, but defines its targets over directions on a sphere rather than masked positions in the input grid.

Panoramic and Spherical Representation Learning. Spherical feature extractors address ERP distortion and rotation through projection-aware convolution or equivariant networks (Su & Grauman, 2017; Coors et al., 2018; Cohen et al., 2018; Esteves et al., 2018). Panoramas have also supported gaze prediction, view synthesis and depth estimation (Xu et al., 2018; Attal et al., 2020; Albanis et al., 2021; Zayene et al., 2025). Large 360° video collections now allow a full panorama to supervise what lies outside a rendered perspective crop (Wallingford et al., 2024). Perspective-to-panorama models use this pairing to generate missing RGB video (Luo et al., 2025; Wu et al., 2026). SphereJEPA instead converts the panorama into sphere-indexed feature targets during training and requires only perspective video at inference.

Alignment and Coordinate Systems. Persistent multi-view features require a coordinate system. NeRF uses calibrated rays and scene geometry (Mildenhall et al., 2020), pose relocalisation directly regresses the camera (Kendall et al., 2015), and equivariant networks encode selected transformations (Cohen & Welling, 2016; Weiler & Cesa, 2019). Multi-view learning instead uses paired observations as an alignment signal (Tian et al., 2020). SphereJEPA combines this signal with a spherical index: rendering supplies the training-time correspondence, while inference uses neither camera parameters nor reconstructed 3D geometry.

Point Tracking Under Camera Motion. TAPIR, BootsTAP and CoTracker3 learn image correspondences over time (Doersch et al., 2023; 2024; Karaev et al., 2025); SpatialTracker and TAPIP3D add 3D structure (Xiao et al., 2024; Zhang et al., 2025), while TAPNext treats tracking as prediction (Zholus et al., 2025). TAPVid360 isolates the difficulty relevant here: under a narrow moving camera, a point can leave the image while its direction in the surrounding scene remains defined (Hudson et al., 2025). We use it to test the spherical index, not to train a dedicated tracker.

3 METHOD

SphereJEPA is trained on a paired dataset of perspective and ERP views generated from the same 360° clip i.e. a perspective video $\mathbf{V}^{\mathrm{persp}}$ rendered from an ERP video $\mathbf{V}^{\mathrm{erp}}$. Built on the V-JEPA 2.1 architecture (Mur-Labadia et al., 2026), the network processes a single sequence of perspective and sphere tokens whose content varies by role: image patches, learned query tokens, or a no-input placeholder. During training, the context encoder (*encoder-x*) receives perspective tokens containing the rendered video together with a set of sphere query tokens. Its task is to encode the perspective video, spatially and temporally, *into* the sphere tokens, writing what it sees onto a gravity-aligned directional index *rotation-locked* to the clip's first frame. The target encoder (*encoder-y*), an EMA copy of encoder-x, takes the opposite as input, receiving no-input placeholders at the perspective positions and patches of the full ERP at the sphere positions. It sees the whole sphere and encodes it onto the same sphere tokens, producing the targets. The predictor must then *outpaint* encoder-x's output, computed from the perspective alone, to match encoder-y's, which saw the full ERP. A visibility head estimates which sphere directions the crop actually covered, and controls what the predictor treats as observed. Unlike a standard JEPA, whose predictor is discarded after pre-training, the predictor remains part of SphereJEPA at inference, producing a full sphere of latent features from perspective video alone.

3.1 PERSPECTIVE CONTEXT AND SPHERICAL TARGETS

Training begins with an ERP video $\mathbf{V}^{\mathrm{erp}}$. We sample a camera trajectory and render a perspective video $\mathbf{V}^{\mathrm{persp}}$ by gnomonic projection. We tokenise both ERP and perspective frames into two-frame 16×16 spatiotemporal tubelet patches. We associate with each patch in the ERP video a spherical direction (azimuth θ and elevation ϕ) corresponding to the unit vector pointing through the centre of the patch. The renderer additionally provides the fractional visibility $m_k^{\mathrm{vis,gt}} \in [0, 1]$ of every spherical tubelet k and a soft correspondence between visible spherical and perspective patches. *Fractional visibility* specifies what proportion of each patch in the ERP lies within the field of view of the corresponding perspective frame (since patches on the boundary of the perspective view may be partially observed). *Soft correspondence* stores which spherical patches are overlapped by each perspective patch. Figure S1 shows one training example.

Camera parameters are used to construct training targets but are not provided as model inputs. ERPs are gravity-canonicalised with GeoCalib (Veicht et al., 2024) and rotation-locked to the first frame

following 360Anything (Wu et al., 2026). Our sampled camera trajectories mix simulated motion, DynPose-100K rotations (Rockwell et al., 2025), TAPVid-style motion and static views. We sample field of view, pitch, yaw, and roll, then roll the ERP such that the first crop centre maps to $\theta = 0$.

The training corpus contains 262,769 ten-second clips from 7,792 source videos after filtering the Argus selection of 360-1M (Wallingford et al., 2024; Luo et al., 2025). We render each perspective input as a crop of the highest-resolution decoded ERP video. The crop contains only the sampled field of view, while the paired ERP provides a target for every direction around the camera.

Perspective and sphere tokens use different spatial position encodings. Perspective patch tokens retain V-JEPA 2.1's temporal, row and column rotary position encodings (RoPE). Sphere tokens retain temporal RoPE, while their spatial direction is encoded using spherical positional encoding (SPE), a learnt projection of real spherical harmonics up to degree 16:

$$\mathrm{SPE}(\theta, \phi) = \mathrm{Linear}\left([Y_\ell^m(\theta, \phi)]_{\ell=0}^{16}\right) \in \mathbb{R}^{D_{\mathrm{feat}}}. \tag{1}$$

Learned modality embeddings $\mathbf{e}^{\mathrm{P}}, \mathbf{e}^{\mathrm{S}} \in \mathbb{R}^{D_{\mathrm{feat}}}$ are added to every perspective and sphere token respectively, so the shared transformer can distinguish the two token types. The context encoder jointly processes the perspective patch tokens and one query token for each spherical direction (Figure 2). Joint attention must infer the unknown perspective-to-sphere binding.

The context encoder returns one feature for every token in the joint sequence. Writing it as $f_{\boldsymbol{\Theta}}$, we split its output into perspective features $\{\mathbf{h}_j^{\mathrm{persp}}\}_{j=1}^{N_{\mathrm{persp}}}$ and context sphere features $\{\mathbf{h}_k^{\mathrm{ctx}}\}_{k=1}^{N_{\mathrm{sph}}}$,

$$f_{\boldsymbol{\Theta}}\left(\mathbf{V}^{\mathrm{persp}}, \{\mathbf{q}_k^{\mathrm{sph}}\}\right) = \left(\{\mathbf{h}_j^{\mathrm{persp}}\}_{j=1}^{N_{\mathrm{persp}}}, \{\mathbf{h}_k^{\mathrm{ctx}}\}_{k=1}^{N_{\mathrm{sph}}}\right). \tag{2}$$

The perspective features encode the input patches. The sphere features are the updated query tokens after joint attention with those patches, so $\mathbf{h}_k^{\mathrm{ctx}}$ contains the information that encoder-x assigns to spherical direction k. We concatenate features from four selected encoder depths before passing them to the predictor; a superscript L denotes the final-depth feature. The visibility head and correspondence loss operate on these final-depth sphere features.

The EMA target encoder receives ERP patches at the sphere positions and placeholders in the perspective slots. We discard its outputs at the perspective positions and denote its normalised sphere outputs by $\{\bar{\mathbf{z}}_k^{\mathrm{tgt}}\}_{k=1}^{N_{\mathrm{sph}}}$. These stop-gradient features provide the training target at both visible and out-of-view directions.

3.2 Visibility-Routed Prediction

After encoding, encoder-x has computed context sphere features from the perspective video, while encoder-y has computed target sphere features from the full ERP. Both are indexed by the same directional grid. We now need to identify which sphere directions were observed, use them as context and predict target features across the sphere.

A linear head reads the final-depth sphere feature and predicts whether the corresponding direction was observed,

$$p_k^{\mathrm{vis}} = \mathrm{sigmoid}\left(g_{\mathrm{vis}}\left(\mathbf{h}_k^{\mathrm{ctx},L}\right)\right). \tag{3}$$

The predictor receives encoder context sphere features at directions treated as visible and a learnt mask token elsewhere,

$$\mathbf{h}_k^{\mathrm{route}} = r_k \mathbf{h}_k^{\mathrm{ctx}} + (1 - r_k)\mathbf{e}^{\mathrm{mask}}, \tag{4}$$

where r_k initially uses thresholded rendered visibility. A deterministic schedule changes to detached predicted visibility between steps 5k and 10k; all later training and inference use the prediction. Loss weighting always uses continuous rendered visibility, so routing errors remain part of the inference pipeline and are exposed by every downstream evaluation.

The predictor processes the complete routed sphere sequence. After projecting the sequence to its own width, it adds spherical positional encoding at every sphere position, including those containing the mask token. It also retains V-JEPA 2.1's temporal RoPE within each transformer block. Following the V-JEPA 2.1 predict-all design (Mur-Labadia et al., 2026), its separate output heads return a context estimate $\hat{\mathbf{z}}_k^{\mathrm{ctx}}$ and an out-of-view prediction $\hat{\mathbf{z}}_k^{\mathrm{pred}}$ at every sphere position,

$$P\left(\{\mathbf{h}_k^{\mathrm{route}}\}_{k=1}^{N_{\mathrm{sph}}}\right) = \left(\{\hat{\mathbf{z}}_k^{\mathrm{ctx}}\}_{k=1}^{N_{\mathrm{sph}}}, \{\hat{\mathbf{z}}_k^{\mathrm{pred}}\}_{k=1}^{N_{\mathrm{sph}}}\right). \tag{5}$$

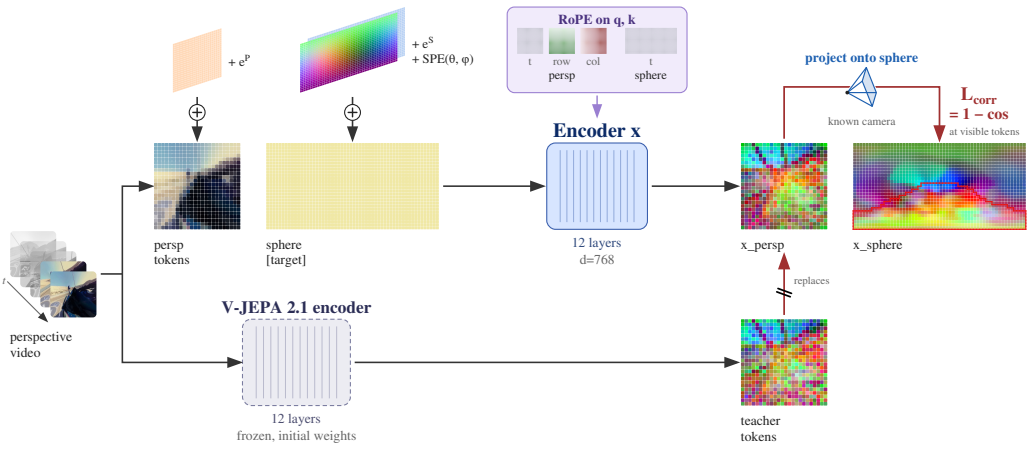

Figure 2: **Position Encoding and Correspondence Supervision.** Perspective tokens use temporal, row and column RoPE; sphere tokens use temporal RoPE and SPE. The camera-derived coverage matrix W maps encoder-x's perspective features to sphere positions for $\mathcal{L}_{\text{corr}}$. In the full teacher mode, frozen V-JEPA features replace these online features.

The context head refines features at observed positions, while the out-of-view head estimates features from the remaining context. We combine their outputs using the same routing value,

$$\hat{\mathbf{z}}_k = r_k \hat{\mathbf{z}}_k^{\text{ctx}} + (1 - r_k)\hat{\mathbf{z}}_k^{\text{pred}}. \tag{6}$$

At inference, $\hat{\mathbf{z}}_k$ forms a full spherical feature field by blending refined context features with out-of-view predictions according to the predicted visibility.

The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{ctx}}\mathcal{L}_{\text{ctx}} + \lambda_{\text{vis}}\mathcal{L}_{\text{vis}} + \lambda_{\text{corr}}\mathcal{L}_{\text{corr}}. \tag{7}$$

Let $w_k = \cos\phi_k$ correct the ERP grid for the solid angle represented by each tubelet, and let $d_1(\mathbf{a}, \mathbf{b})$ denote the mean absolute difference across their feature channels. Prediction and context terms are

$$\mathcal{L}_{\text{pred}} = \frac{\sum_k w_k(1 - m_k^{\text{vis,gt}})d_1(\hat{\mathbf{z}}_k, \bar{\mathbf{z}}_k^{\text{tgt}})}{\sum_k w_k(1 - m_k^{\text{vis,gt}})}, \quad \mathcal{L}_{\text{ctx}} = \frac{\sum_k w_k m_k^{\text{vis,gt}} d_1(\hat{\mathbf{z}}_k^{\text{ctx}}, \bar{\mathbf{z}}_k^{\text{tgt}})}{\sum_k w_k m_k^{\text{vis,gt}}}. \tag{8}$$

Thus, $\mathcal{L}_{\text{pred}}$ encourages the predicted spherical features in regions unobserved by the perspective camera to match the corresponding features extracted from the complete spherical video. $\mathcal{L}_{\text{ctx}}$ applies the same feature matching inside the observed region. As in 360Anything (Wu et al., 2026), the model receives no camera parameters and must learn how the moving perspective view maps onto the spherical grid. $\mathcal{L}_{\text{vis}}$ is binary cross-entropy against the fractional visibility and supervises which directions lie inside the current field of view. Fractional visibility keeps tubelets at a moving crop boundary continuous across their two constituent frames. $\mathcal{L}_{\text{corr}}$ provides direct patch-level correspondence at well-covered visible positions by aligning each sphere feature $\mathbf{h}_k^{\text{ctx},L}$ with a coverage-weighted combination of perspective features that overlap the same sphere position under the training renderer (Figure 2). In its online form, the reference features are encoder-x's perspective-token outputs, so gradients reach both sides of the loss. The full construction instead uses perspective-token features from a frozen copy of the initial V-JEPA 2.1 encoder, providing a fixed teacher target. We argue that this frozen target mainly makes fine-grained correspondence data-efficient, and Section 4.4 tests this with matched controls. We use $\lambda_{\text{ctx}} = 0.5$ and $\lambda_{\text{vis}} = \lambda_{\text{corr}} = 1$.

3.3 Implementation

The context and EMA target encoders use the 12-block, width-768 V-JEPA 2.1 ViT-B/16 backbone; the predictor has 12 blocks and width 384. The model is trained on 32 GH200 GPUs for a 10k-step low-resolution phase with an effective batch size of 512, followed by 20k steps at the final resolution with an effective batch size of 192. Full resolutions, token counts, optimisation and data-mixture details are given in Section S2.

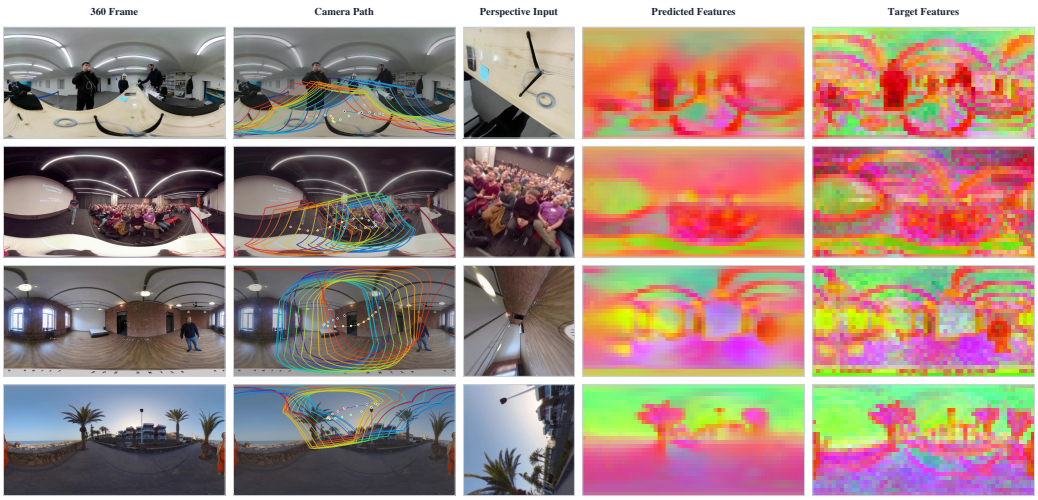

Figure 3: **Spherical Feature Prediction.** Each row shows an ERP frame, the 16-frame camera path, one perspective input, and the predicted and EMA-target sphere features. Features are averaged over eight tubelets and projected through a shared target-fitted PCA basis and colour range.

4 Experiments

We evaluate three capabilities of SphereJEPA: crop-to-sphere binding without camera parameters, scene- and direction-specific prediction beyond the field of view, and directional tracking on TAPVid360 without point-label training. We then ablate correspondence supervision. Crop-to-sphere binding and completion use the same 188 Argus validation clips, whose source videos are absent from the 7,792 training sources. Directional tracking uses the fixed 10,000-clip TAPVid360 test set. We evaluate the fixed final 30k-step checkpoint. To ablate correspondence supervision, we repeat these evaluations on two models trained with the same initialisation, data order and schedule. One replaces its frozen target with online correspondence while the other removes $\mathcal{L}_{\text{corr}}$.

4.1 Crop-to-Sphere Binding

Analytical Visibility Readout. We threshold predicted visibility, take its solid-angle-weighted centroid as viewing direction, and estimate field of view from its angular extent. This analytical readout requires no fitted camera model. Across one fixed trajectory per validation clip, visible-region IoU is 0.636, with 9.57° viewing-direction and 8.91° field-of-view error. The model can therefore correctly place the perspective crop on the sphere from video alone.

Temporal Camera Probe. Predicted visibility cannot recover camera roll, so we fit a lightweight temporal probe to the frozen sphere features and visibility. A pose query reads each tubelet, a temporal transformer combines them, and separate heads estimate field of view and rotation; SphereJEPA remains frozen. Figure 4 compares the two readouts, with architecture and quantitative results in Figure S2 and Table S3.

4.2 Features Beyond the Field of View

We test out-of-view predictions in two ways. Same-model retrieval against full-ERP EMA targets measures scene identity and directional placement, while predict-then-observe compares a prediction with independently encoded features once another trajectory reveals the same direction. Figure 3 shows representative feature fields; predictions retain broad scene structure but are smoother than their targets (Section S4).

For each of 188 clips, predicted out-of-view features rank every full-ERP EMA target by cosine error; the correct scene ranks first in 86.7% of queries, compared with 0.53% chance. To test whether scene information is placed at the correct directions, we subtract per-direction prediction and target means

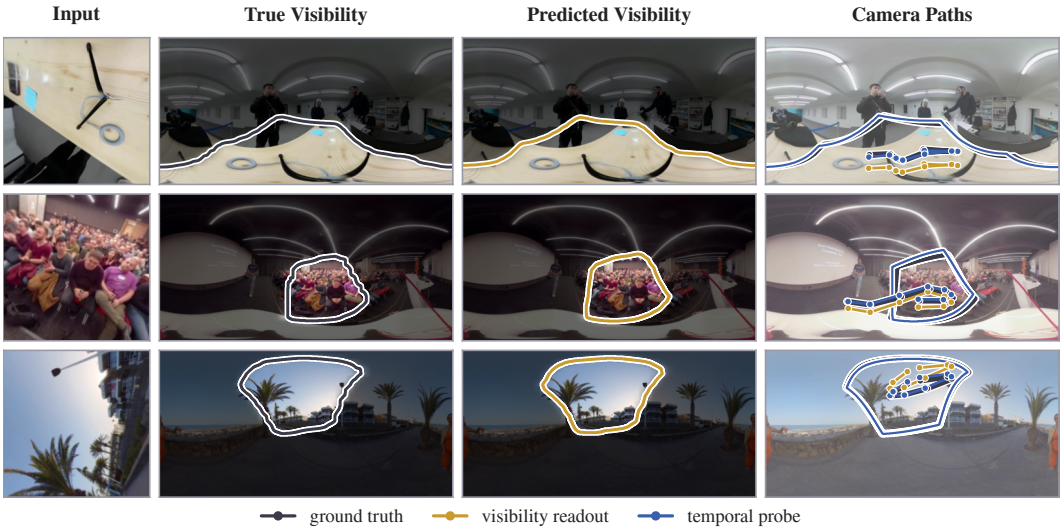

Figure 4: **Visibility and Camera Readout.** The middle columns compare ground-truth and predicted visibility for the central tubelet. The final column overlays ground truth (grey), the analytical visibility readout (gold), and the temporal probe (blue) across all eight tubelets. Central-tubelet frusta are shown for ground truth and the temporal probe. Examples were selected by hand.

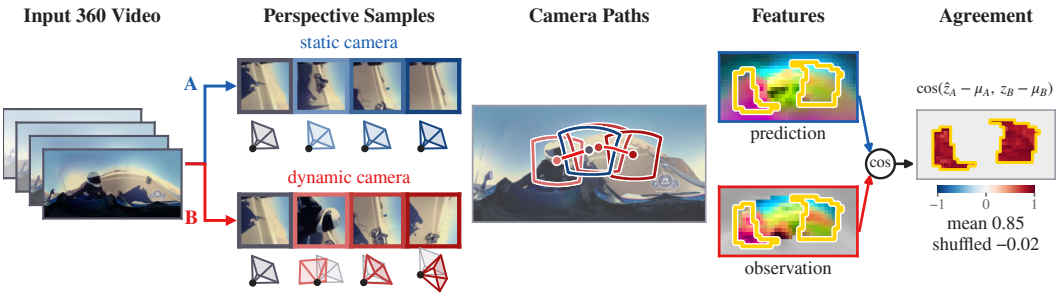

Figure 5: **Predict-Then-Observe Protocol.** Pass A remains at the shared initial pose while independently encoded Pass B follows a trajectory that reveals the evaluated region. Pass A's prediction is compared with Pass B's first observation of each direction. The agreement map shows centred per-direction cosine; its solid-angle-weighted mean is the clip score, alongside the shuffled-scene control.

estimated from 32 calibration clips. On the evaluation set, scene, azimuth and joint top-1 accuracies are 85.9%, 88.5% and 76.9%. Pairing each prediction with a different scene reduces scene and joint accuracy to zero and azimuth accuracy to 14.1%, so positional bias alone cannot explain the result.

Same-model retrieval uses EMA targets, so we also compare predictions with later direct observations. Pass A remains at Pass B's initial pose and predicts unseen directions; independently encoded Pass B later reveals some of them (Figure 5). At first reveal, we compare the Pass A prediction with the corresponding Pass B context feature after branch- and direction-specific centring. The control uses a different scene rendered with the same Pass-A camera geometry.

Across 147 eligible clips, prediction agrees with later observation at 0.792 centred cosine, compared with 0.039 for the control, a paired margin of 0.753. The true prediction wins on every clip. Agreement declines from 0.841 near the visible boundary to 0.643 at 90–180°, but remains above the control in every bin (supplementary Figure S6).

4.3 Analytical Directional Tracking

TAPVid360 tracks query directions through a moving narrow-field-of-view camera, including while they are outside the image (Hudson et al., 2025). We encode the frames, match perspective patches to

Table 1: **Direction-Controlled Retrieval.** Top-1 accuracy (%) on the evaluation set after direction-wise centring; brackets are 95% clip-bootstrap confidence intervals.

Task	Top-1	Shuffled	Chance
Scene at fixed direction	85.9 [80.8, 91.0]	0.0	0.6
Azimuth within scene	88.5 [83.3, 93.0]	14.1	3.6
Scene and azimuth	76.9 [69.9, 83.3]	0.0	0.02

Table 2: **Predict-Then-Observe Evaluation.** Centred cosine agreement on the evaluation set; brackets are 95% clip-bootstrap confidence intervals.

Feature comparison	Agreement ↑	Δ vs. shuffled ↑
SphereJEPA prediction	.792 [.771, .813]	.753 [.710, .795]
Raw sphere context	.485 [.449, .521]	–
Shuffled-scene prediction	.039 [.002, .078]	0
Direct context vs. full-ERP EMA	.681 [.664, .698]	–

Table 3: **TAPVid360 Point Tracking.** Threshold accuracy (↑) averages the benchmark's five angular thresholds; angular distance (AD, ↓) is in degrees. IF and OOF denote in- and out-of-frame points. SphereJEPA and the frozen-feature baselines use no TAPVid labels; Kabsch rows use the same analytical readout.

	Threshold Accuracy (↑)			AD (↓)		
Method	**All**	**IF**	**OOF**	**All**	**IF**	**OOF**
Supervised baselines reported by TAPVid360 (Hudson et al., 2025)						
TAPNext (Zholus et al., 2025)	.0082	.0191	.0004	51.98°	36.59°	62.46°
TAPIR (Doersch et al., 2023)	.0106	.0251	.0003	49.81°	33.88°	60.52°
BootsTAPIR (Doersch et al., 2024)	.0126	.0293	.0005	48.36°	33.32°	58.38°
TAPIP3D (Zhang et al., 2025)	**.2476**	.4698	.0850	36.44°	23.32°	45.80°
SpatialTracker (Xiao et al., 2024)	.2239	.4893	.0303	38.88°	22.16°	50.44°
CoTracker3 (Karaev et al., 2025)	.2435	**.5588**	.0158	37.43°	17.64°	50.98°
CoTracker360 (Hudson et al., 2025)	.2386	.4060	**.1160**	**8.27°**	**3.95°**	**10.98°**
Frozen perspective-feature baselines						
DINOv3 (Siméoni et al., 2025), argmax	.0473	.1108	.0015	43.54°	21.86°	58.41°
DINOv3 (Siméoni et al., 2025), Kabsch readout	**.0632**	**.1395**	.0021	44.61°	25.15°	58.01°
V-JEPA 2.1 (Mur-Labadia et al., 2026), argmax	.0401	.0910	.0010	46.36°	25.31°	61.00°
V-JEPA 2.1 (Mur-Labadia et al., 2026), Kabsch readout	.0455	.0998	.0010	46.68°	27.52°	60.01°
SphereJEPA	.0442	.0805	**.0146**	**19.29°**	**12.32°**	**23.64°**

sphere features, and use Kabsch alignment to estimate camera rotation over time. Unlike all methods reported in TAPVid360, SphereJEPA requires no point-tracking training: the model remains frozen, the readout has no learned parameters, and point labels are used only for evaluation.

SphereJEPA achieves $19.29°$ mean angular distance, compared with $37.43°$ for CoTracker3 and $43.54°$ for the strongest frozen-feature baseline (Table 3). CoTracker360, trained for TAPVid360, reaches $8.27°$, while SphereJEPA's 0.0442 threshold accuracy remains below specialised trackers. The result therefore demonstrates a consistent broad direction through out-of-view intervals, not precise point localisation (Figure 6). We build DINOv3 and V-JEPA 2.1 baselines from released frozen features using direct matching or our Kabsch readout, without TAPVid360 adaptation.

4.4 Correspondence-Loss Ablations

To isolate correspondence supervision, we compare the full frozen-teacher model with matched controls trained without $\mathcal{L}_{\text{corr}}$ and with online encoder-x perspective features. All other training and evaluation settings are fixed. Table 4 summarises their binding, completion and tracking results.

All three constructions retain similar visibility IoU and predict-then-observe agreement, showing that coarse binding and scene-conditioned hidden features arise without the frozen teacher. Their clearest difference is transfer: removing correspondence raises TAPVid error from $19.29°$ to $49.43°$, while online correspondence reaches $32.87°$ and reduces scene and joint retrieval to 47.4% and 41.0%.

The full model beats the no-correspondence and online controls on 97.6% and 91.7% of TAPVid360 clips, respectively, with similar results out of frame (supplementary Figure S8). Online correspondence nevertheless preserves predict-then-observe agreement ($.796$ versus $.792$), suggesting that it reorganises the representation rather than erasing its hidden-scene signal. The frozen target therefore primarily improves fine-grained, transferable directional correspondence.

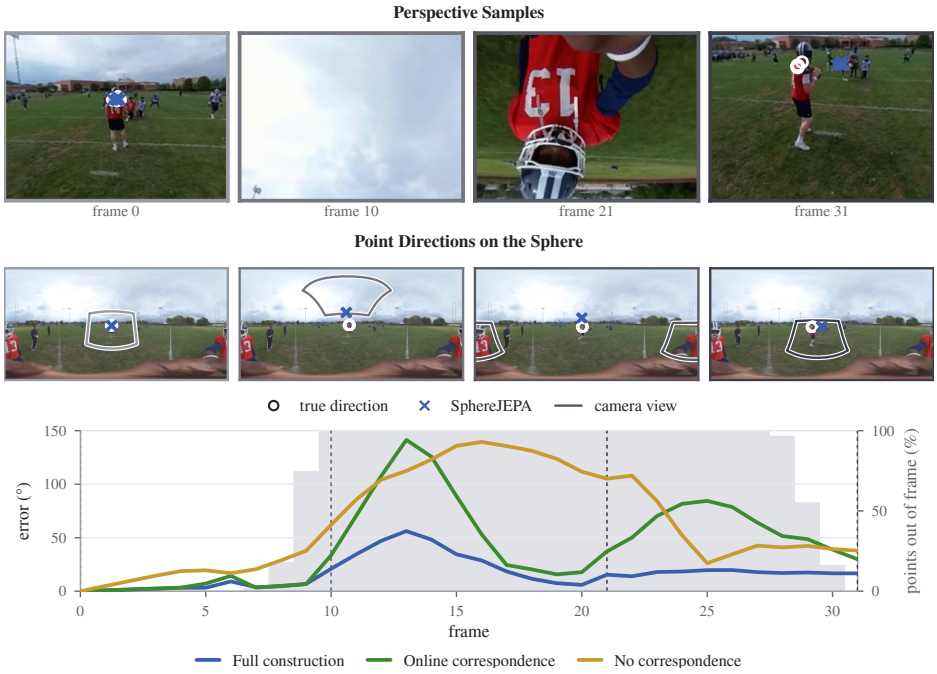

Figure 6: **TAPVid360 Evaluation Example.** Circles mark ground-truth directions and blue crosses SphereJEPA estimates. ERPs below show all points and the current view. The lower panel compares the three matched constructions; shading gives the out-of-frame fraction.

Table 4: **Matched Correspondence-Loss Ablations.** The full model and matched controls across the preceding evaluations. Vis. IoU measures predicted-view overlap; Scene, Azimuth and Joint are retrieval accuracies (%); Observe is centred predict-then-observe agreement; TAP AD is mean angular error in degrees.

	Vis. IoU ↑	Scene ↑	Azimuth ↑	Joint ↑	Observe ↑	TAP AD ↓
Full construction	**.636**	**85.9**	88.5	**76.9**	.792	**19.29**
No correspondence	.633	78.2	**89.1**	69.9	.773	49.43
Online correspondence	.620	47.4	85.3	41.0	**.796**	32.87

5 Discussion and Conclusion

We have introduced SphereJEPA, a joint-embedding predictive architecture that produces spherical feature fields from perspective video. Our evaluations show that the model has learnt a useful directional latent space: it places observed content at coherent sphere directions, retains scene- and azimuth-specific information outside the field of view, and supports downstream directional tracking without task-specific training. Together, these results show that partial observations, predictions and downstream readouts can share one directional coordinate system.

Producing a 360° latent representation from ordinary perspective video could help embodied agents reason beyond their current view, decide where to look next and maintain tracks during navigation without generating missing pixels or reconstructing a complete 3D scene. SphereJEPA is an initial step in this direction: it predicts only one latent completion, uses a dense spherical token grid that limits resolution, and has so far been evaluated through analytical readouts rather than closed-loop control. Future work could model multiple plausible surroundings, incorporate actions and egomotion, develop more efficient spherical representations, and test their value for robotic planning and navigation.

AI USE STATEMENT

Generative AI tools were used as coding and editorial assistants to design evaluation protocols, audit implementation–manuscript consistency, revise prose, and generate or edit code. The authors reviewed all AI-assisted work, including experimental designs, code changes, citations, outputs, interpretations, and scientific claims, and take responsibility for the final content. AI systems did not generate experimental data.

REPRODUCIBILITY STATEMENT

The appendix records the dataset construction, model architecture, complete training schedule, and evaluation protocols. The supplementary submission will include anonymous training, evaluation, and dataset-generation code, immutable split manifests, and configuration files. We will publicly release the complete codebase after the review period.

REFERENCES

Georgios Albanis, Nikolaos Zioulis, Petros Drakoulis, Vasileios Gkitsas, Vladimiros Sterzentsenko, Federico Alvarez, Dimitrios Zarpalas, and Petros Daras. Pano3D: A holistic benchmark and a solid baseline for 360 degrees depth estimation. *CoRR*, abs/2109.02749, 2021.

M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Komeili, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, S. Arnaud, A. Gejji, A. Martin, F. R. Hogan, D. Dugas, P. Bojanowski, V. Khalidov, P. Labatut, F. Massa, M. Szafraniec, K. Krishnakumar, Y. Li, X. Ma, S. Chandar, F. Meier, Y. LeCun, M. Rabbat, and N. Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning, 2025.

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.

Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. MatryODShka: Real-time 6dof video view synthesis using multi-sphere images. In *European Conference on Computer Vision*, 2020.

A. Bardes, J. Ponce, and Y. LeCun. MC-JEPA: A joint-embedding predictive architecture for self-supervised learning of motion and content features, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael G. Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

T. Cohen and M. Welling. Group equivariant convolutional networks, 2016.

Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

B. Coors, A. P. Condurache, and A. Geiger. SphereNet: Learning spherical representations for detection and classification in omnidirectional images, 2018.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, Joao Carreira, and Andrew Zisserman. BootsTAP: Bootstrapped training for tracking-any-point. In *Asian Conference on Computer Vision*, 2024.

C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis. Learning SO(3) equivariant representations with spherical CNNs, 2018.

Christoph Feichtenhofer, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *Advances in Neural Information Processing Systems*, 2022.

Q. Garrido, N. Ballas, M. Assran, A. Bardes, L. Najman, M. Rabbat, E. Dupoux, and Y. LeCun. Intuitive physics understanding emerges from self-supervised pretraining on natural videos, 2025. arXiv:2502.11831.

J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. Gheshlaghi Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2455–2467, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Finlay G. C. Hudson, James A. D. Gardner, and William A. P. Smith. TAPVid-360: Tracking any point in 360 from narrow field of view video. *CoRR*, abs/2511.21946, 2025.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labeling real videos. In *IEEE/CVF International Conference on Computer Vision*, 2025.

A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization, 2015.

Tobias Koch, Lukas Liebel, Marco Körner, and Friedrich Fraundorfer. Comparison of monocular depth estimation methods using geometrically relevant metrics on the iBims-1 dataset. *Computer Vision and Image Understanding*, 191:102877, 2020. doi: 10.1016/j.cviu.2019.102877.

Yann LeCun. A path towards autonomous machine intelligence, 2022. OpenReview position paper, version 0.9.2.

Zhengqi Li and Noah Snavely. MegaDepth: Learning Single-View Depth Prediction From Internet Photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2041–2050, 2018.

Rundong Luo, Matthew Wallingford, Ali Farhadi, Noah Snavely, and Wei-Chiu Ma. Beyond the frame: Generating 360 degrees panoramic videos from perspective videos. In *IEEE/CVF International Conference on Computer Vision*, 2025.

A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman. End-to-end learning of visual representations from uncurated instructional videos, 2020.

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. In *European Conference on Computer Vision*, 2020.

I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: Unsupervised learning using temporal order verification, 2016.

Lorenzo Mur-Labadia, Matthew J. Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *CoRR*, abs/2603.14482, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Chris Rockwell, Joseph Tung, Tsung-Yi Lin, Ming-Yu Liu, David F. Fouhey, and Chen-Hsuan Lin. Dynamic camera poses and where to find them. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking localization and mapping for augmented reality. In *Computer Vision – ECCV 2022*, pp. 686–704, 2022.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A Multi-view Stereo Benchmark with High-Resolution Images and Multi-camera Videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2538–2547, 2017. doi: 10.1109/CVPR.2017.272.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Computer Vision – ECCV 2012*, pp. 746–760, 2012. doi: 10.1007/978-3-642-33715-4_54.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *CoRR*, abs/2508.10104, 2025.

Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360 degrees imagery. In *Advances in Neural Information Processing Systems*, 2017.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, 2020.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022.

A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding, 2018.

Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Learning single-image calibration with geometric optimization. In *European Conference on Computer Vision*, 2024.

Matthew Wallingford, Anand Bhattad, Aditya Kusupati, Vivek Ramanujan, Matt Deitke, Aniruddha Kembhavi, Sham Kakade, Roozbeh Mottaghi, Wei-Chiu Ma, and Ali Farhadi. From an image to a scene: Learning to imagine the world from a million 360 degrees videos. In *Advances in Neural Information Processing Systems*, 2024.

J. Wang, J. Jiao, and Y.-H. Liu. Self-supervised video representation learning by pace prediction, 2020.

L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking, 2023.

M. Weiler and G. Cesa. General E(2)-equivariant steerable CNNs, 2019.

Ziyi Wu, Daniel Watson, Andrea Tagliasacchi, David J. Fleet, Marcus A. Brubaker, and Saurabh Saxena. 360Anything: Geometry-free lifting of images and videos to 360 degrees. In *European Conference on Computer Vision*, 2026.

Y. Xiao, Q. Wang, S. Zhang, N. Xue, S. Peng, Y. Shen, and X. Zhou. SpatialTracker: Tracking any 2D pixels in 3D space, 2024.

Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao. Gaze prediction in dynamic 360 degrees immersive videos, 2018.

M. Zayene, J. Endres, A. Havolli, C. Corbière, S. Cherkaoui, A. Kontouli, and A. Alahi. HELVIPAD: A real-world dataset for omnidirectional stereo depth estimation, 2025.

B. Zhang, L. Ke, A. W. Harley, and K. Fragkiadaki. TAPIP3D: Tracking any point in persistent 3D geometry, 2025.

A. Zholus, C. Doersch, Y. Yang, S. Koppula, V. Patraucean, X. O. He, I. Rocco, M. S. M. Sajjadi, S. Chandar, and R. Goroshin. TAPNext: Tracking any point (TAP) as next token prediction, 2025.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the Continuity of Rotation Representations in Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5745–5754, 2019.

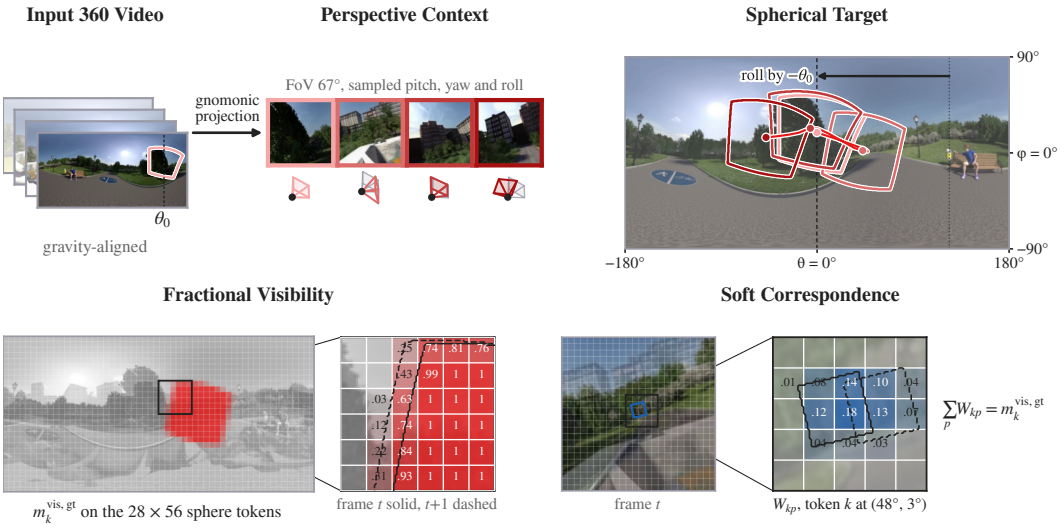

Figure S1: **Training-Example Construction.** A sampled camera path renders perspective context from a gravity-aligned ERP. Rolling the ERP by $-\theta_0$ places the first view at $\theta = 0$, defining target azimuth relative to the initial observation. The lower row shows tubelet visibility and the soft correspondence W_{kp} between sphere token k and perspective patches p.

S1 Dataset Construction

The training corpus starts from the Argus quality-filtered selection (Luo et al., 2025) of 360-1M (Wallingford et al., 2024). We remove missing source files and clips that fail decoding, leaving 262,769 ten-second clips from 7,792 source videos. All reported validation examples and retrieval tests use source videos absent from this training set.

The pre-processing pipeline gravity-canonicalises each ERP with GeoCalib (Veicht et al., 2024) and rotation-locks the video to its first frame following 360Anything (Wu et al., 2026). Perspective crops are rendered from the highest available decoded ERP resolution. The ERP target is resized separately, so changing target resolution does not change the rendered perspective input beyond its final resize. Figure S1 traces one training example through this pipeline.

Spherical Coordinates. ERP pixel coordinates map to spherical directions. For normalised coordinates $(u, v) \in [0, 1]^2$, we use azimuth $\theta = (u - 0.5)2\pi$ and elevation $\phi = (0.5 - v)\pi$. The corresponding unit direction in a right-up-forward frame is

$$\mathbf{d}(\theta, \phi) = (\cos\phi \sin\theta,\ \sin\phi,\ \cos\phi\cos\theta)^\top. \tag{S1}$$

The horizontal ERP boundary is continuous on the sphere. SphereJEPA represents this continuity through spherical-harmonic direction encodings and global attention over sphere-indexed tokens, rather than through a local wrap-around convolution at the ERP seam.

Tubelet-Consistent Geometry. Each token spans two frames. Visibility coverage and perspective-to-sphere correspondence are computed independently in both constituent frames and then averaged. Continuous coverage in $[0, 1]$ supervises the visibility head and weights the visible and out-of-view feature losses. Thresholding the averaged coverage at 0.5 gives the ground-truth routing mask used early in training. This avoids assigning a moving boundary tubelet from only one of its frames.

S2 Architecture and Optimisation

Backbone and Predictor. The context and EMA target encoders use a ViT-B/16 backbone with 12 blocks, width 768, 12 attention heads and a two-frame tubelet embedding. Features from four evenly spaced encoder depths are layer-normalised, concatenated and passed to the predictor. The

Table S1: **Training Schedule.** Model, EMA and optimiser state continue across the two resolution phases.

Phase	ERP	Perspective	Steps	Effective Batch
Low resolution	256×512	224×224	10k	512
High resolution	448×896	384×384	20k	192

predictor has 12 transformer blocks, width 384 and 12 attention heads. Real spherical harmonics through degree 16 give 289 coefficients, which a learnt linear layer projects to the token width.

V-JEPA 2.1 Compatibility. The inherited backbone and predictor parameters are initialised from the released V-JEPA 2.1 model (Mur-Labadia et al., 2026). Perspective inputs use its ImageNet pixel normalisation, LayerNorm epsilon and three-axis RoPE convention. Temporal RoPE is applied to the perspective and sphere queries and keys in every transferred encoder block. Row and column RoPE is applied only to perspective tokens, with coordinates interpolated to the checkpoint's 16×16 reference grid. Sphere tokens instead receive their spatial coordinates from spherical harmonics. The predictor retains temporal RoPE in every transferred block and likewise replaces planar spatial RoPE with spherical harmonics. The new modality embeddings, sphere queries, visibility head, spherical projection and output projections are randomly initialised.

Optimisation. Training uses AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.04 on non-bias and non-normalisation parameters, bfloat16 mixed precision and fixed EMA momentum 0.99925. V-JEPA-initialised encoder parameters use a $0.25\times$ learning-rate multiplier and inherited predictor parameters use $0.5\times$; new parameters use the full rate. The loss weights are $\lambda_{\text{ctx}} = 0.5$ and $\lambda_{\text{vis}} = \lambda_{\text{corr}} = 1$. The full construction uses the frozen-teacher form of $\mathcal{L}_{\text{corr}}$, implemented by `lambda_vjepa_corr`. The code flag `lambda_corr` controls the online form and is zero in the full construction. There is no camera regression or out-of-view encoder penalty.

Matched Correspondence-Loss Controls. The no-correspondence control disables $\mathcal{L}_{\text{corr}}$ by setting both code weights to zero. The online control uses its online form with weight one in place of the frozen-teacher form. Both retain the same initialisation, data order, camera mixture, optimisation schedule and fixed 30k endpoint of the full construction.

The low-resolution phase uses a 500-step warmup followed by cosine decay from 5.25×10^{-4} to 1×10^{-4}. The high-resolution phase restarts the schedule with a 500-step warmup and cosine decay from 7×10^{-5} to 2×10^{-5}. Both phases update the complete model. The run processes 8.96 million samples on 32 GH200 GPUs and reports the fixed final 30k-step checkpoint for every downstream benchmark.

Token Construction. The context encoder $f_{\boldsymbol{\Theta}}$ and EMA target encoder $\bar{f}_{\bar{\boldsymbol{\Theta}}}$ process a joint sequence of N_{persp} perspective tokens and N_{sph} sphere tokens. Both encoders process tubelets, so a 16-frame clip yields eight temporal token slices. At the final training resolution (Table S1) this gives $N_{\text{sph}} = 8 \times 28 \times 56 = 12{,}544$ sphere tokens for the 448×896 ERP and $N_{\text{persp}} = 8 \times 24 \times 24 = 4{,}608$ perspective tokens for the 384×384 crop, a joint sequence of $17{,}152$ tokens. The joint sequence, rather than an encode-then-project design, matches the problem setting because the camera is unknown at inference. No geometric projection from image positions to sphere directions is available, so attention between the two token sets computes the binding of perspective content to directions. The context input contains real perspective patches and placeholder sphere queries:

$$\mathbf{X}_i^{\text{ctx}} = \begin{cases} \mathbf{x}_i^{\text{persp}} + \mathbf{e}^{\text{P}} & i \le N_{\text{persp}}, \\ \mathbf{e}^{\text{tgt}} + \text{SPE}(\theta_{i-N_{\text{persp}}}, \phi_{i-N_{\text{persp}}}) + \mathbf{e}^{\text{S}} & N_{\text{persp}} < i \le N_{\text{persp}} + N_{\text{sph}}, \end{cases} \tag{S2}$$

where $\mathbf{e}^{\text{P}}$ and $\mathbf{e}^{\text{S}}$ are learned modality embeddings and $\mathbf{e}^{\text{tgt}}$ is a learned placeholder marking sphere positions whose content is to be predicted. Within each encoder block, the perspective-token queries and keys receive V-JEPA 2.1's temporal, row and column RoPE. Sphere-token queries and keys receive the same temporal RoPE, while their spatial position is supplied by $\text{SPE}(\theta, \phi)$. Row and column coordinates are interpolated to the 16×16 reference grid of the initial V-JEPA 2.1 model.

The target input contains placeholder perspective tokens and ERP sphere patches:

$$\mathbf{X}_i^{\mathrm{tgt}} = \begin{cases} \mathbf{e}^{\mathrm{nul}} + \mathbf{e}^{\mathrm{P}} & i \le N_{\mathrm{persp}}, \\ \mathbf{x}^{\mathrm{erp}}_{i-N_{\mathrm{persp}}} + \mathrm{SPE}(\theta_{i-N_{\mathrm{persp}}}, \phi_{i-N_{\mathrm{persp}}}) + \mathbf{e}^{\mathrm{S}} & N_{\mathrm{persp}} < i \le N_{\mathrm{persp}} + N_{\mathrm{sph}}. \end{cases} \tag{S3}$$

Here $\mathbf{e}^{\mathrm{nul}}$ is $\mathbf{e}^{\mathrm{tgt}}$'s target-side counterpart, a fixed placeholder standing in for the absent perspective input. The context encoder output at sphere position k is $\mathbf{h}_k^{\mathrm{ctx}}$. The target feature is the stop-gradient, layer-normalised EMA output $\bar{\mathbf{z}}_k^{\mathrm{tgt}}$ at the same sphere position.

Predicted-Visibility Routing. For steps 0–5k, the predictor input uses the thresholded render-derived visibility. Between steps 5k and 10k, a deterministic batchwise schedule increases the probability of detached predicted-visibility routing linearly from zero to one. Predicted routing is used from step 10k onward and at inference. If r_k denotes the selected routing value, the predictor input is

$$\mathbf{h}_k^{\mathrm{route}} = r_k \mathbf{h}_k^{\mathrm{ctx}} + (1 - r_k)\mathbf{e}^{\mathrm{mask}}. \tag{S4}$$

The predictor processes the full routed sphere sequence, applies separate context and out-of-view output heads, and returns their routed combination. The context loss acts on $\hat{\mathbf{z}}_k^{\mathrm{ctx}}$, whereas the prediction loss acts on the routed output $\hat{\mathbf{z}}_k$ at out-of-view positions. Both remain weighted by continuous render-derived visibility; detaching predicted routing prevents the feature objective from changing the visibility scores through this path:

$$P\left(\{\mathbf{h}_k^{\mathrm{route}}\}_{k=1}^{N_{\mathrm{sph}}}\right) = \left(\{\hat{\mathbf{z}}_k^{\mathrm{ctx}}\}_{k=1}^{N_{\mathrm{sph}}}, \{\hat{\mathbf{z}}_k^{\mathrm{pred}}\}_{k=1}^{N_{\mathrm{sph}}}\right), \tag{S5}$$

$$\hat{\mathbf{z}}_k = r_k \hat{\mathbf{z}}_k^{\mathrm{ctx}} + (1 - r_k)\hat{\mathbf{z}}_k^{\mathrm{pred}}. \tag{S6}$$

Camera Distribution. Both resolution phases use the same trajectory mixture: 20% successfully loaded rotations from DynPose-100K (Rockwell et al., 2025), 25% general TAPVid-style motion, 45% simulated or interest-guided motion and 10% static or slow motion. Horizontal field of view spans 30°–150°. The TAPVid-style component includes spin, accumulated rotation, out-and-back, perspective-sweep, spiral and static subfamilies. TAPVid point tracks and point labels are never used for training or model selection.

S3 Evaluation Protocol Details

Camera Readout. The analytical visibility readout thresholds the predicted tubelet mask, computes its solid-angle-weighted spherical centroid and estimates field of view from its angular extent. The temporal probe (Figure S2) receives the frozen context sphere features and predicted visibility from all eight tubelets. A linear layer projects each sphere feature together with its visibility score, and fixed real spherical harmonics are added to the result as positional encoding. A shared pose query cross-attends to the tokens of each tubelet, and a temporal transformer mixes the resulting pose tokens. The field-of-view head reads the mean pose token and predicts one value per clip. The rotation head predicts a 6D rotation (Zhou et al., 2019) that corrects a zero-roll look-at prior, whose forward axis is the visibility-weighted mean direction of the predicted mask. Probe fitting does not update SphereJEPA. All held-out-video camera metrics use a probe fitted for a fixed 5,000 steps and report its final step, so the 188 evaluation clips are never used to select that checkpoint. The qualitative examples in Figures 4 and S9 and the single-image results use an earlier fit of the same probe on the full model. That checkpoint was selected at step 4,500 by loss on these clips; its weighted held-out-video metrics lie within $0.4°$ of the final-step refit.

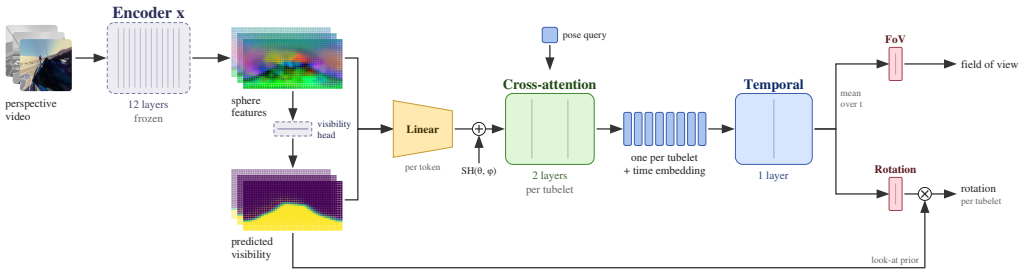

Figure S2: **Temporal Camera Probe.** A pose query pools each tubelet's frozen sphere features and predicted visibility, and a temporal transformer combines the eight pose tokens. Separate heads predict field of view and a correction to the visibility-centroid look-at rotation. Dashed blocks are frozen.

The single-image protocol preserves each benchmark image's aspect ratio. An image is duplicated across the two frames required for one tubelet; the probe therefore receives no temporal evidence. The feature-registration readout searches horizontal field of view from 20° to 160°, matches perspective and inferred sphere tokens, and applies Kabsch alignment to the matched rays.

Same-Model Target Retrieval. All 188 candidates are produced by the canonical 30k checkpoint using predicted visibility. Candidate targets are encoded by the same model's EMA branch. The out-of-view test uses ground-truth visibility only after inference to select scoring positions. The direction-controlled test reserves 32 clips to estimate separate per-token means for the predictor and target branches, then evaluates the remaining 156 clips. Scene retrieval holds direction fixed; azimuth retrieval holds scene identity fixed and ranks 28 cyclic longitude shifts. Confidence intervals are obtained by bootstrap resampling clips.

Predict-Then-Observe. For each clip, Pass A remains at the initial pose while Pass B follows one fixed trajectory. Of the 156 evaluation trajectories, nine reveal fewer than the required 32 previously hidden tokens, leaving 147 eligible clips. Branch- and direction-specific means are estimated on the 32 calibration clips and subtracted before scoring. Before this centring, the unconditional directional mean alone reaches 0.950 cosine, showing why the branch- and direction-specific means must be removed.

TAPVid360. The benchmark supplies 32 frames, whereas SphereJEPA consumes 16. We uniformly subsample 16 inputs over the complete benchmark interval, producing eight tubelet-level rotations, and use SLERP to interpolate the rotation chain back to all 32 timestamps before propagating query directions. DINOv3 and V-JEPA 2.1 controls use the same Kabsch rotation-chain readout with each backbone's native preprocessing. SphereJEPA and these two constructed frozen-feature baselines are not tuned on TAPVid360.

S4 Additional Results

Prediction Fidelity. Figure 3 shows that the predicted features preserve broad scene structure but are smoother than the EMA targets, particularly outside the camera view. The visualisation compresses 768-dimensional features to three PCA components, averages over eight temporal tubelets and enlarges a 28×56 spatial grid. These operations make the loss of local variation more visible, but the targets use the same visualisation and retain finer structure.

SphereJEPA predicts one feature per direction and minimises its per-token ℓ_1 distance to one EMA target. Many unseen scenes are consistent with the same perspective video, so this objective favours a central feature estimate rather than one detailed completion. Ordinary V-JEPA masking is more strongly constrained by visible content from the same perspective clip, and feature prediction can discard unpredictable appearance detail. In its qualitative analysis, V-JEPA visualises predicted features through a separately trained conditional diffusion decoder (Bardes et al., 2024). 360Anything instead samples decodable image or video latents from pretrained diffusion models (Wu et al., 2026).

Table S2: **Same-Model Target Retrieval.** Top-1 accuracy (%) when matching each prediction to its ERP target among N_{cand} candidates; brackets are 95% bootstrap confidence intervals. Pools below 188 average 200 random partitions, and chance is $1/N_{\text{cand}}$. Predicted visibility controls routing.

Scored tokens	N_{cand}=4 (↑)	N_{cand}=32 (↑)	N_{cand}=64 (↑)	N_{cand}=188 (↑)
All sphere tokens	99.0 [98.3, 99.6]	94.6 [92.1, 96.9]	92.3 [88.9, 95.3]	87.8 [83.0, 92.0]
Out-of-view tokens	98.9 [98.1, 99.5]	93.7 [90.8, 96.2]	91.2 [87.6, 94.5]	86.7 [81.9, 91.5]
Visible tokens	99.3 [98.9, 99.7]	95.5 [93.2, 97.6]	93.4 [90.3, 96.2]	89.4 [84.6, 93.6]

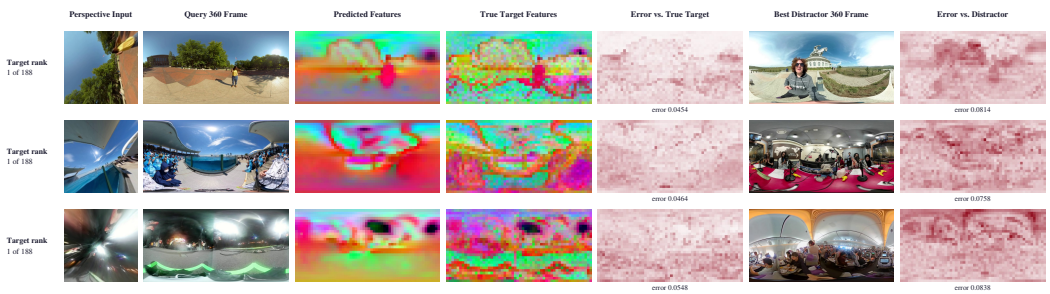

Figure S3: **Same-Model Target Retrieval Examples.** Each hand-selected row shows the perspective input, ERP target, predicted and target sphere features, prediction error, and closest distractor. All three rank the correct target first among 188; aggregate results use every query. Error scales are shared within each row.

This can produce sharp, plausible detail without guaranteeing that it matches the true hidden scene. The retrieval tests in Section 4.2 show that SphereJEPA's smoother out-of-view features still retain scene and azimuth information, but do not establish recovery of individual objects or their exact arrangement.

Same-Model Target Retrieval Results. A JEPA's own prediction error can serve as a label-free evaluation signal: in the violation-of-expectation protocol of Garrido et al. (2025), V-JEPA assigns higher error ("surprise") to physically impossible continuations of the *same* scene. Our setting predicts a whole surrounding scene rather than a continuation, so no matched impossible counterpart exists; we adapt the idea into a *cross-scene retrieval* test. Given a context, the model should assign lower prediction error to the target from the same scene than to targets from other scenes. We instantiate this over 188 clips whose source videos do not appear in the training corpus. For each perspective-video query, we compute the cosine prediction error between SphereJEPA's predicted sphere features and each candidate ERP target's features, then rank the matching target against $N_{\text{cand}} - 1$ distractors. The predictor is gated by the model's predicted visibility, not the ground-truth mask. Where the visibility head wrongly marks an out-of-view direction as visible, the scored prediction therefore blends in the context head's output, so the out-of-view mode measures the complete inference pipeline rather than a pure prediction-head readout. We report candidate pools of size $N_{\text{cand}} = 4$, 32, 64, and 188, the last ranking every held-out clip against all others.

Table S2 reports top-1 retrieval rates with bootstrap confidence intervals over queries. At $N_{\text{cand}} = 4$ the correct ERP target is top-ranked for 99.0% of queries using all sphere tokens. With the full 188-clip pool, it remains top-ranked for 87.8% of queries (95% CI $[83.0, 92.0]$), compared with 0.53% chance. Its median rank is 1, and it is among the top ten candidates for 98.4% of queries. Scoring only out-of-view tokens gives 86.7% at $N_{\text{cand}} = 188$ with median rank 1, so the result cannot be explained solely by matching features inside the crop. Pairwise accuracy in this out-of-view full-pool test is 99.6%. Figure S3 visualises representative queries, their predicted and target features, and the closest distractor under the same score used by the quantitative evaluation.

Direction-Controlled Retrieval. The full-pool test shows that a prediction is specific to its source video, but does not establish that the video-specific information has been placed at the correct longitudes. Every sphere token always denotes the same tubelet, latitude and longitude, and therefore

carries a consistent positional signal. Across videos, the predictor and EMA-target branches can each develop a characteristic average feature at every token position. If these features are compared directly, the unshifted target can score well simply because its position-dependent averages line up with the prediction. The azimuth test would then reward matching token coordinates rather than placing scene-specific content at the correct directions.

We suppress this shortcut using a separate calibration subset. Of the 188 held-out clips, 32 are used only to estimate a per-token feature mean separately for the predictor and EMA-target branches. The means are estimated separately because the two branches have different average feature signatures. We subtract the corresponding mean from each branch, then ℓ_2-normalise each residual token and evaluate the remaining 156 clips. This suppresses mean directional priors, such as average features associated with sky or ground, so retrieval emphasises the part of each token that varies with the input video.

The model receives the complete 16-frame perspective video: the observed crop is the context for completion and is not removed from the input. Predictor routing uses the model's predicted visibility. Ground-truth visibility is used only after inference to assign zero weight to sphere tokens that lie inside the perspective view. For query q, target candidate m, token index k, solid-angle weight $w_k = \cos\phi_k$, and ground-truth visibility $m_{q,k}^{\mathrm{vis,gt}}$, the retrieval distance is

$$\delta_{\mathrm{ret}}(q,m) = 1 - \frac{\sum_k w_k (1 - m_{q,k}^{\mathrm{vis,gt}}) \left\langle \tilde{\mathbf{z}}_{q,k}^{\mathrm{pred}}, \tilde{\mathbf{z}}_{m,k}^{\mathrm{tgt}} \right\rangle}{\sum_k w_k (1 - m_{q,k}^{\mathrm{vis,gt}})}, \tag{S7}$$

where the tildes denote separately mean-centred and normalised predictor and target features. The index k spans all eight temporal tubelets and all spherical positions. The scene test ranks all 156 target videos at fixed sphere directions. The azimuth test holds the target scene fixed and ranks 28 cyclic longitude shifts, spaced by 12.86°. Thus the former asks whether the prediction identifies the scene, while the latter asks whether its features occupy the correct spherical longitudes.

Figure S4 reports the aggregate results. At fixed directions, the correct scene ranks first for 85.9% of queries (95% CI $[80.8, 91.0]$), compared with 0.64% chance. Within the correct scene, the unshifted target ranks first for 88.5% of queries ($[83.3, 93.0]$), compared with 3.57% chance. Both decisions are correct for 76.9% of queries ($[69.9, 83.3]$), against 0.023% chance. This intersection is not a single 4,368-way ranking. Shuffling the perspective context reduces scene and joint accuracy to 0%. Shuffled-context azimuth accuracy is 14.1%, above chance, but its median angular error is 77.1°, compared with 0° for the true context. The residual shuffled accuracy indicates that a direction-specific bias remains after mean subtraction; it is not sufficient to identify the scene or make the joint decision. An unconditional per-direction mean is identically zero after this centring, so all of its cosine candidates tie at chance. The true-context results and shuffled controls show that predictions contain scene-specific information at the correct azimuth, although a small residual directional bias remains after centring. They do not establish object-level recovery: the targets are EMA features, azimuth is sampled at 12.86° intervals, and each validation clip contributes one deterministic crop and one temporal sample.

Figure S5 makes the scene and azimuth decisions concrete for one joint top-1 example. Grey regions, outlined in red, were visible in the perspective input and receive zero weight. The matching scene has cosine distance 0.412, compared with 0.748 for the closest of 155 distractors. Rotating the true target by 90° or 180° raises the distance to 0.648 and 0.654, and the unshifted target ranks first among 28 azimuth shifts. The clip has 81.7% out-of-view tokens. It is an illustrative successful case; the aggregate claims use all 156 evaluation clips.

Predict-Then-Observe Examples. Figure S6 stratifies prediction–observation agreement by angular distance from Pass A's visible boundary.

Figure S7 applies the locked independent-observation protocol of Figure 5 to two further held-out clips.

Camera Readout Results. Table S3 reports both camera readouts for the full model and the two matched controls of Table 4. The analytical rows repeat the visibility readout of Section 4.1. The probe rows fit the temporal probe of Figure S2 to each frozen backbone with the same fixed schedule

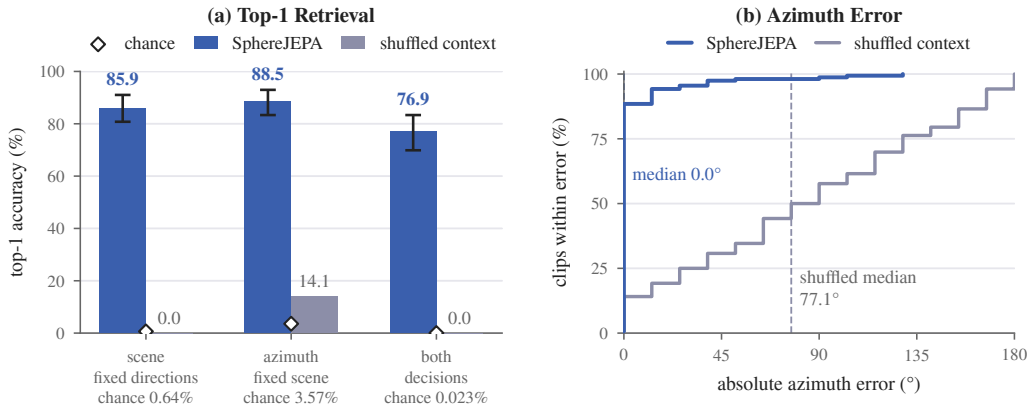

Figure S4: **Direction-Controlled Out-of-View Retrieval.** (a) Scene, azimuth and joint top-1 accuracy after direction-wise centring; bars are 95% clip-bootstrap intervals and diamonds show chance. (b) Cumulative absolute azimuth error. Predicted visibility controls routing; ground-truth visibility only selects out-of-view tokens.

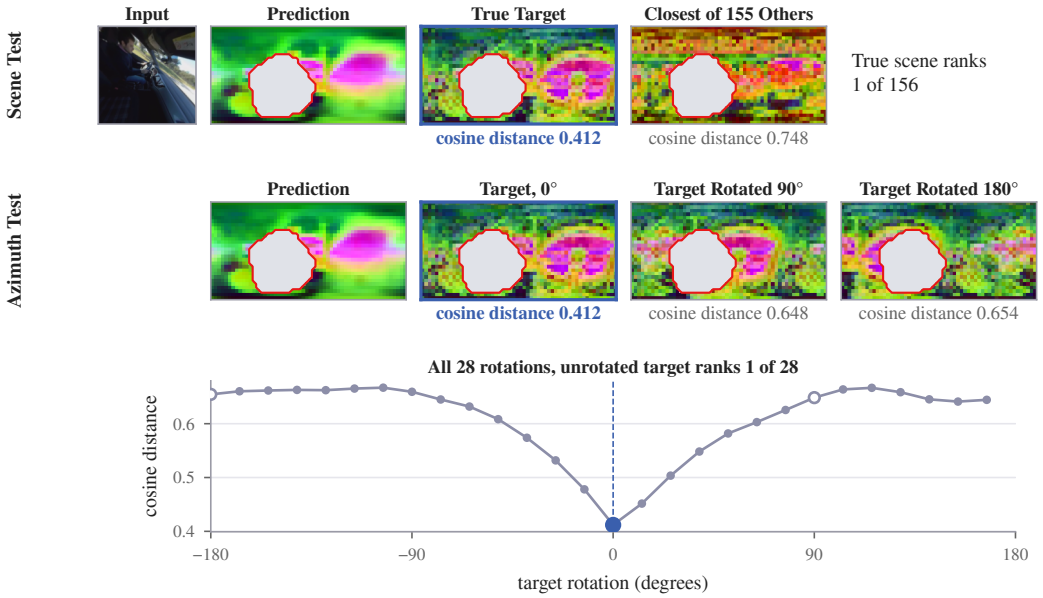

Figure S5: **Direction-Controlled Retrieval Example.** The prediction is compared with the correct scene and closest distractor (top), then with rotated versions of the correct target (middle and bottom). Grey regions were visible and are excluded from scoring. This is an illustrative success; aggregate results use the full evaluation set.

and report its final step, weighted across the four trajectory families in the proportions of the training mixture. On the full model the probe estimates field of view within $3.84°$ and relative rotation within $8.75°$. Roll and absolute rotation are coarser at $13.83°$ and $18.47°$. The probe does not favour the full construction. The no-correspondence backbone is slightly better on every metric. The online-correspondence backbone supports field of view but not roll, whose error stays near $60°$, the error of an uninformed guess over the $\pm 120°$ training range, throughout fitting. Camera parameters are therefore recoverable from the frozen features with or without the frozen correspondence target. The frozen target does not improve camera readout; its clearest benefit is transferable directional correspondence. Figure 4 shows the visibility and camera readouts on four validation clips.

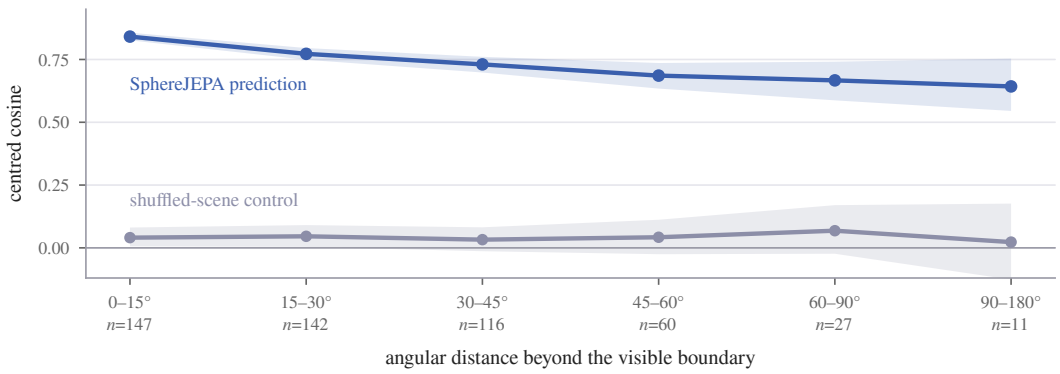

Figure S6: **Prediction–Observation Agreement Beyond the Image.** Centred cosine agreement with later observations remains high with increasing distance from the visible boundary. Bands are 95% clip-bootstrap intervals; counts give eligible clips in each bin.

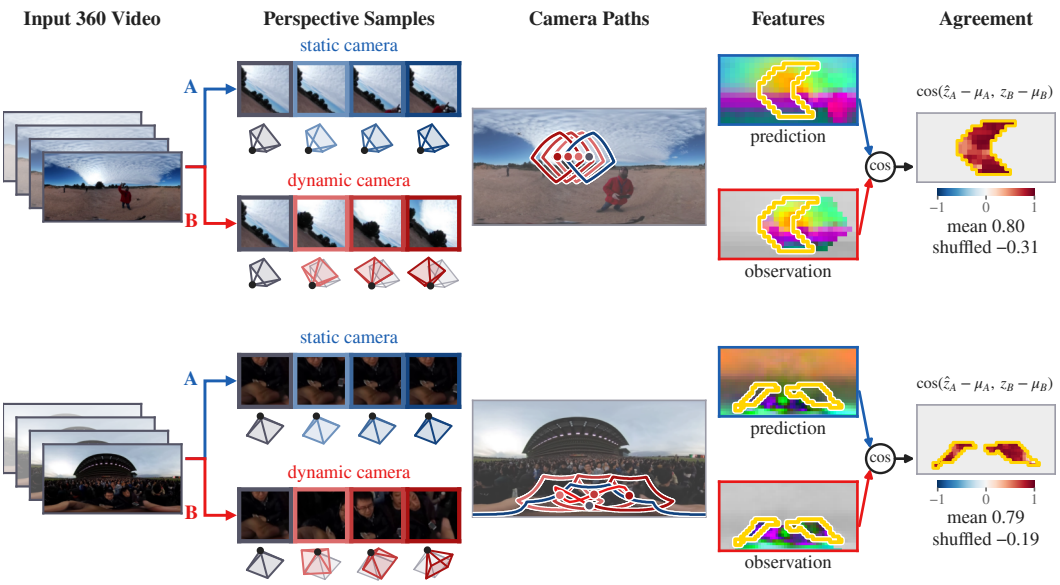

Figure S7: **Additional Predict-Then-Observe Examples.** Coastal (top) and concert (bottom) scenes evaluated using the protocol in Figure 5. Aggregate claims use the full eligible evaluation set.

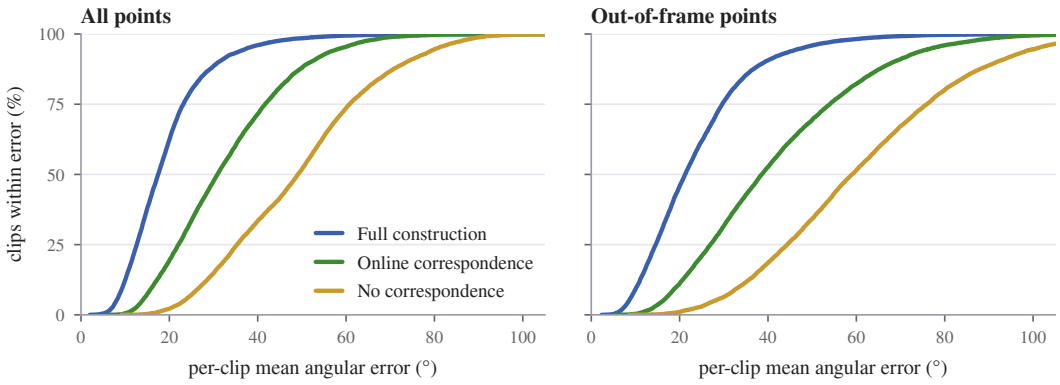

Figure S8: **Correspondence-Loss Ablation Distributions.** Per-clip TAPVid360 angular-error distributions for the full model and matched controls. The left panel includes all points and the right only out-of-frame points; curves farther left are better.

Table S3: **Camera Estimation on Held-Out Perspective Video.** The analytical readout uses predicted visibility; the temporal probe is fitted separately to each frozen backbone for 5,000 steps. Results use the same fixed trajectories and report angular errors in degrees.

Readout	Backbone	FoV (↓)	View Dir. (↓)	Roll (↓)	Abs. Rot. (↓)	Rel. Rot. (↓)	Vis. IoU (↑)
Analytical visibility	Full construction	8.91	9.57	–	–	–	**.636**
	No correspondence	8.91	9.44	–	–	–	.633
	Online correspondence	9.15	11.73	–	–	–	.620
Temporal probe	Full construction	3.84	9.60	13.83	18.47	8.75	–
	No correspondence	**3.10**	**8.43**	**12.74**	**16.38**	**7.73**	–
	Online correspondence	3.23	16.14	59.52	64.52	19.68	–

Table S4: **Zero-Shot Single-Image Camera Estimation.** Mean errors in degrees on the benchmarks used by 360Anything; lower is better. The first three datasets evaluate horizontal field of view, and the final two evaluate roll and pitch.

	FoV Error			Roll / Pitch Error	
Method	**NYUv2**	**ETH3D**	**iBims-1**	**MegaDepth2k**	**LaMAR2k**
360Anything (Wu et al., 2026)	**3.90**	**5.68**	**5.21**	**0.87 / 2.56**	**0.68 / 1.23**
Upright-camera prior	–	–	–	3.97 / 22.83	6.66 / 20.37
SphereJEPA, temporal probe	18.23	10.27	28.25	8.15 / 8.73	10.72 / 10.55
SphereJEPA, feature registration	24.65	18.51	36.18	9.07 / 10.55	12.65 / 9.70

Correspondence-Loss Ablation Statistics. Removing correspondence raises TAPVid error by $30.14°$ (95% paired clip-bootstrap CI $[29.78, 30.48]$), while online correspondence remains $13.58°$ above the full model ($[13.35, 13.81]$). Removing correspondence reduces scene retrieval by 7.7 points ($[-12.8, -2.6]$), joint retrieval by 7.1 points ($[-13.5, -0.6]$), and predict-then-observe agreement by .019 ($[-.027, -.012]$). Online correspondence reduces scene retrieval by 38.5 points ($[-47.4, -29.5]$) and joint retrieval by 35.9 points ($[-44.9, -26.9]$), while predict-then-observe agreement changes by only .004 ($[-.012, .020]$).

Single-Image Transfer. We next test whether these readouts transfer without adaptation to the public single-image calibration protocol used by 360Anything (Wu et al., 2026). Field of view is evaluated on NYUv2 (Silberman et al., 2012), ETH3D (Schöps et al., 2017) and iBims-1 (Koch et al., 2020). Roll and pitch are evaluated on the MegaDepth2k (Li & Snavely, 2018) and LaMAR2k (Sarlin et al., 2022) subsets prepared by GeoCalib (Veicht et al., 2024). Each image is evaluated independently at its original aspect ratio. We duplicate it only across the two frames consumed by one tubelet, which supplies one temporal token and no additional visual evidence.

We apply the temporal probe without fitting it to any of these datasets. We also test a label-free feature-registration readout. It matches perspective tokens to inferred spherical tokens, searches horizontal fields of view from $20°$ to $160°$ and uses Kabsch alignment to recover orientation. This is the feature-space counterpart of 360Anything's panorama search, although its coarse semantic tokens provide a different matching signal from dense generated pixels.

Table S4 reports mean errors and Figure S9 shows representative inputs and estimates from this protocol. The external result is substantially weaker than the held-out-video result and does not make SphereJEPA competitive with a dedicated calibration method. Nevertheless, both readouts improve pitch estimation over the upright-camera prior. The temporal probe reduces mean pitch error from $22.83°$ to $8.73°$ on MegaDepth2k and from $20.37°$ to $10.55°$ on LaMAR2k relative to an upright-camera prior. Feature registration gives $10.55°$ and $9.70°$ without a learnt camera readout. Across both readouts and datasets, median pitch error is $4.66°$–$6.45°$, so the larger means arise from a substantial failure tail. SphereJEPA has therefore learnt transferable information about gravity and camera elevation, while field of view and roll transfer poorly. The transfer setting differs from training in three ways. During pre-training, we sample the base field of view uniformly from $30°$ to $150°$, pitch from $-80°$ to $80°$ and roll from $-120°$ to $120°$, so unusual cameras are more common

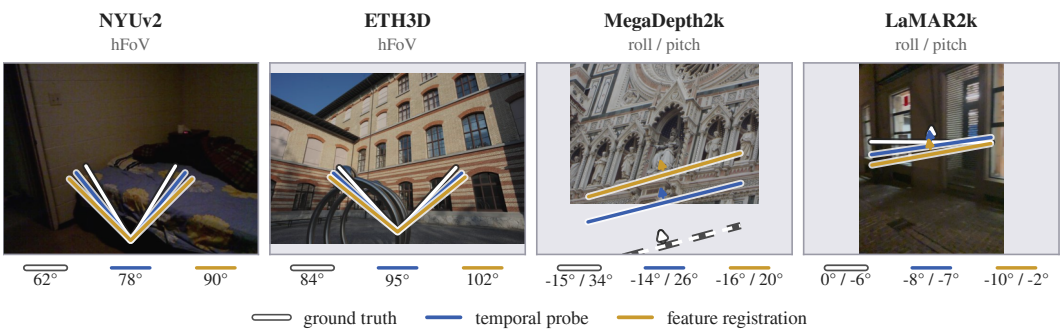

Figure S9: **Zero-Shot Single-Image Camera Estimates.** White shows ground truth, blue the temporal probe, and gold feature registration. Field-of-view rows show horizontal FoV; orientation rows show roll and pitch. Each example is median by the two SphereJEPA readouts' combined error within its dataset.

than in these photographic datasets. The video probe is fitted to eight tubelets from square rendered clips, whereas each benchmark image supplies one tubelet at its original aspect ratio. The appearance also shifts from perspective crops of web 360° video to conventional photographs. The external result therefore combines camera, temporal and appearance shifts. Section S5 describes changes that would separate them.

S5 Limitations and Future Work

Representation Scope. SphereJEPA indexes features by direction from a moving origin. It contains no explicit depth or 3D coordinate field, and we have not tested whether metric depth can be decoded from its features. The moving-camera training videos may allow the representation to encode depth, translation and parallax cues. The current experiments therefore do not establish explicit 3D reconstruction or recovery of every out-of-view object. Latent world models additionally learn dynamics that can be rolled forward for control or planning (Ha & Schmidhuber, 2018; Hafner et al., 2025). The spherical index could organise such a state, but the present model has no action-conditioned dynamics, temporal rollout or planning objective. It also predicts features rather than pixels. Panorama generators such as 360Anything (Wu et al., 2026) remain the appropriate choice when the required output is an image.

Camera Training. A better camera distribution would separate base pose from relative motion. Most samples could follow an upright-biased pose and focal-length distribution, while a smaller tail retained the broad rotations and fields of view needed for coverage. Variable aspect ratios and single-tubelet examples would reduce the remaining input mismatch. A matched single-tubelet probe on the current frozen backbone would test whether probe mismatch explains the transfer gap. If not, the revised distribution should be used during pre-training as well as probe fitting.

Mixed Perspective and Spherical Training. The current objective uses perspective patches as context and applies its prediction targets on the sphere. An additional forward pass could preserve the standard V-JEPA 2.1 task by masking perspective patches and predicting their EMA features at the same planar space-time positions. This would allow ordinary perspective video to train the shared encoder, while paired 360° video retained the spherical prediction, visibility and correspondence losses. An ablation over perspective-only, spherical-only and mixed batches would test whether the larger corpus improves the representation without weakening crop-to-sphere binding or out-of-view completion.