

Can 4D Foundation Models Remember?

Guangzhao He
Cornell University
gh466@cornell.edu

Hadar Averbuch-Elor*
Cornell University
hadarel@cornell.edu

Wei-Chiu Ma*
Cornell University
wm347@cornell.edu

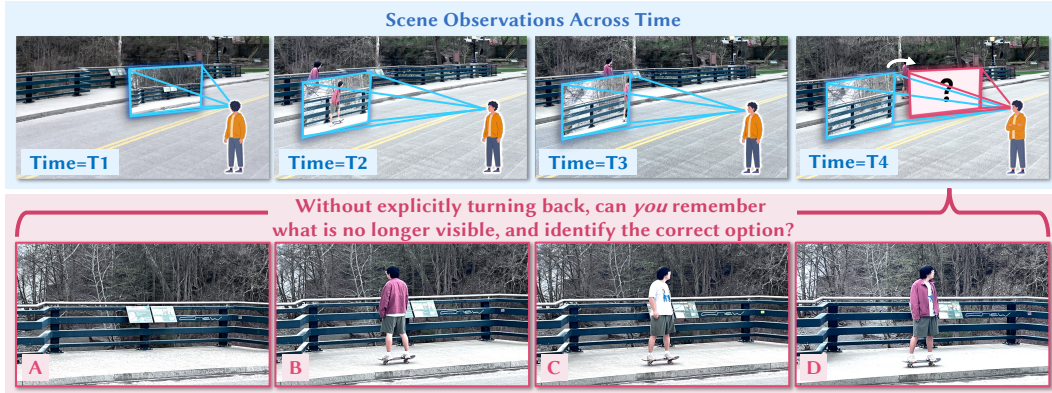


Figure 1: As humans, we effortlessly maintain a dynamic mental representation of the world: after briefly observing a scene (top), we can infer what lies beyond our current field of view and anticipate how objects continue to move and appear over time. Test which option best matches *your* expectation (bottom). Were you able to quickly identify that: (a) violates object permanence, (b) breaks motion continuity, and (c) fails to preserve the person’s appearance? In this work, we explore whether 4D foundation models can remember the 4D world as we do.

Abstract

Perceiving and remembering the visual world is fundamental to navigating and interacting with our environment. Current 4D foundation models, such as camera-controllable video models or 4D reconstruction models, can perceive and reconstruct dynamic environments, but how well they remember what they have perceived remains an open question. Existing benchmarks largely rely on pixel-level metrics and lack ground truth for objects once they leave the field of view, making them unable to evaluate visual memory in an object-centric manner against references. To fill this gap, we introduce PERSISTBENCH, a dataset and metric suite that leverages 360° videos as omniscient ground truth and proposes three evaluation aspects: object permanence, motion continuity, and appearance preservation. Evaluating various models across diverse categories reveals that current models can only maintain short-term consistency that degrades significantly once objects leave the field of view. Our findings highlight the gap between current model capabilities and robust visual memory (“seeing is not remembering”), providing guidance for future development of 4D foundation models. Dataset and code are available on the project page: <https://guangzhaohe.com/persistbench>.

1 Introduction

“... a foolish extravagant spirit full of forms, figures, shapes, objects, ideas, apprehensions, motions, revolutions; these are begot in the ventricle of memory.”

— William Shakespeare, *Love’s Labour’s Lost*

*Denotes equal advising.

Memory is a fundamental component of human cognition [1]. It enables us to maintain and reason about the state of the world beyond immediate perception. Early theories, dating back to Aristotle [66], described memory as static impressions, akin to marks on a wax seal. Later perspectives in the medieval and early modern periods, reflected in the writings of William Shakespeare [64], expanded this view into a richer repository of mental content that encompasses not only appearance, but also motion and temporal evolution. Modern cognitive science studies further refined this perspective, showing that humans develop dynamic *object-centric* representations that encode how entities *persist*, *move*, and *evolve* over time [70, 73, 10]. Such representations enable humans to reason about unobserved states of the world: for example, after observing a person riding a skateboard, we can anticipate how they continue to move and appear even after leaving the field of view (Fig. 1).

This naturally raises the question: do existing *4D foundation models*, more specifically, 4D vision models that reconstruct or generate dynamic 3D scenes, such as GEN3C [61] and 4DGT [86], possess similar capabilities, allowing them to remember and extrapolate beyond what they observe? Despite its apparent simplicity, answering this question is surprisingly non-trivial, for **two** critical reasons. **First**, there is a fundamental lack of real-world data to effectively validate out-of-view memory in dynamic scenes. Standard videos possess a limited field of view. When an object moves out of frame, its ongoing state is no longer captured. Without multi-viewpoint “ground truth” of these unobserved dynamics, we cannot quantitatively measure if a model correctly anticipates an object’s evolution. **Second**, existing metrics predominantly focus on perceptual quality and pixel-level consistency [17, 37, 94, 38], falling short of assessing true memory behavior such as semantic and dynamic persistence. Thus, while there has been rapid progress in the development of 4D foundation models, whether these systems truly exhibit memory-like capabilities remains elusive.

With these challenges in mind, we introduce PERSISTBENCH, to our knowledge, the *first* benchmark that provides *real-world, ground truth observations* for systematically evaluating visual memory in 4D foundation models. Our core insight is to construct real-world scenarios in which parts of the dynamic scene are unobserved to the 4D foundation models. Then we probe the models to assess whether they can accurately recover the hidden state of the world beyond its field of view. To overcome the data scarcity challenge, we leverage the relatively untapped resource of 360° videos [76, 48]. Unlike standard video inputs, omnidirectional videos provide full access to the environment, allowing us to define both observed inputs and withheld ground-truth reference trajectories (see Fig. 3). Furthermore, we propose a suite of object-centric metrics that quantify three fundamental aspects of visual memory: *object permanence* (e.g., does the rider in Fig. 1 continue to exist after leaving the field of view, or is it forgotten as in option A?), *motion continuity* (e.g., does the rider follow a consistent trajectory, or deviate as in option B?), and *appearance preservation* (e.g., does the rider’s appearance remain consistent, or if not, to what extent does it change?). Together, these components enable a direct and principled evaluation of whether models truly maintain coherent, persistent representations of dynamic scenes.

Using PERSISTBENCH, we conduct extensive evaluations of various 4D foundation models spanning multiple categories, and identify a problem that prior work has largely overlooked: current 4D foundation models lack persistent visual memory. Our experiments show that every model performs substantially worse once the target object becomes invisible by leaving the input field of view. This reveals a critical gap between “seeing” and “remembering,” which we attribute to a training-data bias: current 4D models are trained predominantly on videos where target objects remain visible throughout, providing little supervision for memory. We further analyze how different architectural design choices influence this behavior, suggesting potential directions for developing 4D foundation models with more persistent visual memory.

2 Related Work

4D foundation models. The rise of visual foundation models capable of perceiving and reasoning in 3D space [45, 92, 80, 42] has spurred increasing interest in models that operate over temporally-evolving 3D scenes. This includes 4D reconstruction models, which process input video frames and reconstruct the underlying dynamic 3D scene. These models often use implicit representations [79, 15], encoding dynamic scenes as latent sequences decoded by a neural renderer, while some prior work pursue explicit modeling [86, 92] that yields directly renderable representations such as Gaussian Splatting [40]. Unlike 4D reconstruction models, 4D generative models [29, 4, 90, 61, 81, 36, 23, 33, 88, 13, 14]—often referred to as *world models*—aim to synthesize

novel views and timesteps given input observations, typically building upon video generation backbones [50, 77, 89, 32]. In particular, Video-to-360° models [48, 84, 71, 20, 43] predict panoramic video from perspective inputs, and can be probed by cropping perspective views at any desired region. In this work, we probe the internal dynamic scene representations of these various 4D foundation models through their rendering interface. **We leave for future work models that require tailored evaluation setups such as those that do not accept video as conditioning input [27, 25, 30], those that output non-renderable 4D representations [78, 80] (e.g., point clouds), and those that operate in the language domain [2, 47] (e.g. Vision-Language Models).**

Benchmarking visual memory. Evaluating how well vision models retain and reason about what they have observed is essential for understanding their capacity to act as reliable perception systems in dynamic environments. Vision-language model benchmarks such as VSI-Bench [87] and VSI-SUPER [72] probe spatial memory through question-answering about object distances and positions in egocentric videos, reducing visual memory evaluation to simple textual descriptions that fail to capture the rich visual information of the real world. Reconstruction benchmarks [6, 41, 54, 52, 24, 44] can measure memory implicitly through photometric accuracy (PSNR, SSIM [82], LPIPS [91]) on held-out views. However, by only evaluating rendering fidelity, they fail to capture the object-centric nature of visual memory. Among generative model benchmarks [37, 38, 94, 35, 26, 46, 83, 19, 9], VBench [37, 38, 94], WorldScore [17], and WCS [58] mainly evaluate pixel-level consistency, while concurrent efforts [49, 18], perhaps most related to our setting, have begun to realize the importance of object permanence after moving out of view. However, most of these benchmarks are fundamentally *reference-free*: they evaluate visual plausibility or internal consistency without verifying whether a model faithfully retains the appearance, location, and motion state of specific observed entities. Our benchmark instead enables *reference-based* evaluation by leveraging real-world 360° video as ground truth while providing only perspective crops as input, offering the first systematic evaluation of object-centric visual memory across multiple aspects with real-world data.

Visual memory in cognitive science. Research in human cognition [69, 68] suggests that visual memory is fundamentally object-centric, composed of multiple perceptual competencies that emerge over development. Early on, humans acquire *object permanence*—the ability to represent objects as continuing to exist even when they are occluded [53]. Violation-of-expectation studies demonstrate that even young infants exhibit surprise when objects behave inconsistently with this principle [5]. In parallel, humans develop sensitivity to *spatiotemporal continuity*, expecting objects to follow coherent trajectories through space and time [67, 70, 73]. This capability supports tracking and prediction of future states [31]. In contrast, *appearance-based* representations emerge later: infants initially rely on spatiotemporal cues and often fail to individuate objects based on featural differences under occlusion, with robust appearance-based reasoning developing only subsequently [85, 10]. Together, these findings indicate that visual memory is not a monolithic ability, but rather a hierarchy of object-centric representations. This motivates our approach of decomposing visual memory into distinct, measurable components for evaluating computational models.

3 Quantifying Visual Memory

Although there is a common intuition that certain 4D foundation models remember better than others, no prior work has *systematically* quantified this phenomenon. Many fundamental questions remain unanswered: do these models know an object still *exists* after it leaves the frame? Do they maintain *consistent motion* for moving objects beyond the camera’s view? Can they preserve the object’s *appearance* when revisited from a new viewpoint?

This gap in evaluating visual memory stems from two main issues. First, there is no formal definition of what it means for a model to *remember*. Second, existing datasets lack ground truth for regions beyond the camera’s view. In this section, we formally define visual memory, characterize its essential properties, and operationalize these properties into concrete computational metrics. We address the data challenge in Sec. 4.

3.1 Problem Formulation

Drawing on findings from cognitive science regarding how humans perceive and reason about the physical world [69, 68, 31], we define a model’s ability to *remember* as its capacity to (i) maintain

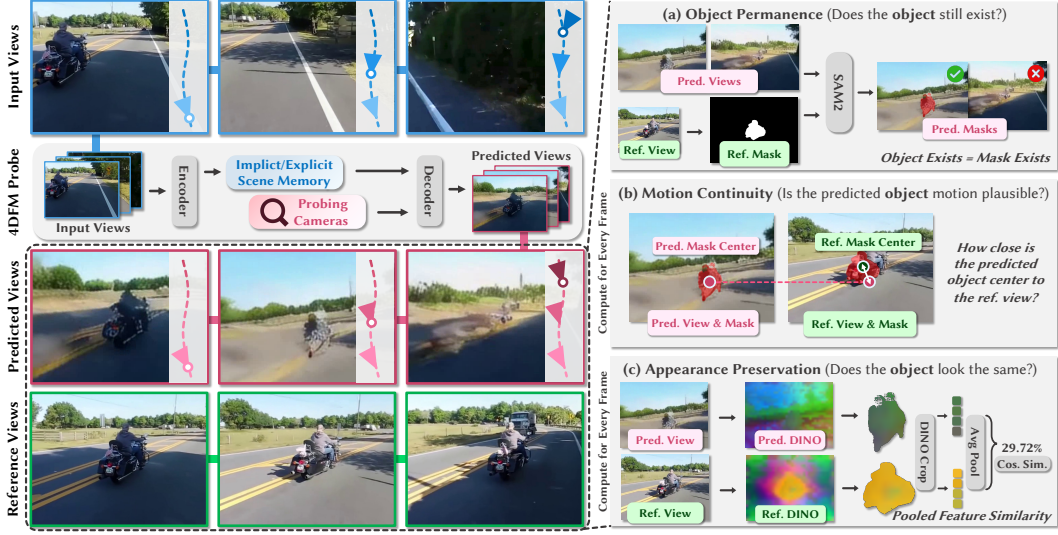


Figure 2: **Evaluating visual memory.** We construct input videos to 4D foundation models, where an observed object (motorcycle) leaves the field of view, and then probe the models with virtual cameras to render its internal memory. Using the ground-truth reference views, we evaluate three object-centric aspects of visual memory, as illustrated on the right.

the existence of observed objects, (ii) retain their spatial location and motion state, and (iii) preserve their visual appearance over time and across viewpoints not observed in the inputs.

Formally, given a set of input observations $\{\mathcal{I}_i\}_{i=1}^N$ and a 4D foundation model f , we first query the model to generate outputs $\{\mathcal{O}_t\}_{t=1}^T$ using a set of target probes $\{\mathbf{p}_t\}_{t=1}^T$:

$$\{\mathcal{O}_t\} = f(\{\mathcal{I}_i\}, \{\mathbf{p}_t\}). \quad (1)$$

We then quantify the model’s memory by comparing its generated outputs against ground-truth references $\{\mathcal{O}_t^{\text{ref}}\}_{t=1}^T$ across various criteria g :

$$\text{Score} = g(\{\mathcal{O}_i\}_{i=1}^T, \{\mathcal{O}_i^{\text{ref}}\}_{i=1}^T). \quad (2)$$

As we will detail in Sec. 3.2, g can be instantiated in several ways. For instance, it could measure how well the model preserves visual appearances.

While the inputs $\{\mathcal{I}_i\}$ and outputs $\{\mathcal{O}_t\}$ of modern 4D foundation models can take various forms, we assume both to be videos in this work, as this setup encompasses a wide range of models (see Sec. 6). However, we note that this formulation easily extends to other modalities (e.g., point clouds).

We make no assumptions about the model’s internal architecture and interact with it solely through the target probes, which we define as camera trajectories $\{\mathbf{p}_t \in \mathbb{SE}(3)\}$. Intuitively, by setting these probes to novel viewpoints different from the original input, we force the model to generate scenes that could be completely invisible in the input. This allows us to validate whether the model has truly built a persistent, object-centric 4D representation or is merely memorizing 2D pixels.

3.2 Operationalizing Visual Memory Evaluation

Effectively quantifying visual memory requires carefully designed evaluation criteria g . Unfortunately, existing metrics focus on pixel-level accuracy [17, 37], conflating a model’s memory capability with its rendering fidelity. To decouple these concepts, we introduce three distinct yet complementary metrics designed to isolate and capture the fundamental properties of visual memory.

Object permanence. Our first metric evaluates whether an object observed in the input sequence continues to exist within the model’s underlying representation, even after leaving the original field of view. If a model possesses true object permanence, it should successfully render the object when queried with a novel viewpoint where that entity is expected to be visible.

To operationalize this, we evaluate the generated outputs $\{\mathcal{O}_t\}_{t=1}^T$ using a dual-verification mechanism. Given a reference mask of the target object from the input, we first utilize SAM2 [60] to track

the object across the generated frames. If a model truly “remembers” an object, when probed with the right viewpoint, it should consistently generate an entity with sufficient coherence to track.

However, because a tracker might continue following a spatial region even if its visual identity severely drifts or degrades, we further utilize a VLM to evaluate appearance fidelity [93, 21, 83]. We consider an object to be successfully preserved in a given frame only if it satisfies both criteria: the tracker produces a valid mask, and the VLM confirms the target object’s presence.

Finally, the object permanence score is defined as the percentage of valid frames across the sequence:

$$\text{Perm} = \frac{1}{T} \sum_{t=1}^T \mathbb{1}[\text{SAM}_t \wedge \text{VLM}_t], \quad (3)$$

where $\text{SAM}_t \in \{0, 1\}$ and $\text{VLM}_t \in \{0, 1\}$ are binary indicators of the tracker’s and VLM’s response at frame t , and $\mathbb{1}[\cdot]$ is the indicator function.

Motion continuity. Besides permanence, a model must continuously update an entity’s state based on observed dynamics, regardless of whether the object remains within the field of view or not. Static objects should remain stationary, while dynamic objects should naturally continue their motion based on prior trajectory patterns.

To assess this capability, we evaluate the spatial alignment between the predicted and ground-truth object locations over time. Building upon the tracking masks extracted during the permanence evaluation, we first compute the ℓ_2 distance between the centroids of the predicted object mask $\mathbf{c}_t^{\text{pred}}$ and the reference object mask $\mathbf{c}_t^{\text{ref}}$ at each frame t .

Because an absolute pixel distance implies different error severities depending on the object’s scale, we further normalize this distance using the average area of the reference object $\bar{A}^{\text{ref}}$. Finally, following [62], we define motion continuity as an exponentially decayed, scale-normalized distance:

$$\text{Cont} = \frac{1}{T} \sum_{t=1}^T \exp\left(-\frac{\alpha \|\mathbf{c}_t^{\text{pred}} - \mathbf{c}_t^{\text{ref}}\|_2}{\sqrt{\bar{A}^{\text{ref}}}}\right), \quad (4)$$

where $\alpha > 0$ is a temperature hyperparameter. Empirically, we set $\alpha = 1.0$. In practice, since Cont is designed to capture motion continuity which assumes the object exists, we compute it over frames where the object is successfully tracked (i.e., $\text{SAM}_t = 1$). **We examine alternative motion continuity definitions and discuss the ambiguity of reference-based motion evaluation in Supp. Mat., Secs. C.3 and E.4, respectively.**

Appearance preservation. Our final metric (App) evaluates how well a model preserves an object’s visual identity. Compared to object permanence, it provides a more fine-grained estimate of the output’s visual faithfulness. We focus on measuring feature similarity [51] between the generated entity and the ground-truth observation, as these features capture rich visual semantics that are highly robust to scale and lighting variations.

Specifically, at each time step t , we extract DINOv2 [51] feature maps $\mathbf{F}_t^{\text{pred}}$ and $\mathbf{F}_t^{\text{ref}}$ from the generated and reference frames. To isolate the target object, we apply the corresponding object masks and perform spatial average pooling to produce a single feature vector for each. Finally, we compute the appearance preservation score as the average cosine similarity between these pooled feature vectors across the generated sequence:

$$\text{App} = \frac{1}{T} \sum_{t=1}^T \cos\left(\phi(\mathbf{F}_t^{\text{pred}}), \phi(\mathbf{F}_t^{\text{ref}})\right), \quad (5)$$

where $\phi(\cdot)$ denotes the masked average pooling operation. By comparing global semantic features rather than raw pixels, this metric remains invariant to minor pose shifts and partial occlusions. Similar to Cont, we compute App over frames where the object is successfully tracked, so that the metric captures appearance fidelity regardless of how long the object exists in the model’s memory. **Ablations of the object permanence and appearance score designs, together with a comparison of tracked-frame and all-frame aggregation, are provided in Supp. Mat., Secs. C.2, C.1, and C.4.**

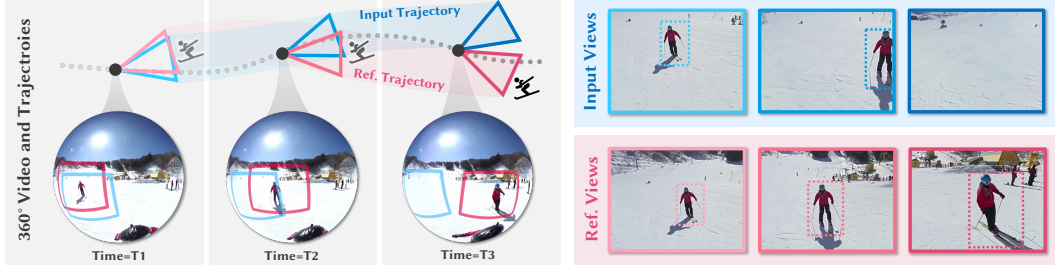


Figure 3: **Evaluating with 360° video.** Using 360° video enables arbitrary camera projections of the same underlying scene. Given one 360° video and a target object, we generate two camera trajectories (input and ref.). In the input trajectory, the target object is visible during the first half and leaves the view in the second. In the reference (Ref.) trajectory, the object remains in frame throughout, providing a ground-truth reference even after it has left the input view.

4 Evaluating Visual Memory with 360° Video

4.1 360° Video as Natural Visual Memory Testbed

Now that we have established how to quantify visual memory, the remaining challenge is sourcing the right data for evaluation: *on which data can we most rigorously test these 4D foundation models?*

Ideally, such data should capture not only real-world dynamics but also provide comprehensive multi-view observations. This ensures we can rigorously validate a model’s predictions even after an entity exits the original field of view. Unfortunately, standard videos are not sufficient as once an object leaves the frame, its ongoing state is no longer captured.

To overcome this data challenge, we leverage the relatively untapped resource of 360° videos. Unlike common perspective recordings, a 360° video captures the full surrounding environment at every timestep (Fig. 3). It serves as an omniscient record that can be freely cropped into normal perspective views at any desired camera angle *post-capture*. This flexibility allows us to construct input videos by specifying arbitrary camera trajectories, strictly controlling exactly what the model observes. Crucially, for any object that is not physically occluded, we can simply obtain its ground truth by directing a virtual camera toward its location at any point in time. This provides a reliable reference to evaluate whether the model retains its internal representation of the entity beyond direct observation. Next, we describe in detail how we leverage this boundless field of view to derive high-quality video pairs for memory evaluation.

4.2 Extracting Input-Reference Pairs

Given a 360° video and its camera poses, our goal is to extract a *pair* of perspective video sequences: an *input* sequence that serves as the model’s input, and a *reference* sequence that serves as the ground truth for evaluation. Both sequences are paired with their respective camera poses, which act as target probes for the generation process.

To effectively evaluate the three core properties of visual memory, we partition the input into visible and invisible segments. Specifically, we devise a camera trajectory sampling strategy where the object of interest is initially visible but leaves the field of view in later frames of the input sequence. In contrast, the object remains visible throughout the reference sequence. **We report the distributions of visible and invisible segment durations in Supp. Mat., Sec. A.5.** The 4D foundation model is tasked to synthesize the reference sequence using the input sequence as its only signal of the world’s state. If the model can render the object in the reference viewpoint accurately across both visible and invisible segments, we say it demonstrates true persistent visual memory. Our dataset construction pipeline consists of two steps:

Tracking objects of interest. We begin by identifying salient objects in the 360° video using a VLM labeler [56], which generates a text description for each object. Since our evaluation requires precise control over when each object enters and exits the camera’s field of view, we need to know its position at every timestep. We therefore use SAM3 [11] to segment and track all object instances throughout the video. One challenge is that 360° videos are stored in equirectangular format, where the scene wraps around at the left and right edges. Naively applying a tracker in this format fails

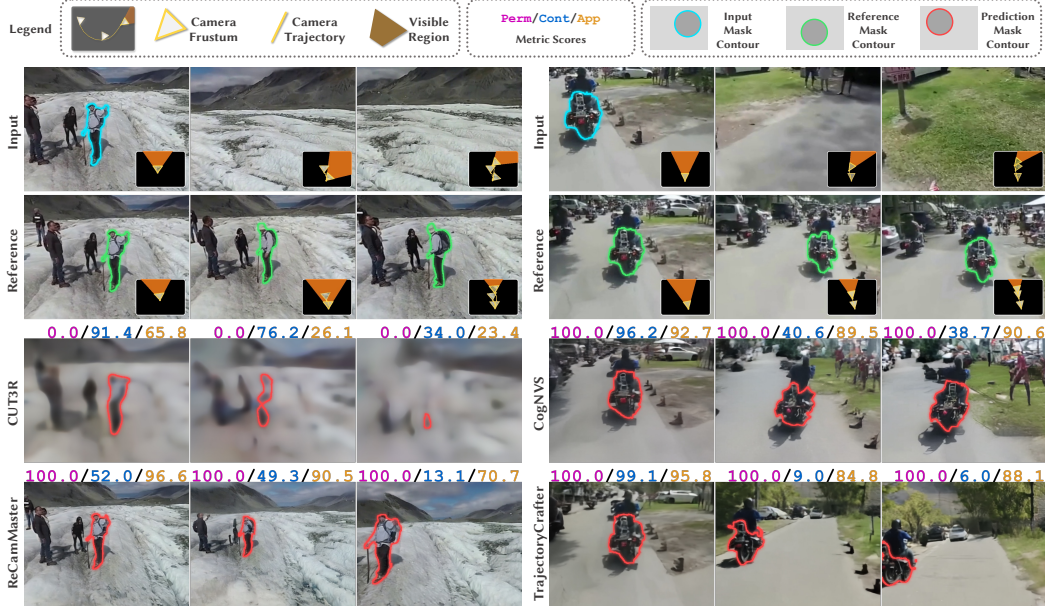


Figure 4: **Qualitative comparisons with per-frame metrics.** We show per-frame evaluation results for two cases, each from two representative models: CUT3R [79], ReCamMaster [4], CogNVS [13] and TrajectoryCrafter [90]. SAM2 tracking are highlighted. **Left (CUT3R):** our permanence score Perm correctly identifies cases where SAM2 maintains a valid track but the predicted object identity has degraded beyond recognition. **Left (ReCamMaster):** our motion continuity score Cont accurately reflects whether the predicted object motion agrees with the reference views. **Right (TrajectoryCrafter):** our appearance preservation score App robustly captures appearance consistency regardless of motion prediction quality. More results for all models are provided in Supp. Mat., Sec. G.

when objects cross this boundary, producing discontinuous tracks that cannot associate the same object across time. To resolve this, we horizontally duplicate the equirectangular canvas and run tracking on the augmented clip, ensuring that an object crossing one edge always has a continuous duplicate near the center. The resulting tracks are then filtered and mapped back to the original frame to produce seamless and persistent object trajectories.

Constrained camera trajectory sampling. Given object tracks, the next step is to come up with camera trajectories for the input and reference sequences such that they are physically plausible, free of sudden jitters, and satisfy our visibility constraints (i.e., each object enters and exits the field of view at the intended times). Since naively interpolating camera poses often results in abrupt motion or repeated pattern, we follow prior work in motion planning [59, 39, 63] and formulate this as a trajectory optimization problem. Specifically, we represent camera trajectories as B-splines [12] and optimize the control points using L-BFGS [8] to balance visibility constraints with kinematic smoothness, minimizing velocity, acceleration, and jerk. See Supp. Mat., Sec. A for more details.

Once we obtain the camera trajectories for both the input and reference sequences, we crop the corresponding perspective views from the 360° video. This yields paired video clips that serve as a rigorous testbed for our visual memory evaluation. An example video pair is shown in Fig. 3.

5 The PERSISTBENCH

Our final step is to leverage our carefully designed metrics (Sec. 3) and our data curation pipeline (Sec. 4) to construct a rigorous evaluation suite (dubbed PERSISTBENCH) for testing visual memory.

We construct PERSISTBENCH from a subset of the publicly available 360-1M [76] dataset. We first exploit a SLAM system [34] to compute initial camera poses, and then leverage GeoCalib [75] to align camera orientations with gravity. We discard any 360° clips where pose estimation fails or where the subsequent object tracking and B-spline trajectory optimization do not meet our smoothness and visibility constraints. From an initial pool of 24K raw 360° clips, this process yields 2,000 high-quality, paired evaluation sequences. To analyze model performance across different object

Models	Memory	Static Objects			Dynamic Objects		
		Permanence	Continuity	Appearance	Permanence	Continuity	Appearance
4D Recon.	CUT3R [79]	3.14% (66.43%)	70.66% (86.76%)	20.97% (56.71%)	0.99% (43.99%)	58.27% (76.94%)	18.04% (49.49%)
	4DGT [86]	16.03% (97.00%)	70.34% (93.03%)	37.36% (83.69%)	3.89% (96.21%)	61.12% (91.25%)	23.96% (79.98%)
	CogNVS [13]	41.98% (99.29%)	67.91% (96.10%)	64.08% (92.65%)	61.99% (99.84%)	61.53% (96.15%)	57.14% (92.75%)
	NeoVerse [88]	78.92% (97.23%)	63.66% (85.84%)	80.42% (91.77%)	54.31% (88.04%)	48.01% (71.33%)	65.54% (83.08%)
Cam. Ctrl.	ReCamMaster [4]	70.06% (95.48%)	51.73% (64.31%)	73.48% (89.36%)	45.03% (91.52%)	35.52% (54.65%)	50.07% (82.08%)
	TrajectoryCrafter [90]	81.56% (99.55%)	70.14% (96.96%)	77.52% (96.30%)	77.80% (99.74%)	56.45% (97.01%)	66.74% (94.24%)
	GEN3C [61]	83.97% (95.96%)	76.79% (86.98%)	82.61% (93.26%)	87.06% (96.13%)	64.62% (88.54%)	72.28% (90.94%)
	HyDRA [14]	70.38% (N/A)	38.90% (N/A)	65.65% (N/A)	55.49% (N/A)	35.41% (N/A)	44.82% (N/A)
360° Gen.	ViewPoint [20]	23.84% (77.29%)	46.14% (64.81%)	47.82% (74.28%)	16.63% (73.71%)	37.70% (59.66%)	34.61% (73.75%)
	Imagine360 [71]	34.17% (98.24%)	53.78% (93.93%)	40.78% (83.24%)	25.84% (98.14%)	44.86% (94.30%)	22.76% (80.09%)
	Argus [48]	84.06% (99.25%)	80.60% (95.86%)	68.49% (87.69%)	62.42% (98.97%)	59.86% (95.22%)	49.31% (85.20%)
	CubeComposer [43]	73.56% (98.94%)	75.51% (96.33%)	68.42% (93.18%)	45.29% (99.27%)	52.95% (95.33%)	37.86% (87.55%)

Table 1: **PERSISTBENCH results.** Experiments are conducted over multiple 4D reconstruction models (4D Recon.), camera-controllable video generation models (Cam. Ctrl.), and video-to-360° generation (360° Gen.) models. We report performance scores across static and dynamic subsets on both the invisible and visible (in parenthesis) segments. **Green** denotes the highest score and **yellow** denotes the second highest. Results for visible segments of HyDRA [14] are not applicable due to it only predicting frames after the object leaves view.

types, we use VLM [56] to categorize clips into 10 categories, including human, vehicle, animal, structure, and furniture. The clips are further divided into static and dynamic subsets based on whether the target object moves within the scene. These pairs represent a diverse range of complex motion and appearance scenarios, providing a statistically significant foundation for evaluating the limits of visual memory in foundation models.

Our automated evaluation pipeline uses off-the-shelf models: DINOv3 [65] localizes the target object in the first visible generated frame, SAM2 [60] tracks it through subsequent frames, an open-weight VLM Qwen3.8-27B [55] serves as the VLM judge to verify object existence, and DINOv2 [51] provides features for the App score.

6 Results

We evaluate 12 publicly available 4D foundation models spanning three distinct categories: (i) **4D reconstruction models** supporting dynamic novel-view synthesis (CUT3R [79], 4DGT [86], CogNVS [13], NeoVerse [88]); (ii) **camera-controlled video models** (ReCamMaster [4], TrajectoryCrafter [90], GEN3C [61], HyDRA [14]); and (iii) **video-to-360° generation models** (ViewPoint [20], Imagine360 [71], Argus [48], CubeComposer [43]), which we evaluate by cropping the generated view to ensure a fair perspective comparison.

We evaluate all models across both visible and invisible segments, and decouple objects of interest into static and dynamic subsets to isolate specific model behaviors. Tab. 1 and Fig. 4 summarizes our results. **The supplementary material provides model-inference details (Sec. F) and a human study assessing agreement with our metrics (Sec. D).** We also report category-wise results (Sec. E.2), temporal stability analysis (Sec. E.3), and an evaluation of 4DGT on longer videos (Sec. E.1). We describe our key findings below.

6.1 Key Findings

Seeing is not remembering. A common hope is that if a 4D foundation model can effectively reconstruct or generate dynamic scenes from alternative viewpoints, it must inherently understand the underlying dynamics well enough to extrapolate beyond the observed frames. However, as shown in Tab. 1 and Fig. 5, all models suffer from a substantial performance drop on invisible segments compared to visible ones. **This drop suggests that visual memory is a capability distinct from reconstruction or novel-view synthesis when the target object remains observed.** This also suggests that current models still rely primarily on interpolating directly from observed inputs, rather than constructing an internal memory of the world that allows them to persist through space and time.

Explicit conditioning helps visual memory. Another observation is that models which explicitly project input observations onto target views before inpainting (GEN3C [61], TrajectoryCrafter [90], and NeoVerse [88]) generally exhibit stronger visual memory. Notably, they significantly outperform others on invisible segments. We conjecture this is because most 4D models are predominantly

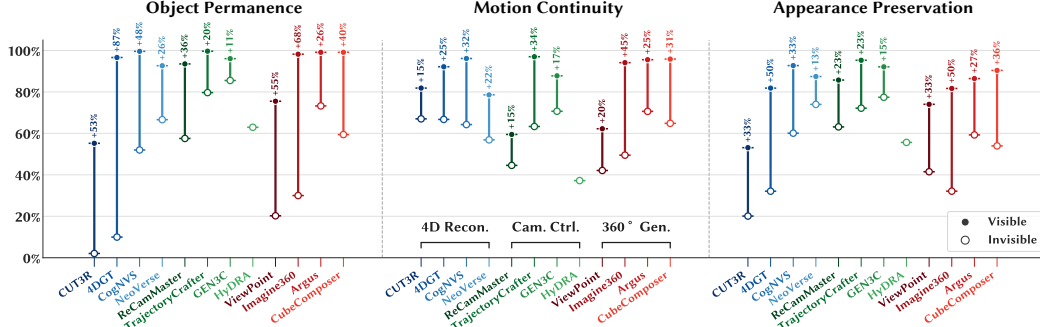


Figure 5: **Performance gap between visible and invisible segments on both the dynamic and static subsets.** We report, for each model and evaluation aspect, the drop in score from the visible to the invisible segment. Results for visible segments of HyDRA [14] are not applicable due to it only predicting frames after the object leaves view. All other models show a large gap across all aspects, suggesting that current models struggle to remember objects once they leave the input field of view.

trained on videos where target objects remain continuously visible [18, 49]. Consequently, when tasked with generating invisible segments, a scenario rarely encountered during training, they fall short. In contrast, explicit memory conditioning provides necessary guidance, reducing the complex generation process to a simpler refinement task.

Memory is multi-faceted. While the overall trends hold (i.e., models “remember” better on visible segments and static objects), each model excels at different aspects of visual memory. Using our complementary metrics, we are able to dissect their distinct strengths and weaknesses. For instance, as shown in Fig. 5, while GEN3C [61] and 4DGT [86] both excel at modeling motion continuity, GEN3C [61] is significantly better at maintaining object permanence. Similarly, while NeoVerse [88] and Imagine360 [71] track dynamics equally well, NeoVerse [88] is much better at preserving object identity.

Static memory is predictive of dynamic memory. While models generally remember static objects better than dynamic ones (Tab. 1), the results for both subsets are strongly correlated across all evaluated metrics (Fig. 6(d)). This suggests that models may rely on shared underlying mechanisms for both object types. Consequently, improvements in static memory are likely to translate to dynamic memory, shedding light on future direction for data curation and model refinement.

Properly curated 360° data enhances visual memory. Since 360° videos capture the entire environment, it is natural to assume that models trained to predict the full panorama will inherently learn to remember. **However, this is only partly true.** As shown in Tab. 1, 360° models, specifically Argus and CubeComposer, perform well on the static subset, while suffering from large performance drops on the dynamic subset. We conjecture this is because they are predominantly trained on scenes with few dynamic objects [20, 71] or simple camera trajectories [43], rendering complex dynamic evaluation scenarios out-of-distribution. Within the group, Argus [48], in contrast, is trained on more dynamic 360° scenes and explicitly models camera noise during training, and thus achieves the highest Perm and Cont scores among all models on the static subset, despite being built on a relatively old pretrain model [7]. This suggests that 360° data is a powerful resource, not just for evaluation, but for actively cultivating visual memory in foundation models.

7 Limitations

Our evaluation relies on off-the-shelf models for both data construction and scoring. Camera poses estimated by ViPE [34] may contain errors, while pretrained models used for object matching, tracking, and scoring may introduce additional inaccuracies. These errors can affect the measured memory scores independently of the evaluated model’s capabilities. We examine the sensitivity to evaluation-model choices in the supplementary material (Supp. Mat., Sec. C.5).

Our dataset is drawn from YouTube 360° videos [76]. Although it covers diverse object categories, its scene distribution reflects the available source videos. Extending the data collection to other sources and less represented environments could broaden the benchmark’s coverage.

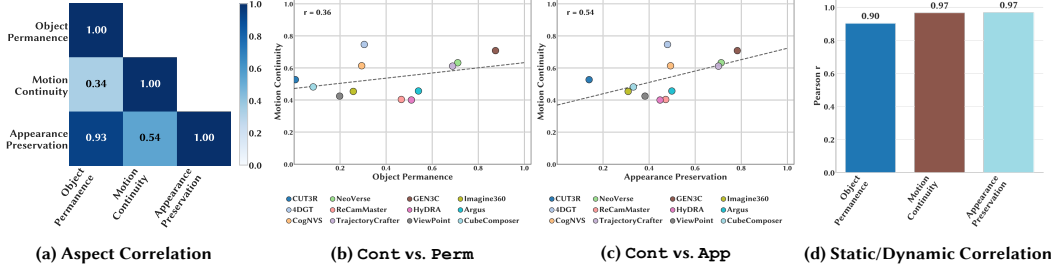


Figure 6: **Correlation analysis.** (a) The Pearson correlation matrix (computed on invisible segments across both static and dynamic subsets) shows that Perm and App are strongly correlated ($r = 0.93$), while Cont is weakly correlated with each ($r = 0.34$ and $r = 0.54$). The Cont-Perm (b) and Cont-App (c) scatter plots confirm that motion continuity is a largely independent capability, suggesting that visual memory is multi-faceted and should not be reduced to a single metric. (d) Static and dynamic subset scores (on invisible segments) correlate highly across all aspects, indicating that memory for both static and dynamic objects draws on shared underlying capabilities.

8 Conclusion

In this paper, we introduce PERSISTBENCH, a benchmark for evaluating object-centric visual memory in 4D foundation models. Our results reveal that while current models perform well at synthesizing novel views where most of the content is visible, they struggle to maintain object persistence once the object leaves the input field of view, indicating a fundamentally distinct capability. We further identify a key contributor to this gap: current 4D foundation models are predominantly trained on videos in which target objects remain visible throughout, leaving little supervision signal for the model to learn regarding *memory*. Architectures that condition on explicit geometric structure partially compensate for this by encoding object persistence directly into the input, but it remains unclear whether such architectural advantages would persist under a memory-aware training paradigm. Our analysis points to two complementary directions for future work: curating training data with frequent occlusion and reappearance, and further exploring architectures that combine explicit geometric conditioning with the generative priors of video models. We believe PERSISTBENCH can serve as a valuable testbed for driving progress toward 4D foundation models capable of true persistent memory, guiding future advances in both training and architecture design.

References

- [1] A new conceptualization of human visual sensory-memory. *Frontiers in Psychology*, 2016. doi: 10.3389/fpsyg.2016.00830. URL <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2016.00830>.
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.
- [3] Anthropic. Introducing Claude Sonnet 5, June 2026. URL <https://www.anthropic.com/news/claude-sonnet-5>.
- [4] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. Recammaster: Camera-controlled generative rendering from A single video. In *ICCV*, 2025.
- [5] Renée Baillargeon. Object permanence in 3½- and 4½-month-old infants. *Developmental Psychology*, 1987.
- [6] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.
- [7] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv*, 2023.
- [8] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyu Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 1995.
- [9] Xiao Cai, Sitong Su, Jingkuan Song, Pengpeng Zeng, Ji Zhang, Qinhong Du, Mengqi Li, Heng Tao Shen, and Lianli Gao. Gt23d-bench: A comprehensive general text-to-3d generation benchmark, 2025. URL <https://arxiv.org/abs/2412.09997>.
- [10] Susan Carey and Fei Xu. Infants’ knowledge of objects: Beyond object files and object tracking. *Cognition*, 2001.
- [11] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL <https://arxiv.org/abs/2511.16719>.
- [12] Arindam Chaudhuri. B-splines, 2021. URL <https://arxiv.org/abs/2108.06617>.
- [13] Kaihua Chen, Tarasha Khurana, and Deva Ramanan. Reconstruct, inpaint, test-time finetune: Dynamic novel-view synthesis from monocular videos, 2026. URL <https://arxiv.org/abs/2507.12646>.
- [14] Kaijin Chen, Dingkan Liang, Xin Zhou, Yikang Ding, Xiaoqiang Liu, Pengfei Wan, and Xiang Bai. Out of sight but not out of mind: Hybrid memory for dynamic video world models. *arXiv preprint arXiv:2603.25716*, 2026.
- [15] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025.
- [16] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/568_ECCV_2022_paper.php.

- [17] Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. *arXiv preprint arXiv:2504.00983*, 2025.
- [18] Zicheng Duan, Jiatong Xia, Zeyu Zhang, Wenbo Zhang, Gengze Zhou, Chenhui Gou, Yefei He, Feng Chen, Xinyu Zhang, and Lingqiao Liu. Liveworld: Simulating out-of-sight dynamics in generative video world models, 2026. URL <https://arxiv.org/abs/2603.07145>.
- [19] Shivam Duggal, Yushi Hu, Oscar Michel, Aniruddha Kembhavi, William T Freeman, Noah A Smith, Ranjay Krishna, Antonio Torralba, Ali Farhadi, and Wei-Chiu Ma. Eval3d: Interpretable and fine-grained evaluation for 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13326–13336, 2025.
- [20] Zixun Fang, Kai Zhu, Zhiheng Liu, Yu Liu, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Viewpoint: Panoramic video generation with pretrained diffusion models. In *NeurIPS*, 2025.
- [21] Aarash Feizi, Sai Rajeswar, Adriana Romero-Soriano, Reihaneh Rabbany, Valentina Zantedeschi, Spandana Gella, and João Monteiro. Pairbench: Are vision-language models reliable at comparing what they see?, 2025. URL <https://arxiv.org/abs/2502.15210>.
- [22] Stephanie Fu, Netanel Y. Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems*, 2023.
- [23] Xiao Fu, Shitao Tang, Min Shi, Xian Liu, Jinwei Gu, Ming-Yu Liu, Dahua Lin, and Chen-Hsuan Lin. Plenoptic video generation. *CoRR*, abs/2601.05239, 2026. doi: 10.48550/ARXIV.2601.05239. URL <https://doi.org/10.48550/arXiv.2601.05239>.
- [24] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *NeurIPS*, 2022.
- [25] Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *CoRR*, abs/2601.20540, 2026. doi: 10.48550/ARXIV.2601.20540. URL <https://doi.org/10.48550/arXiv.2601.20540>.
- [26] Dhruva Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Abstract-Datasets_and_Benchmarks.html.
- [27] Google DeepMind. Genie 3: A new frontier for world models. <https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/>, 2025. Accessed: 2025-03-05.
- [28] Google DeepMind. Gemini 3.5 Flash. Model card, May 2026. URL <https://deepmind.google/models/model-cards/gemini-3-5-flash/>.
- [29] Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. Cameractrl II: dynamic scene exploration via camera-controlled video diffusion models. *CoRR*, abs/2503.10592, 2025. doi: 10.48550/ARXIV.2503.10592. URL <https://doi.org/10.48550/arXiv.2503.10592>.
- [30] Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Cyrus Wu, Wei Li, Xuchen Song, Yang Liu, Eric Li, and Yahui Zhou. Matrix-game 2.0: An open-source, real-time, and streaming interactive world model. *CoRR*, abs/2508.13009, 2025. doi: 10.48550/ARXIV.2508.13009. URL <https://doi.org/10.48550/arXiv.2508.13009>.
- [31] Susan J Hespos and Kristy vanMarle. Physics for infants: characterizing the origins of knowledge about objects, substances, and number. *Wiley Interdiscip Rev Cogn Sci.*, 2012.

- [32] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.
- [33] Basile Van Hoorick, Dian Chen, Shun Iwase, Pavel Tokmakov, Muhammad Zubair Irshad, Igor Vasiljevic, Swati Gupta, Fangzhou Cheng, Sergey Zakharov, and Vitor Campagnolo Guizilini. Anyview: Synthesizing any novel view in dynamic scenes. *CoRR*, abs/2601.16982, 2026. doi: 10.48550/ARXIV.2601.16982. URL <https://doi.org/10.48550/arXiv.2601.16982>.
- [34] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixé, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. *CoRR*, 2025.
- [35] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/f8ad010cdd9143dbb0e9308c093aff24-Abstract-Datasets_and_Benchmarks.html.
- [36] Zhening Huang, Hyeonho Jeong, Xuelin Chen, Yulia Gryaditskaya, Tuanfeng Y. Wang, Joan Lasenby, and Chun-Hao Huang. Spacetimepilot: Generative rendering of dynamic scenes across space and time. *CoRR*, abs/2512.25075, 2025. doi: 10.48550/ARXIV.2512.25075. URL <https://doi.org/10.48550/arXiv.2512.25075>.
- [37] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.
- [38] Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *TPAMI*, 2025.
- [39] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *ICRA*, 2011.
- [40] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. URL <https://arxiv.org/abs/2308.04079>.
- [41] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM TOG*, 36(4), 2017.
- [42] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r, 2024. URL <https://arxiv.org/abs/2406.09756>.
- [43] Lingeng Li, Guangzhi Wang, Xiaoyu Li, Zhaoyang Zhang, Qi Dou, Jinwei Gu, Tianfan Xue, and Ying Shan. Cubecomposer: Spatio-temporal autoregressive 4k 360° video generation from perspective video. *arXiv preprint arXiv:2603.04291*, 2026.
- [44] Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard A. Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video. In *CVPR*, 2022.
- [45] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views, 2025. URL <https://arxiv.org/abs/2511.10647>.
- [46] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. *arXiv preprint arXiv:2404.01291*, 2024.

- [47] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.
- [48] Rundong Luo, Matthew Wallingford, Ali Farhadi, Noah Snaveley, and Wei-Chiu Ma. Beyond the frame: Generating 360° panoramic videos from perspective videos. In *ICCV*, 2025.
- [49] Ziqi Ma, Mengzhan Liufu, and Georgia Gkioxari. Out of sight, out of mind? evaluating state evolution in video world models, 2026. URL <https://arxiv.org/abs/2603.13215>.
- [50] NVIDIA, :, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezani, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai, 2026. URL <https://arxiv.org/abs/2511.00062>.
- [51] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khaidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido 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. *TMLR*, 2024.
- [52] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. *ACM TOG*, 2021.
- [53] Jean Piaget. *The Construction of Reality in the Child*. Basic Books, 1954.
- [54] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021.
- [55] Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL <https://qwen.ai/blog?id=qwen3.8>.
- [56] QwenTeam. Qwen3-vl: Sharper vision, deeper thought, broader action. <https://qwen.ai/blog?id=99f0335c4ad9ff6153e517418d48535ab6d8afef>, 2025. Accessed: 2026-02-28.
- [57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>.
- [58] Akshat Rakheja, Aarsh Ashdhir, Aryan Bhattacharjee, and Vanshika Sharma. World consistency score: A unified metric for video generation quality. *arXiv preprint arXiv:2508.00144*, 2025.
- [59] Nathan Ratliff, Matt Zucker, J. Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *ICRA*, 2009.
- [60] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloé Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross B. Girshick, Piotr Dollár, and

- Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.
- [61] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, 2025.
 - [62] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *ICCV*, pages 369–378, 2017. doi: 10.1109/ICCV.2017.48. URL <https://doi.org/10.1109/ICCV.2017.48>.
 - [63] John Schulman, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *RSS*, 2013.
 - [64] William Shakespeare. *Love’s Labour’s Lost*. The Arden Shakespeare, Third Series. Bloomsbury, London, 1998. Originally performed c. 1594–1595; first published 1598.
 - [65] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun 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, 2025. URL <https://arxiv.org/abs/2508.10104>.
 - [66] Richard Sorabji. *Aristotle on Memory*. University of Chicago Press, Chicago, 2nd edition, 2006. ISBN 9780226768236. First edition published 1972 by Duckworth, London.
 - [67] Elizabeth S. Spelke. Principles of object perception. *Cognitive Science*, 14(1):29–56, 1990. ISSN 0364-0213. doi: [https://doi.org/10.1016/0364-0213\(90\)90025-R](https://doi.org/10.1016/0364-0213(90)90025-R). URL <https://www.sciencedirect.com/science/article/pii/036402139090025R>.
 - [68] Elizabeth S. Spelke. *What Babies Know: Core Knowledge and Composition Volume 1*. Oxford University Press, 2022.
 - [69] Elizabeth S Spelke and Katherine D Kinzler. Core knowledge. *Developmental Science*, 2007.
 - [70] Elizabeth S Spelke, Karen Breinlinger, Janet Macomber, and Kristen Jacobson. Origins of knowledge. *Psychological Review*, 1992.
 - [71] Jing Tan, Shuai Yang, Tong Wu, Jingwen He, Yuwei Guo, Ziwei Liu, and Dahua Lin. Imagine360: Immersive 360 video generation from perspective anchor. In *NeurIPS*, 2025.
 - [72] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv*, 2024.
 - [73] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 1980.
 - [74] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Al-abdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. URL <https://arxiv.org/abs/2502.14786>.
 - [75] Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. Geocalib: Learning single-image calibration with geometric optimization. In *ECCV*, 2024.
 - [76] Matthew Wallingford, Anand Bhattad, Aditya Kusupati, Vivek Ramanujan, Matt Deitke, Aniruddha Kembhavi, Roozbeh Mottaghi, Wei-Chiu Ma, and Ali Farhadi. From an image to a scene: Learning to imagine the world from a million 360° videos. In *NeurIPS*, 2024.

- [77] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wenten Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.
- [78] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. VGGT: visual geometry grounded transformer. In *CVPR*, pages 5294–5306. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00499. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.
- [79] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025.
- [80] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, pages 20697–20709. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01956. URL <https://doi.org/10.1109/CVPR52733.2024.01956>.
- [81] Yiming Wang, Qihang Zhang, Shengqu Cai, Tong Wu, Jan Ackermann, Zhengfei Kuang, Yang Zheng, Frano Rajic, Siyu Tang, and Gordon Wetzstein. Bulletime: Decoupled control of time and camera pose for video generation. *CoRR*, abs/2512.05076, 2025. doi: 10.48550/ARXIV.2512.05076. URL <https://doi.org/10.48550/arXiv.2512.05076>.
- [82] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004.
- [83] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas J. Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v(ision) is a human-aligned evaluator for text-to-3d generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 22227–22238. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02098. URL <https://doi.org/10.1109/CVPR52733.2024.02098>.
- [84] 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°. *CoRR*, abs/2601.16192, 2026. doi: 10.48550/ARXIV.2601.16192. URL <https://doi.org/10.48550/arXiv.2601.16192>.
- [85] Fei Xu and Susan Carey. Infants’ metaphysics: The case of numerical identity. *Cognitive Psychology*, 30(2):111–153, 1996. ISSN 0010-0285. doi: <https://doi.org/10.1006/cogp.1996.0005>. URL <https://www.sciencedirect.com/science/article/pii/S0010028596900055>.
- [86] Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4dgt: Learning a 4d gaussian transformer using real-world monocular videos. In *NeurIPS*, 2025.
- [87] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *CVPR*, 2025.
- [88] Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. *arXiv preprint arXiv:2601.00393*, 2026.
- [89] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

- [90] Mark YU, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *ICCV*, 2025.
- [91] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.
- [92] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. FLARE: feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *CVPR*, pages 21936–21947. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02043. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhang-FLARE_Feed-forward_Geometry_Appearance_and_Camera_Estimation_from_Uncalibrated_Sparse_CVPR_2025_paper.html.
- [93] Xinlu Zhang, Yujie Lu, Weizhi Wang, An Yan, Jun Yan, Lianke Qin, Heng Wang, Xifeng Yan, William Yang Wang, and Linda Ruth Petzold. Gpt-4v(ision) as a generalist evaluator for vision-language tasks, 2023. URL <https://arxiv.org/abs/2311.01361>.
- [94] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv*, 2025.

Can 4D Foundation Models Remember?

Supplementary Material

Contents

A	Additional Details on Dataset Curation	20
A.1	Data Collection and Preprocessing	20
A.2	Object Discovery and Tracking	20
A.3	Constrained Camera Trajectory Generation	21
A.4	Post-Processing and Quality Filtering	22
A.5	Segment Duration Statistics	22
B	Metric Implementation Details	22
C	Metric Design Ablations	24
C.1	Appearance Score Design	24
C.2	Object Permanence Score Design	25
C.3	Motion Continuity Score Design	26
C.4	Tracking-Conditioned Aggregation	26
C.5	Off-the-Shelf Model Ablations	27
C.6	Camera Perturbation Analysis	28
C.7	Rotation-Dominant Subset Analysis	28
C.8	Correlation with Existing Benchmarks	28
C.9	Reference Score	29
D	Human Study	30
D.1	Study Protocol	30
D.2	Participant Instructions	30
D.3	Agreement with Automatic Metrics	31
E	Additional Experiments	32
E.1	Evaluation on Longer Videos	32
E.2	Category-Wise Results	32
E.3	Temporal Stability Analysis	33
E.4	Motion Continuity and Motion Ambiguity	33
F	Additional Details on Models	34
F.1	Per-Model Inference Details	34

A Additional Details on Dataset Curation

We provide additional details on each stage of the PERSISTBENCH dataset construction pipeline: data collection and preprocessing, object discovery and tracking, and constrained camera trajectory generation.

A.1 Data Collection and Preprocessing

Following [48], we collect 360° videos from YouTube. The downloaded videos undergo a filtering step to remove static clips, screen recordings, and low-quality footage before being segmented into 10-second clips.

Camera pose estimation. For each clip, we crop its center region into a perspective video and run ViPE [34], an optimization-based SLAM method, to obtain pseudo-ground-truth camera poses in metric scale. Following [84], we stabilize the estimated poses to maintain a consistent orientation within each clip. To enforce gravity alignment, we uniformly sample 10 frames per clip, crop 8 perspective views per frame, and run GeoCalib [75] to estimate the average gravity direction, which we then use to re-align all camera poses.

A.2 Object Discovery and Tracking

Object identification. We run Qwen3-VL-Plus [56] on 5 uniformly sampled frames from each 360° clip, prompting it to identify objects that match our construction requirements (static or dynamic). The prompt explicitly distinguishes *dynamic* from *potentially movable* objects (e.g., a parked car is static), and we provide in-context examples to reduce ambiguity. The model also outputs a category-level object description and, for dynamic objects, the motion type. The full prompts used for dynamic and static object identification are shown in Fig. 7 and Fig. 8, respectively.

Role: You are an expert in identifying the dominant object in a video.

Task: Given a sequence of subsampled frames from a 10-second clip of a 360° video, identify the dominant *dynamic* object in the video. If there are multiple dominant objects, identify a random dynamic object with visible motion.

Instructions:

1. The images are subsampled frames from a 10-second 360° video clip.
2. Look carefully at the images as well as how objects move throughout all the images.
3. Identify the dominant object or the object with the most amount of visible motion.
4. Separate object motion from camera motion. If the object remains static while the camera is moving, the object is **not** a dynamic object.
5. Separate whether the object is *movable* or *dynamic*. A car is movable but not dynamic if it is parked, and should **not** be identified as a dynamic object.
6. Classify the object into a short category description, e.g., “car”, “person”, “animal”, “bike”.
7. Output your result strictly in the format below. If no dominant object is found, output `null` for all fields.

Output: A JSON object with fields: "object_name" (string), "object_motion" (specific motion such as “driving” or “walking”; "static" if static; null if not found), "object_confidence" (float).

Figure 7: VLM prompt for dynamic object identification.

360° tracking with SAM3. Given the object description, we prompt SAM3 [11] to segment and track all matching instances throughout the 360° clip. Directly applying SAM3 to equirectangular video fails when objects cross the left or right boundary: the object re-enters from the opposite edge and is often assigned a new identity, producing discontinuous tracks.

To address this, we horizontally concatenate a second copy of the same 360° clip to the right, so that edge-crossing trajectories remain continuous within the duplicated canvas. This introduces duplicate tracks, which we remove using two rules:

Role: You are an expert in identifying the static object in a video. Task: Given a sequence of subsampled frames from a 10-second clip of a 360° video, identify a <i>static</i> object in the video. If there are multiple static objects, identify a random one and describe its general category. Instructions: 1. The images are subsampled frames from a 10-second 360° video clip. 2. If an object is certain not to move in real life, it is a static object. 3. If an object is movable but not dynamic in this clip, it is still considered static (e.g., a parked car). 4. The object should not be too large (e.g., “ground”, “sky”) nor too small to be visible. 5. Classify the object into a short category description, e.g., “building”, “pillar”. 6. Only name objects with clear, unambiguous edges. Do not describe objects like “mountain”, “snow”, or “water”. 7. Output your result strictly in the format below. If no dominant object is found, output <code>null</code> for all fields. Output: A JSON object with fields: "object_name" (string), "object_motion" ("static", or null if not found), "object_confidence" (float).
--

Figure 8: VLM prompt for static object identification.

1. If a mask never touches the center seam or either boundary, a duplicate always exists in the copied half; we remove the instance in the right half.
2. If a mask touches the center seam or either boundary, a duplicate exists that touches one of the boundaries; we keep the seam duplicate.

Valid masks are then merged back to the original 360° frame coordinates. We discard any track that begins after the first frame or is dropped mid-clip, yielding stable persistent segmentations for downstream trajectory generation.

A.3 Constrained Camera Trajectory Generation

Object selection. For each clip, we rank tracked objects by their average mask area over time and select the top 10. For each selected object we generate one paired perspective sequence (input and reference) via camera trajectory optimization.

Visibility interval precomputation. A perspective camera with fixed FOV can be parameterized by yaw-pitch angles (θ_t, ϕ_t) at frame t . We precompute per-frame angle intervals for both axes: $\Theta_t^{\text{full}}/\Phi_t^{\text{full}}$ are the ranges that make the object *fully* visible, and $\Theta_t^{\text{part}}/\Phi_t^{\text{part}}$ are the ranges ensuring *at least partial* visibility. By definition, $\Theta_t^{\text{full}} \subseteq \Theta_t^{\text{part}}$ and $\Phi_t^{\text{full}} \subseteq \Phi_t^{\text{part}}$. These intervals serve as frame-wise constraints during trajectory optimization.

Visibility schedule. The input camera follows an *invisible-visible-invisible* pattern: the object is unseen at the start and end, and briefly enters the frame during a middle window. This middle visible window is what provides the model its only direct observation of the object; evaluating the invisible segments tests whether memory persists beyond that window. We sample four key timesteps: (1) end of the first invisible segment, (2) start of the visible segment, (3) end of the visible segment, and (4) start of the final invisible segment. The gaps between (1)–(2) and (3)–(4) allow smooth transitions between visibility states. Concretely, we sample the visible segment length $T_{\text{visible}} \in [0.1T, 0.6T]$, its start time $t_{\text{visible-start}} \in [0.2T, 0.8T - T_{\text{visible}}]$, and the first invisible segment length $T_{\text{first-invisible}} \in [0.1 t_{\text{visible-start}}, 0.6 t_{\text{visible-start}}]$. The reference camera keeps the object visible at all times.

Keyframe sampling. At each key timestep t , we sample (θ_t, ϕ_t) conditioned on the desired visibility state. For visible keyframes, we draw from a clipped Gaussian $\mathcal{N}(\mu, \sigma)$ inside the corresponding interval, with μ at the interval midpoint and σ equal to one-quarter of the interval width. For invisible keyframes, we sample uniformly outside the visibility interval. All keyframe samples are constrained so the per-frame angular change does not exceed 1° (i.e., 30°/s at 30 fps) and the to-

tal rotation within any window stays within 20° . For invisible windows, we apply only θ (yaw) constraints in practice, as natural camera motion is predominantly horizontal.

Trajectory optimization. Given sampled keyframes, we solve for dense trajectories $\{(\theta_t, \phi_t)\}_{t=1}^T$ via constrained optimization. We initialize piecewise trajectories between consecutive keyframes using B-splines [12] and optimize control points with L-BFGS-B [8]. Let $x_t \in \{\theta_t, \phi_t\}$ denote the optimization variable at frame t . The full objective is:

$$\begin{aligned} \mathcal{L} = & \lambda_{\text{vis}}(\mathcal{L}_{\text{inc}} + \mathcal{L}_{\text{exc}}) + \lambda_{\text{smooth}}(\mathcal{L}_{\text{vel}} + \mathcal{L}_{\text{acc}} + \mathcal{L}_{\text{jerk}}) \\ & + \lambda_{\text{cont}}(\mathcal{L}_{\text{pos-cont}} + \mathcal{L}_{\text{vel-cont}} + \mathcal{L}_{\text{acc-cont}}) + \lambda_{\text{center}}\mathcal{L}_{\text{center}}, \end{aligned} \quad (6)$$

where:

- $\mathcal{L}_{\text{inc}}$ penalizes (via L1 distance to interval limit) any visible frame where (θ_t, ϕ_t) falls outside $\Theta_t^{\text{full}} \times \Phi_t^{\text{full}}$.
- $\mathcal{L}_{\text{exc}}$ penalizes any invisible frame where (θ_t, ϕ_t) falls inside $\Theta_t^{\text{part}} \times \Phi_t^{\text{part}}$.
- $\mathcal{L}_{\text{vel}}, \mathcal{L}_{\text{acc}}, \mathcal{L}_{\text{jerk}}$ are finite-difference smoothness penalties on velocity, acceleration, and jerk of $\{x_t\}$.
- $\mathcal{L}_{\text{pos-cont}}, \mathcal{L}_{\text{vel-cont}}, \mathcal{L}_{\text{acc-cont}}$ enforce position, velocity, and acceleration continuity at spline junctions.
- $\mathcal{L}_{\text{center}}$ is a weak centering regularizer that pulls visible frames toward the midpoint of the visibility interval, improving optimization stability.

Trajectories that violate any visibility constraint after optimization are rejected.

A.4 Post-Processing and Quality Filtering

Trajectory post-processing. We further smooth the optimized camera trajectories before rendering the final video pairs. Concretely, we apply a low-pass filter to the per-frame camera angle sequences in the frequency domain, removing high-frequency oscillations while preserving the overall motion envelope. This is done for both the reference and input views after camera trajectory generation, which reduces residual jitter without changing the downstream mask-based validity checks.

Mask refinement. Because masks projected from the equirectangular 360° segmentation can be low-resolution, we refine object masks on the generated video pairs using SAM2 [60]. Specifically, we run SAM2 on the reference video (where the object is always visible) using the projected 360° mask on the first frame as a prompt, and then project the refined reference masks onto the input video. We discard any sequence in which the refined SAM2 mask and the original projected SAM3 mask do not overlap on any frame.

Scale filtering. We also filter sequences with extreme object scale based on the minimum and maximum mask area over time: a sequence is dropped if the largest-frame mask exceeds 70% of image pixels or if the smallest-frame mask falls below 2% of image pixels.

A.5 Segment Duration Statistics

Fig. 9 summarizes the durations of visible and invisible segments for the static and dynamic subsets. The median visible duration is 2.62 s in both subsets, while the median invisible duration is 2.50 s for dynamic objects and 2.62 s for static objects.

B Metric Implementation Details

All metrics are computed after resizing the target-view videos and masks to a common evaluation resolution of 384×512 pixels. We first identify the first frame in which the target object is visible according to the trusted visibility annotation. Frames before this point are excluded from the final averages, since they do not provide a reliable target-object observation from which to initialize tracking.

Because model outputs do not include object masks, we infer predicted masks before applying the metrics defined in Sec. 3.2. At the first visible frame, we crop the reference object using its

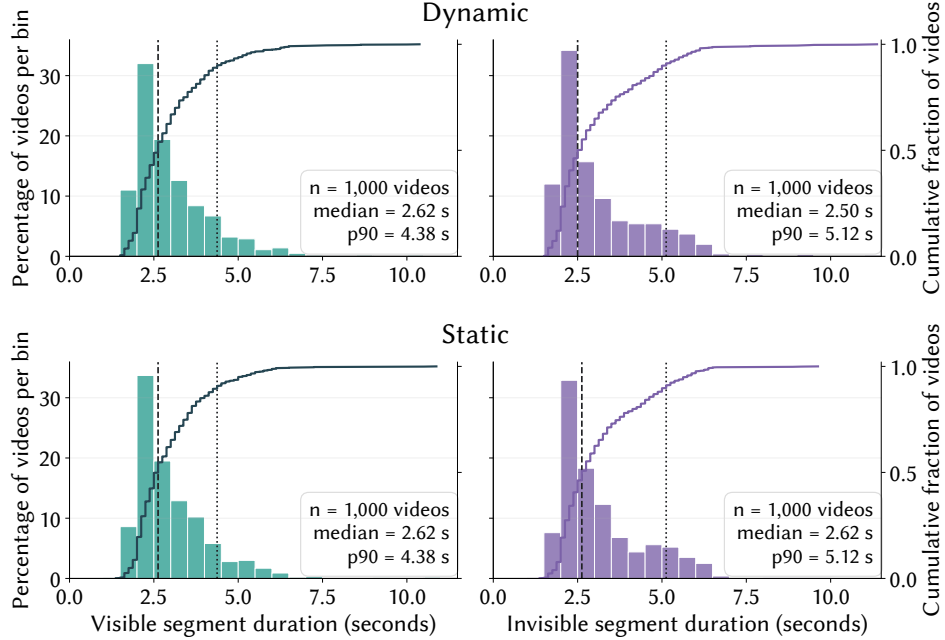


Figure 9: **Visible and invisible segment durations for both subsets.** Distributions are shown for 1,000 dynamic object videos (top) and 1,000 static object videos (bottom), with visible segments on the left and invisible segments on the right. Bars show the percentage of videos per bin, and curves show the cumulative fraction of videos. Dashed and dotted vertical lines mark the median and 90th percentile, respectively. Durations are measured in seconds.

reference mask and search for the most similar region in the corresponding generated frame using dense DINO feature matching. The reference mask support is then geometrically transferred into the matched predicted bounding box to form an initial predicted mask. Starting from this initialization, SAM2 propagates the mask through the remaining generated frames. A frame is treated as trackable if SAM2 returns a non-empty mask; otherwise, the predicted mask is empty and the SAM2 indicator for that frame is set to 0.

For the VLM component of Perm, the identity reference for each queried frame is produced by cropping the same-time-step reference frame using the reference mask bounding box with 10% padding and resizing the result to 192×256 . The generated frame is cropped using its SAM2-predicted mask bounding box with 10% padding, resized to the same size, and tagged with a frame identifier. The prompt asks the VLM to judge whether the same semantic target object appears, without requiring matching appearance and while ignoring spatial alignment. Changes in pose, scale, lighting, texture, color, shape details, partial viewpoint, and moderate cropping are allowed as long as the object remains recognizable. The full prompt is shown in Fig. 10.

Recursive VLM labeling. The VLM judge is applied adaptively, with independently queried frame pairs, rather than querying every generated frame. The full procedure is shown in Alg. 11. Each pairwise VLM call is retried up to three times and accepted only when the parsed JSON provides a Boolean visibility label for the queried frame tag.

For Cont, invalid masks are handled before aggregation: frames with an empty reference mask, an empty predicted mask, or a failed SAM2 track are marked as invalid and skipped. The reference area used for scale normalization is the average mask area over the evaluated window after the tracking start, excluding invalid reference masks. The active implementation uses only image-plane center distances; camera-aware reprojection and velocity-distribution diagnostics exist in the evaluation code but are not used for the benchmark scores.

For App, DINO features are extracted from the full reference and generated frames after resizing each frame so that its height and width are multiples of the DINO patch size. The reference and predicted masks are downsampled to the DINO patch grid using area interpolation, and positive mask locations select the object tokens used for masked average pooling. If a mask selects no patch

Role: You are judging whether the target object in a prediction crop is the same object category/instance as the target object in a reference crop.

Input description: The first image is the reference object crop tagged as REF. It comes from the GT frame at the same time step as the prediction frame and is cropped using the target object’s GT mask bounding box with 10% padding. The second image is a prediction crop tagged with its unique frame ID, produced using the SAM2-predicted mask bounding box with 10% padding.

Internal reasoning: Use chain-of-thought internally to identify and describe the target object in REF, compare the tagged prediction crop against the reference, and decide whether the object is visible in that crop without requiring matching appearance.

Judging rules:

1. It is OK if the object changes pose, scale, lighting, texture, color, shape details, generated appearance, partial viewpoint, or is moderately cropped.
2. Count it as visible as long as it can reasonably be recognized as the same semantic target object.
3. If the prediction crop is empty, shows only background, shows a different object, or is too ambiguous, mark it not visible.
4. Use appearance only as a weak clue, and ignore whether the crop is spatially aligned with the reference frame.
5. Evaluate only the target object from REF.

Output: Return exactly:

```

<thinking>
short reasoning
</thinking>
<answer>
{
  "reference_description": "<brief description of the object in
  REF>",
  "tag": "{frame_tag}",
  "visible": true,
  "reason": "<short reason>"
}
</answer>

```

Output rules: The JSON inside <answer> must be valid JSON. The frame tag must exactly match the queried frame tag, and visible must be a JSON boolean.

Figure 10: VLM prompt for object permanence judging.

tokens, all patch tokens are used as a fallback for that frame. Auxiliary CLIP, DreamSim, DINO CLS-token, crop-based DINO, and max-pooling diagnostics are computed for analysis, but they are not included in the reported benchmark table.

C Metric Design Ablations

C.1 Appearance Score Design

We conduct an evaluation study to compare candidate appearance metrics and validate the final global DINO average-pooling design used for App. We consider crop-based CLIP similarity, crop-based DreamSim similarity [22], DINO CLS-token similarity from object crops, DINO average patch-token similarity from crop features, and DINO max-pooled patch-token similarity from crop features. We also compare two global variants: global DINO average pooling, which first runs DINO on the full image and then average-pools patch tokens inside the object mask, and global DINO max pooling, which uses the same full-image features but aggregates the masked patch tokens with max pooling.

To quantify which metric best separates high-quality and low-quality object appearances, we sample 2,000 prediction-reference pairs from a model output and compute their object IoU, equivalently the $\mathcal{J}$ -measure. We then extract the top 100 and bottom 100 pairs according to IoU. Since IoU close to

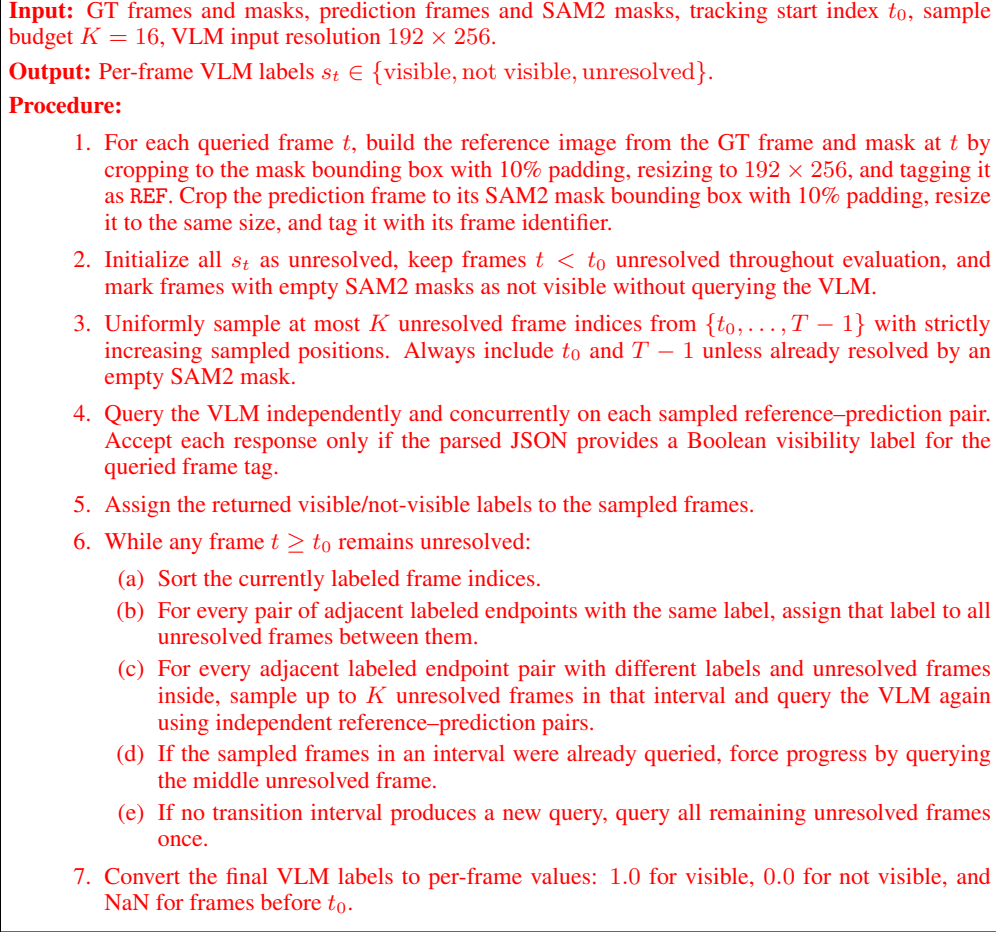


Figure 11: Adaptive VLM labeling algorithm for the object permanence judge.

1 indicates strong contour agreement with the reference object, we treat the top-IoU set as a pseudo high-quality appearance set and the bottom-IoU set as a pseudo low-quality set. For each candidate appearance metric, we compute the Earth Mover’s Distance (EMD) between its score distributions on these two sets. A larger EMD indicates that the metric better separates clearly good object renderings from clearly poor ones.

As shown in Fig. 12, global DINO average pooling consistently gives the largest distribution gap across evaluated models, outperforming crop-based CLIP, DreamSim, crop-level DINO variants, and global DINO max pooling. This supports our final design choice: aggregating DINO patch features from the full image within the object mask provides a more robust appearance signal than relying only on cropped inputs or max-pooled patch responses.

C.2 Object Permanence Score Design

We also ablate the dual-verification design of Perm. As described in Sec. 3.2, object permanence should require more than a plausible image region: the target object must remain spatially coherent enough to track and semantically recognizable as the same object. We therefore combine a SAM2 tracking signal with a VLM identity signal.

Fig. 13 illustrates why both components are needed. In the first case, the target is a tree and SAM2 correctly indicates that the original tracked object has been lost. The VLM alone, however, can confuse other nearby trees with the reference tree and incorrectly mark the object as present. This shows why tracking is necessary: it detects discontinuities in the object instance that a semantic judge may overlook. In the second and third cases, SAM2 continues to track a heavily degraded

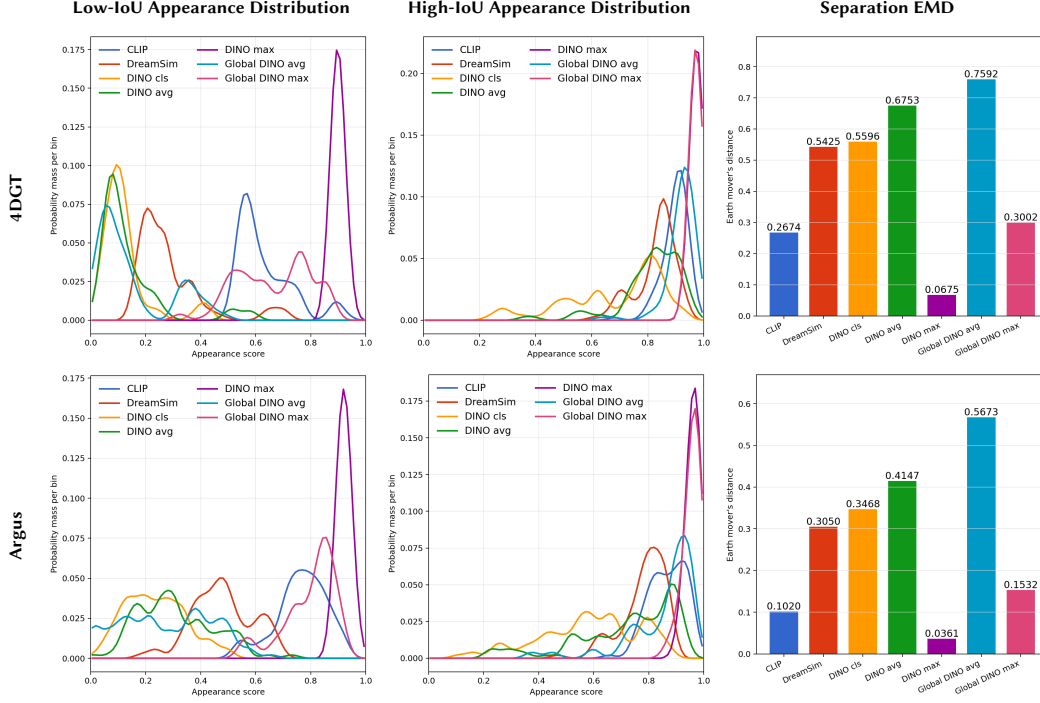


Figure 12: **Appearance metric ablation.** We rank prediction-reference pairs by object IoU and compare each metric’s score distributions on the top and bottom 100 pairs using EMD. CLIP assigns high scores to both sets, DreamSim shows weak separation with degraded ordering, DINO CLS under-scores high-quality pairs, crop DINO average pooling separates reasonably but less strongly, and DINO max/global DINO max over-score both sets due to max aggregation. Global DINO average pooling gives the largest EMD and clearest separation, motivating its use in App.

object-like blob. The tracked region remains spatially coherent, but the object identity is no longer recognizable. The VLM correctly rejects these cases, showing that semantic identity checking is necessary when tracking alone follows a corrupted region.

These examples motivate the logical-AND design in Perm: SAM2 verifies instance-level continuity, while the VLM verifies object identity and recognizability. Their combination better matches the goal of measuring whether the same observed object persists in the generated sequence.

C.3 Motion Continuity Score Design

To examine whether our results depend on the centroid-based definition of Cont, we recompute motion continuity using three alternatives: bounding-box center L2 distance, mask IoU, and dynamic time warping (DTW) over the centroid trajectory. We reuse the existing tracked masks across all 12 evaluated methods on the same subset of 200 videos (100 dynamic and 100 static), changing only the motion continuity calculation. As shown in Tab. 3, Spearman’s rank correlations with default Cont range from 0.979 to 1.000 on visible segments and from 0.944 to 0.993 on invisible segments. The method rankings are therefore largely preserved under all three alternatives, suggesting that our comparisons do not depend strongly on the specific choice of centroid distance.

C.4 Tracking-Conditioned Aggregation

We compare the Spearman correlations of Cont and App with Perm under all-frame and tracked-frame aggregation in Tab. 2, using the 12 methods over the full dataset. All-frame aggregation assigns zero to untracked frames, approximately doubling the correlation between Cont and Perm. This makes motion continuity more dependent on object permanence, contrary to our intended sep-

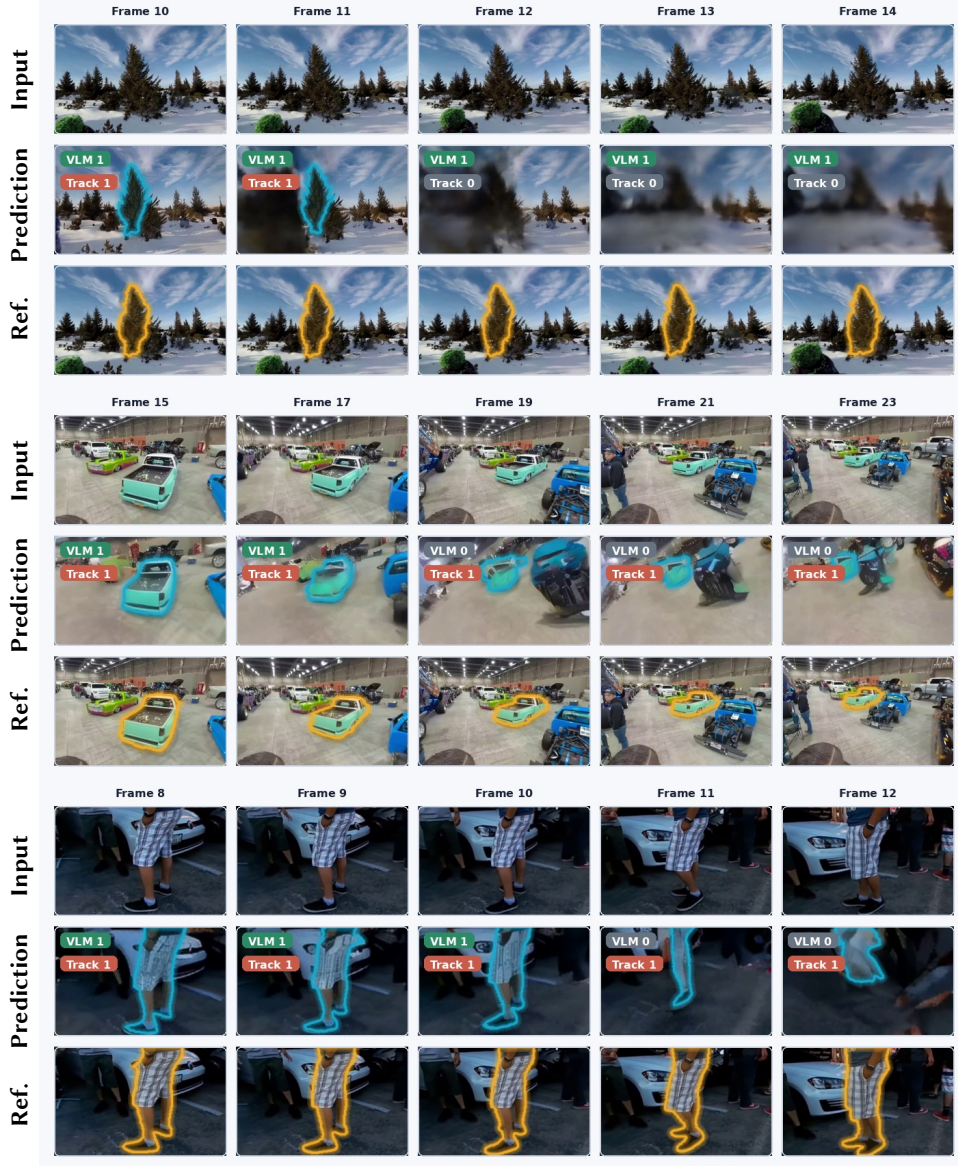


Figure 13: **Object permanence score ablation.** SAM2 tracking and VLM identity checking capture complementary failure modes. In case 1, SAM2 correctly detects that the tracked object is lost, while the VLM mistakes other trees for the target tree. In cases 2 and 3, SAM2 continues tracking a severely degraded object-like blob, but the VLM rejects it as no longer recognizable as the target object. Requiring both signals therefore better matches our object-centric definition of permanence.

eration of these aspects. Our default tracked-frame aggregation preserves this distinction more effectively, while App remains strongly correlated with Perm under both settings.

C.5 Off-the-Shelf Model Ablations

We conduct these ablations across all 12 evaluated methods on a subset of 200 videos, comprising 100 dynamic and 100 static cases. To examine how the choice of off-the-shelf models affects our evaluation, we replace the initial matching model, tracking model, VLM judge, and appearance feature extractor separately. Fig. 14 shows the resulting rankings across the 12 evaluated methods for both visible and invisible segments. For initial matching, we replace DINOv3-L/16 [65] with DINOv2-B/14 [51], an earlier and smaller model from the same family, and CLIP-L/14 [57] and

Aggregation	Cont-Perm	App-Perm
All-frame	0.685	0.972
Tracked-frame (default)	0.343	0.965

Table 2: **Tracking-conditioned aggregation ablation.** Spearman’s rank correlations between all-frame and tracked-frame aggregation across 12 methods and all 2,000 cases. All-frame aggregation assigns untracked frames with zero scores, whereas tracked-frame aggregation averages only successfully tracked frames.

SigLIP2-L/16 [74], which use image-text training rather than DINO’s image-only self-supervised training. For tracking, we replace SAM2.1 Large [60] with SAM2.1 Tiny to test a smaller model within the same generation, SAM3.1 [11] to test a newer generation of the SAM family, and XMem [16], an earlier video object segmentation model with a different memory-based architecture. For the VLM judge, we replace Qwen3.8-27B [55] with Gemini 3.5 Flash [28] and Claude Sonnet 5 [3], testing whether object-presence judgments remain consistent across models from different providers. For appearance scoring, we replace DINOv2-L/14 [51] with DINOv2-B/14 to test model size, DINOv3-L/16 to test model generation, and CLIP-L/14 and SigLIP2-L/16 to test different feature families. We evaluate initial matching and appearance scoring separately to distinguish their effects on the final scores. We also report Spearman’s rank correlations between the default and variant configurations in Tab. 3, showing that method rankings are largely preserved across component replacements.

C.6 Camera Perturbation Analysis

We conduct this analysis across all 12 evaluated methods on a subset of 200 videos, comprising 100 dynamic and 100 static cases. Since our evaluation uses rendered outputs, errors in camera control can affect the measured memory scores. To show the effect of camera control capability on the metrics, we apply random shift sequences to the generated videos and recompute the three metrics while keeping the reference videos fixed. We compare the default setting with low and high perturbation levels corresponding to 3% and 9% of the image width. As shown in Fig. 15, Perm and App rankings are largely preserved, while Cont shows slightly more ranking changes across all methods, consistent with its use of object position. This shows that Perm and App are largely robust to model’s camera control capability, while Cont can be slightly affected.

C.7 Rotation-Dominant Subset Analysis

Video-to-360° methods produce target views using camera rotation alone, whereas other methods also account for target-camera translation. To examine whether this difference affects cross-method comparisons, we re-aggregate existing predictions on a rotation-dominant subset of 200 videos and compare the resulting rankings with the default rankings across all 12 methods. As shown in Tab. 3, the invisible-segment Spearman correlations are 0.895, 0.916, and 0.930 for Perm, Cont, and App, respectively. The corresponding visible-segment correlations are 0.825, 0.944, and 0.965. These results show that method rankings remain broadly consistent on the rotation-dominant subset, although some changes remain. This supports the robustness of the overall comparison when translation effects are reduced, without fully isolating camera handling from other differences between methods.

C.8 Correlation with Existing Benchmarks

To examine whether our metrics reflect general video quality, we also evaluate the generated outputs using WorldScore [17], which measures video consistency, quality, and dynamics. These metrics evaluate the generated video without comparing the target object with its reference observation. We compute Spearman’s rank correlations between the six WorldScore metrics and our default Perm, Cont, and App scores across the 12 evaluated methods on a subset of 200 videos (100 dynamic and 100 static). As shown in Tab. 3, more than half of the correlations with WorldScore are zero or negative, indicating substantial differences between its method rankings and ours. These results sug-

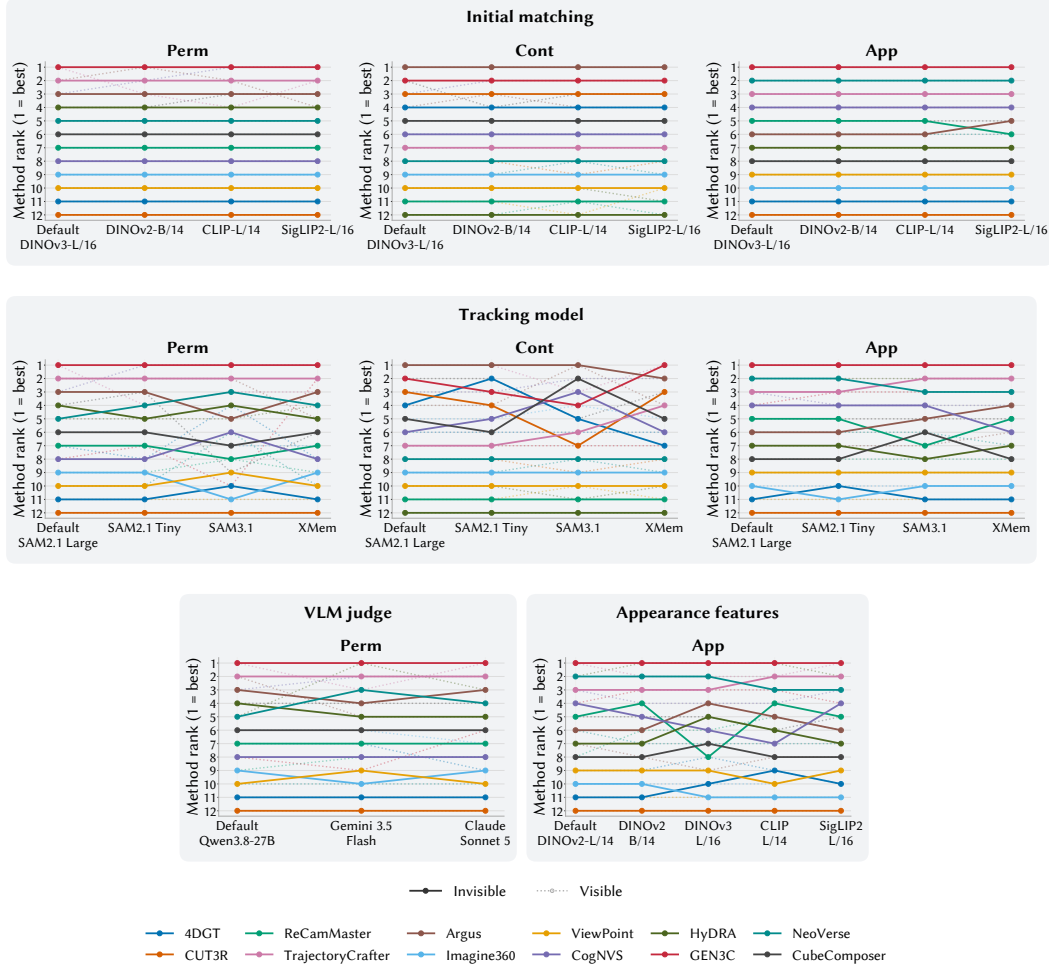


Figure 14: **Effect of off-the-shelf model choices on method rankings.** We replace the initial matching model (top), tracking model (middle), VLM judge (bottom left), and appearance feature extractor (bottom right), and compare the rankings of the 12 evaluated methods with the default configuration on a subset of 200 videos (100 dynamic and 100 static). The default and replacement models are listed on each horizontal axis. Solid and dotted lines show invisible and visible segment rankings, respectively. Rank 1 denotes best. While changing the model slightly affects the method rankings, they are mainly preserved across largely different off-the-shelf models spanning diverse categories and performance tiers.

gest that our reference-based, object-centric evaluation provides information beyond general video quality and consistency, rather than reproducing the rankings from existing metrics.

C.9 Reference Score

Tab. 4 reports the score obtained by evaluating the reference videos with the same metric pipeline used for all model outputs. Values outside parentheses are measured on invisible frames, and values inside parentheses are measured on visible frames.

The reference scores are close to, but not exactly, 100% because the reference video is treated as a baseline prediction and passed through the same mask-inference pipeline as every model output. In particular, SAM2 is run on the reference video to produce tracked masks, and repeated SAM2 tracking introduces slight mask noise. This mask noise affects centroid-based Cont and masked-feature App, while Perm remains perfect. Since all reference scores remain very close to 100%, this

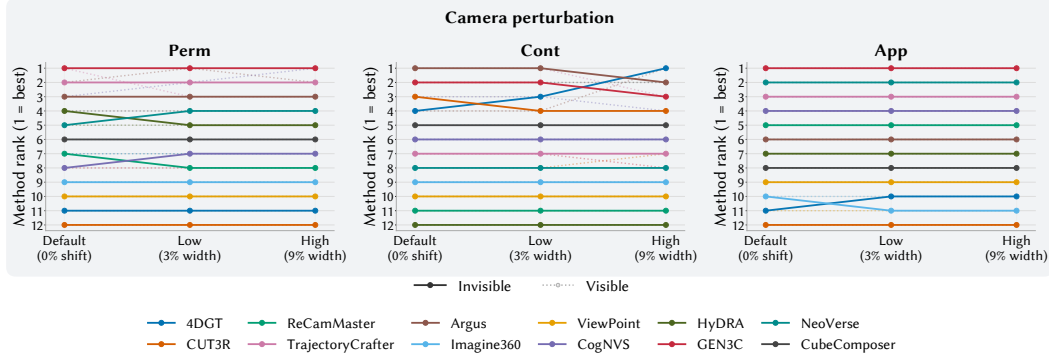


Figure 15: **Effect of camera perturbation on method rankings.** We compare the rankings of the 12 evaluated methods on a subset of 200 videos (100 dynamic and 100 static) under the default setting and low and high frame perturbation levels corresponding to 3% and 9% of the image width. Solid and dotted lines show invisible and visible segment rankings, respectively. Rank 1 denotes the best. Rankings are largely stable for Perm and App, and Cont.

small deviation is acceptable and reflects measurement noise rather than a failure of the reference video.

D Human Study

We conduct a human study to compare our automatic memory metrics with human judgments of object permanence, motion continuity, and appearance preservation across the 12 evaluated methods.

D.1 Study Protocol

In each trial, participants compare two videos generated by different methods for the same input, alongside the reference video and a copy with the target object highlighted in cyan. The generated videos are labeled A and B, with method names and automatic scores hidden. Controls allow participants to pause, replay, move through the videos, and adjust playback speed. Participants are asked to compare each memory property by selecting *A better*, *Tie*, *B better*, or *Not sure*.

The study covers all 66 pairs of methods, with 24 distinct cases per pair covering both dynamic and static subset, as well as visible and invisible segments. We then randomly shuffle and divide these into 66 questionnaires, each containing 24 main trials, two introductory examples, one gold-standard control trial, and one repeated main trial with the positions of A and B swapped.

We exclude a submitted questionnaire if the participant selects the generated video over the reference for any metric in the gold-standard trial, or selects different videos for the same metric in the original and repeated trials. After collecting and filtering, 864 main comparisons remain.

D.2 Participant Instructions

Fig. 16 shows the participant instructions and an example case. Participants are asked to focus on the cyan-highlighted target object and ignore the background, other objects, people, camera quality, and unrelated differences between the videos. Each trial uses the following three prompts:

1. **Object permanence:** Which video contains the target object for longer? Judge only whether the object continues to exist. Ignore how it looks or moves.
2. **Motion continuity:** Does the object follow similar motion to the target object in the reference video? If the target object is static, its position shouldn't change (up to camera movement). Appearance doesn't matter.
3. **Appearance preservation:** Which one of the generated objects looks more similar to the target object in the reference video? Motion doesn't matter.

Group	Variant	Perm	Cont	App
Initial matching	DINOv2-B/14	1.000 (0.979)	1.000 (0.979)	1.000 (1.000)
	CLIP-L/14	1.000 (0.951)	1.000 (0.979)	1.000 (1.000)
	SigLIP2-L/16	1.000 (0.965)	1.000 (0.986)	0.993 (1.000)
Tracking model	SAM2.1 Tiny	0.993 (0.944)	0.972 (1.000)	0.993 (0.993)
	SAM3.1	0.930 (0.776)	0.860 (0.902)	0.958 (0.993)
	XMem	0.993 (0.790)	0.930 (0.993)	0.965 (0.986)
VLM judge	Gemini 3.5 Flash	0.972 (0.888)	—	—
	Claude Sonnet 5	0.993 (0.916)	—	—
Appearance features	DINOv2-B/14	—	—	0.993 (0.965)
	DINOv3-L/16	—	—	0.916 (0.951)
	CLIP-L/14	—	—	0.930 (0.944)
	SigLIP2-L/16	—	—	0.986 (0.972)
Motion continuity definition	BBox center L2	—	0.993 (0.993)	—
	Mask IoU	—	0.944 (1.000)	—
	DTW	—	0.986 (0.979)	—
Camera perturbation	Low (3% width)	0.986 (0.979)	0.993 (1.000)	0.993 (1.000)
	High (9% width)	0.986 (0.972)	0.958 (0.944)	0.993 (1.000)
Evaluation subset	Rotation dominant	0.895 (0.825)	0.916 (0.944)	0.930 (0.965)
WorldScore [17]	3D Consistency	0.490 (0.175)	-0.231 (-0.161)	0.524 (0.559)
	Photometric Consistency	-0.063 (-0.308)	0.000 (-0.448)	-0.049 (0.021)
	Style Consistency	-0.105 (0.105)	0.524 (0.371)	-0.294 (-0.280)
	Subjective Quality	0.497 (0.179)	-0.382 (-0.095)	0.550 (0.634)
	Motion Magnitude	0.175 (0.503)	-0.133 (0.427)	-0.021 (0.147)
	Motion Smoothness	-0.245 (-0.517)	-0.063 (-0.559)	-0.049 (-0.119)

Table 3: **Ranking correlations under off-the-shelf model replacements, motion continuity definitions, camera perturbations, rotation-dominant subset selection, and with WorldScore [17].** We report Spearman’s ρ between the default and variant rankings across the 12 evaluated methods for each metric. All experiments use 200 sampled videos. Off-the-shelf variants replace the initial matching model, tracking model, VLM judge, and appearance feature extractor separately. The motion continuity variants reuse the existing masks and compare bounding-box center L2 distance, mask IoU, and dynamic time warping (DTW) over centroid trajectories with default Cont. Camera perturbations apply low and high shifts of 3% and 9% of the image width. The rotation-dominant row compares default rankings with those obtained on the rotation-dominant subset. The WorldScore rows report correlation between each WorldScore metric and our default metric rankings. Values outside parentheses are measured on invisible segments, and values inside parentheses are measured on visible segments. Dashes indicate metrics not affected by the corresponding ablation.

	Static Objects			Dynamic Objects		
	Perm	Cont	App	Perm	Cont	App
Reference	99.99% (99.99%)	98.68% (98.88%)	99.79% (99.82%)	99.62% (99.96%)	97.84% (98.76%)	99.69% (99.86%)

Table 4: **Reference scores on the full benchmark.** The reference video is evaluated by the same pipeline as model predictions.

D.3 Agreement with Automatic Metrics

For each metric, we compare human judgments with our default PERSISTBENCH scores for the same cases. We include a comparison only when the human response is A, B, or a tie and both methods have valid automatic scores.

We assign one point for a win, half a point for a tie, and zero for a loss, for both human judgments and automatic scores. For automatic comparisons, the method with the higher score wins. Each method’s win ratio is its total points divided by its number of valid comparisons. We combine comparisons across all visible/invisible segments and dynamic/static objects. We then compute Spearman’s rank correlation between these win ratios across the 12 methods, weighting each method equally.

As shown in Fig. 17, Spearman’s rank correlations are 0.944 for Perm, 0.566 for Cont, and 0.986 for App, based on 826, 676, and 675 valid comparisons, respectively. These results show strong

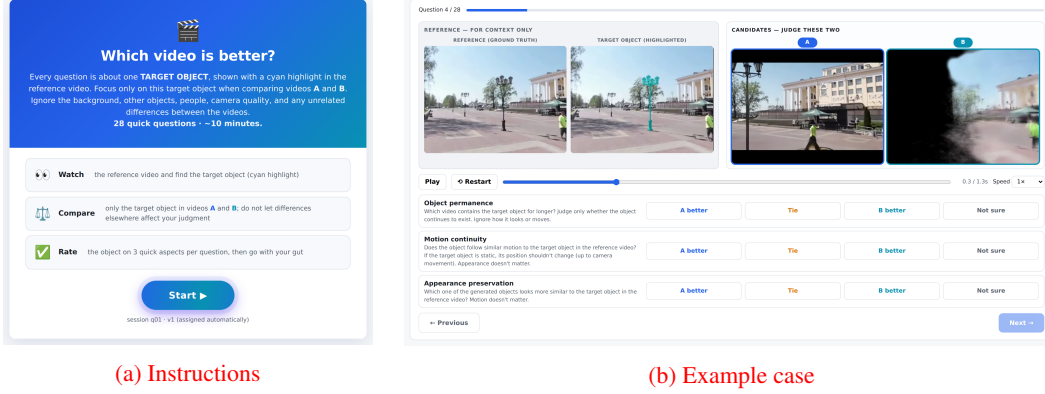


Figure 16: **Human-study interface.** Participant instructions (left) and an example case with a street lamp as the target object (right). Each trial shows the reference video, a copy with the target object highlighted in cyan, and generated videos A and B, along with playback controls and the three metric questions. No participant responses are shown.

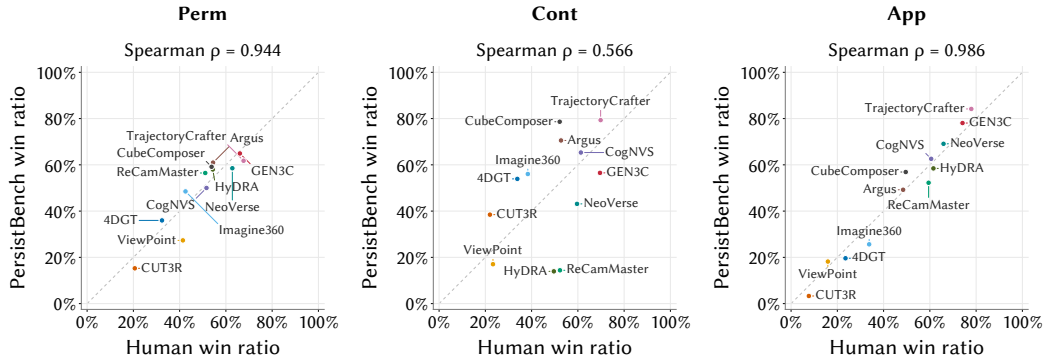


Figure 17: **Alignment with human judgments.** We compare human win ratios ranking (horizontal axis) with PERSISTBENCH win ratios (vertical axis) across the 12 evaluated methods. Each point represents one method. Win ratios assign one point to a win, half a point to a tie, and zero to a loss, using the same valid comparisons across visible/invisible segments and dynamic/static objects. Spearman’s correlations are 0.944 for Perm, 0.566 for Cont, and 0.986 for App.

agreement with human rankings for object permanence and appearance preservation, and moderate agreement for motion continuity. The correlations measure agreement in method rankings, rather than the accuracy of absolute scores.

E Additional Experiments

E.1 Evaluation on Longer Videos

To show that our data construction and evaluation pipeline can be easily scaled beyond the current clip lengths, we evaluate 4DGT [86] on 10 additional dynamic object sequences of approximately 60 seconds each, constructed from longer 360° videos. We report Perm, Cont, and App separately for visible and invisible segments together with our default dynamic subset results.

E.2 Category-Wise Results

Fig. 18 reports the three memory scores across object categories for all 12 evaluated methods, separately for dynamic and static objects. Performance varies across categories, and a method’s strength in one memory property does not always extend to the others. For example, GEN3C achieves high App scores across both dynamic and static categories, while its Cont ranking varies across cate-

Duration	Perm	Cont	App
~10 s (default)	3.89% (96.21%)	61.12% (91.25%)	23.96% (79.98%)
~60 s	2.69% (42.07%)	45.32% (67.30%)	61.44% (48.13%)

Table 5: **4DGT evaluation on longer videos.** Both rows report dynamic-object results, with invisible scores followed by visible scores in parentheses. The longer-video evaluation uses 10 cases with 128 prediction frames per case. Following Tab. 1, case scores are equally weighted for each metric. The higher invisible-segment App score on longer videos is averaged over tracked frames from only 3 of the 10 cases with valid scores, so it should not be interpreted as improved appearance preservation across the full set.

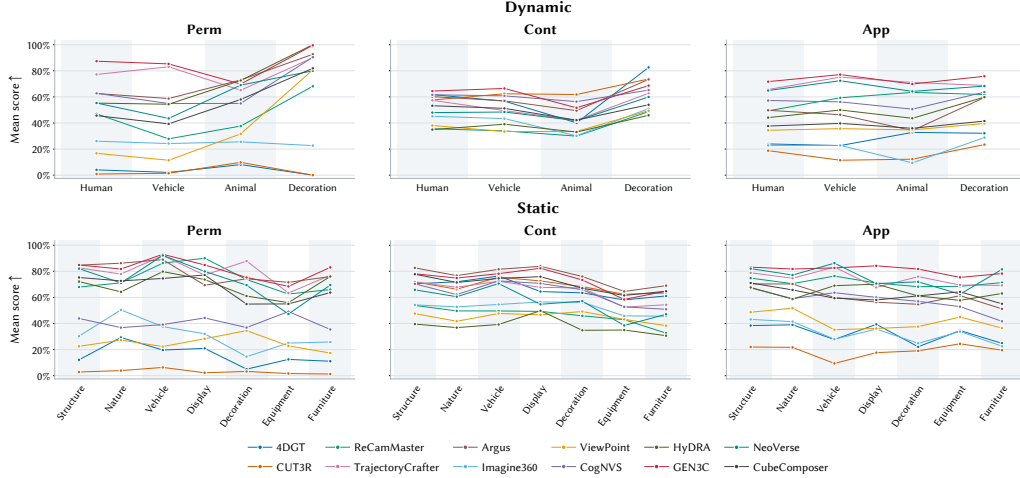


Figure 18: **Category-wise visual memory results.** We report mean Perm, Cont, and App scores (left to right) across the 12 evaluated methods, separately for dynamic objects (top) and static objects (bottom). Each curve represents one method, and higher scores indicate better performance. The results show differences in performance across object categories and memory properties.

gories. Within the dynamic subset, several methods obtain lower Cont scores for animals than for decorations. The static subset also shows category-dependent differences in scores and method rankings. These results provide a more detailed view of model performance than the overall averages.

E.3 Temporal Stability Analysis

To examine how visual memory scores change over time, we report mean Perm, Cont, and App scores against normalized progress through both visible and invisible segments in Fig. 19. Across the 12 evaluated methods, scores generally decrease slightly over time without sudden changes. While some methods change ranking, the overall performance trends remain similar across time. This supports our use of average scores to summarize each method’s performance, while the curves provide additional detail on how performance changes within each segment.

E.4 Motion Continuity and Motion Ambiguity

As a dynamic object remains out of view longer, its future motion becomes less certain, and multiple trajectories may be plausible. Our Cont metric compares the predicted object position with the reference and does not account for these alternative trajectories. As shown in the invisible-segment results in Fig. 19, Cont remains relatively stable for several methods and decreases gradually for others. These results support the use of reference-based Cont in our setting, although motion ambiguity may become more significant over longer invisible intervals.

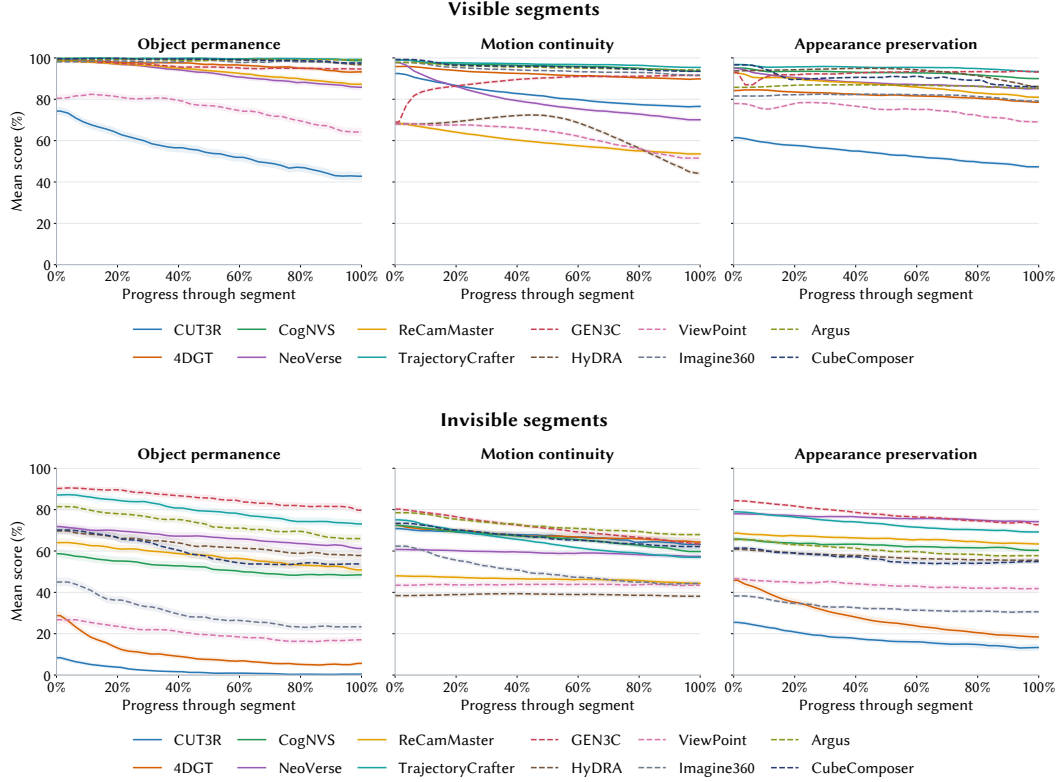


Figure 19: **Temporal analysis of visual memory scores.** Mean Perm, Cont, and App scores (left to right) are plotted against normalized progress through visible segments (top) and invisible segments (bottom) across 12 evaluated methods. Scores generally decrease over time without sudden changes. While some methods change ranking, the overall performance trends remain similar across time, supporting our use of average scores to summarize each method’s performance.

F Additional Details on Models

Baseline inference protocol. For each method, the input is the view_0 perspective video and its camera trajectory, and the query is the target camera trajectory from the paired center view. We run each baseline once to produce RGB frames for the target view, then evaluate the saved predictions with PERSISTBENCH. Unless otherwise noted, the benchmark visibility annotations are not provided as model inputs; they are used only for metric computation and for selecting the tracking start frame. The only exception is HyDRA, whose wrapper uses the visibility mask to split the sequence into a visible conditioning prefix and a post-visible target segment.

Frame staging. Different baselines impose different input sizes and temporal lengths. Before inference, we therefore stage each sequence according to the model interface: for fixed-length models, frames and camera poses are uniformly sampled with rounded linspace indices; for fixed-resolution models, images are center-cropped to the target aspect ratio and resized, while the pinhole intrinsics are updated accordingly. Models declared with arbitrary length and resolution receive the original frames and camera files directly. All camera files store OpenCV camera-to-world poses and a pinhole intrinsic matrix.

F.1 Per-Model Inference Details

Argus. Argus [48] is run through its ground-truth-pose inference path. The model first synthesizes a 360°/equirectangular video from the staged perspective input using Stable Video Diffusion image-to-video weights, with source ground-truth poses used to build its motion cache. The final target predictions are obtained by cropping perspective views from the generated equirectangular video

Model	Frames	Resolution
Argus [48]	25	384×512
CogNVS [13]	49	480×720
CubeComposer [43]	27	–
CUT3R [79]	–	–
4DGT [86]	128	504×672
GEN3C [61]	121	704×1280
HyDRA [14]	77	480×832
Imagine360 [71]	32	384×512
NeoVerse [88]	81	336×560
ReCamMaster [4]	81	480×832
TrajectoryCrafter [90]	49	576×1024
ViewPoint [20]	49	384×512

Table 6: **Fixed inference length and resolution for the 12 baselines.** A dash indicates that our evaluator does not impose a fixed value, although the external model may still resize internally.

according to target rotations and field of view, so target translation is not represented in the final crop. Argus ignores the benchmark visibility masks during inference.

CogNVS. CogNVS [13] estimates depth on the input frames with Depth Anything 3, forward-warps each source frame into the corresponding target camera, and then runs CogVideoX/CogNVS inpainting diffusion over the warped video and masks. It uses an empty prompt, 50 denoising steps, guidance scale 6.0, bf16 inference, a DPM scheduler, model CPU offload, VAE slicing/tiling, and the implementation’s fixed random seed of 42. The known cameras are used for the target-view warping stage, while visibility masks are ignored.

CubeComposer. CubeComposer [43] projects the perspective input video into equirectangular and cubemap conditioning, completes the panoramic video over temporal windows, and crops target perspective views from the generated panorama. Its cached 3k configuration uses 27 frames, window length 9, cubemap size 768, all six cube faces, diagonal-context conditioning, two history windows, future context, and a global sink token. The method is rotation-centric in this setup: camera translations and visibility masks are ignored during inference.

CUT3R. CUT3R [79] receives the original staged sequence without evaluator-imposed sampling or resizing, although the model internally resizes/crops images for inference. It processes the input sequence recurrently to build a 3D memory, then probes that memory with target ray maps. In our relative setting, target translations are scaled into CUT3R’s coordinate system using the ratio between CUT3R-predicted and ground-truth first-to-last input-camera displacement. CUT3R uses its predicted poses, focal lengths, intrinsics, and 3D points, and ignores visibility masks.

4DGT. 4DGT [86] converts the benchmark OpenCV camera-to-world poses to the model’s OpenGL convention, encodes input frames, cameras, and timestamps into a dynamic Gaussian/video representation, and sequentially renders RGB frames for the requested target cameras. The method is geometry/rendering-centric rather than diffusion-inpainting-centric, and it ignores benchmark visibility masks during inference.

GEN3C. GEN3C [61] uses the staged input frames, cameras, and repeated intrinsics to construct a depth- and camera-conditioned 4D cache. If explicit depth is unavailable, it estimates depth with Depth Anything 3. It then renders target-view warp guidance and synthesizes the final video with the Gen3C-Cosmos-7B video2world pipeline. We use an empty prompt and enable offloading for the diffusion transformer, tokenizer, text encoder, prompt upsampler, and guardrail models. Visibility masks are not used during inference.

HyDRA. HyDRA [14] is the only baseline wrapper that uses the benchmark visibility mask during inference. It finds the last visible frame, treats the visible prefix as conditioning, samples both conditioning and target segments to the 77-frame model interface, and restores the full timeline after generation. Target camera-to-world poses are converted into HyDRA camera embeddings,

with every fourth camera used by the model. The diffusion run uses Wan2.1 components, the HyDRA checkpoint, 50 denoising steps, classifier-free guidance scale 5.0, and generic positive/negative prompts.

Imagine360. Imagine360 [71] converts the perspective input into panorama conditioning, runs the Imagine360/AnimateDiff pipeline with the fixed prompt “A realistic 360 video.”, and saves generated equirectangular frames. Target predictions are cropped from these ERP frames using relative rotations, with GeoCalib pitch estimates used when enabled by the configuration. As with other panorama-crop baselines, target translation is only weakly represented, and visibility masks are ignored.

NeoVerse. NeoVerse [88] first runs a DA3-based reconstructor to obtain Gaussian splats, rendered intrinsics/extrinsics, and timestamps. It aligns the benchmark cameras to the reconstructor coordinate frame using the first input frame, renders target RGB/depth/mask guidance with the Gaussian renderer, and then completes the result with WanVideoNeoVerse diffusion. The method uses target cameras explicitly through the rendering stage, but ignores visibility masks.

ReCamMaster. ReCamMaster [4] runs a camera-conditioned Wan video diffusion pipeline with the ReCamMaster checkpoint. It captions sampled input frames with its VLM captioner, falling back to a generic prompt if captioning fails, and conditions generation on target camera embeddings relative to the first input camera. The embeddings are downsampled every four frames, and inference uses 50 denoising steps, classifier-free guidance scale 5.0, and seed 0. Visibility masks are ignored.

TrajectoryCrafter. TrajectoryCrafter [90] generates a BLIP2 caption when no prompt is supplied, estimates depth with DepthCrafter, and aligns those depths to Depth Anything 3 metric depth using robust scale/shift fitting. It converts source and target OpenCV camera-to-world poses to world-to-camera poses, forward-warps source frames to target views with the metric-aligned depth, and inpaints the warped video and masks with CogVideoX. It uses the first ten input frames as reference and ignores visibility masks.

ViewPoint. ViewPoint [20] generates a cubemap/equirectangular video from the input with the fixed prompt “realistic video, natural motion”, converts the generated packed view into a panorama, and crops target perspective views using relative rotations and target intrinsics. It assumes a shared camera center for the crop, so target translations are ignored in this setup. Visibility masks are ignored.

Camera handling. The baselines differ substantially in how they use the target camera. Argus, CubeComposer, Imagine360, and ViewPoint produce an intermediate panoramic or equirectangular video and obtain the final prediction by cropping according to target rotations and intrinsics; target translations are therefore ignored or only weakly represented. CogNVS, GEN3C, NeoVerse, and TrajectoryCrafter explicitly warp or render target-camera guidance before diffusion completion. 4DGT directly renders target cameras from its reconstructed dynamic representation. CUT3R stores a recurrent 3D memory from the input and queries it with target ray maps; in our relative setting, target translations are normalized by the ratio between CUT3R-predicted and ground-truth input-camera displacement. ReCamMaster uses target poses through downsampled relative camera embeddings, taking every fourth target pose.

Evaluation-time auxiliary models. The PERSISTBENCH metric is applied after baseline inference and does not re-run the baseline model. For each predicted video, we resize/copy the ground-truth frames, ground-truth masks, and predictions to the metric resolution of 384×512 . We use SAM2 [60] to track the predicted object from the first visible reference frame, DINOv3 [65] features to localize, DINOv2 [51] to compare object appearance, and the VLM judge described in Sec. 3.2 to verify object presence. Thus, SAM2, DINOv3, DINOv2, and the VLM judge are evaluation auxiliaries rather than components of the baseline inference pipelines.

G Additional Visualization

We provide additional qualitative results to complement the examples in Fig. 4. These visualizations show more static-subset cases across representative baseline predictions, illustrating how the object-centric metrics reflect object permanence, motion continuity, and appearance preservation beyond the main-paper examples. We also include an interactive visualizer in the supplementary material ([index.html](#)), which allows readers to browse additional benchmark examples and model predictions.



Figure 20: **Additional qualitative comparisons on the dynamic subset.**



Figure 21: **Additional qualitative comparisons on the dynamic subset.**

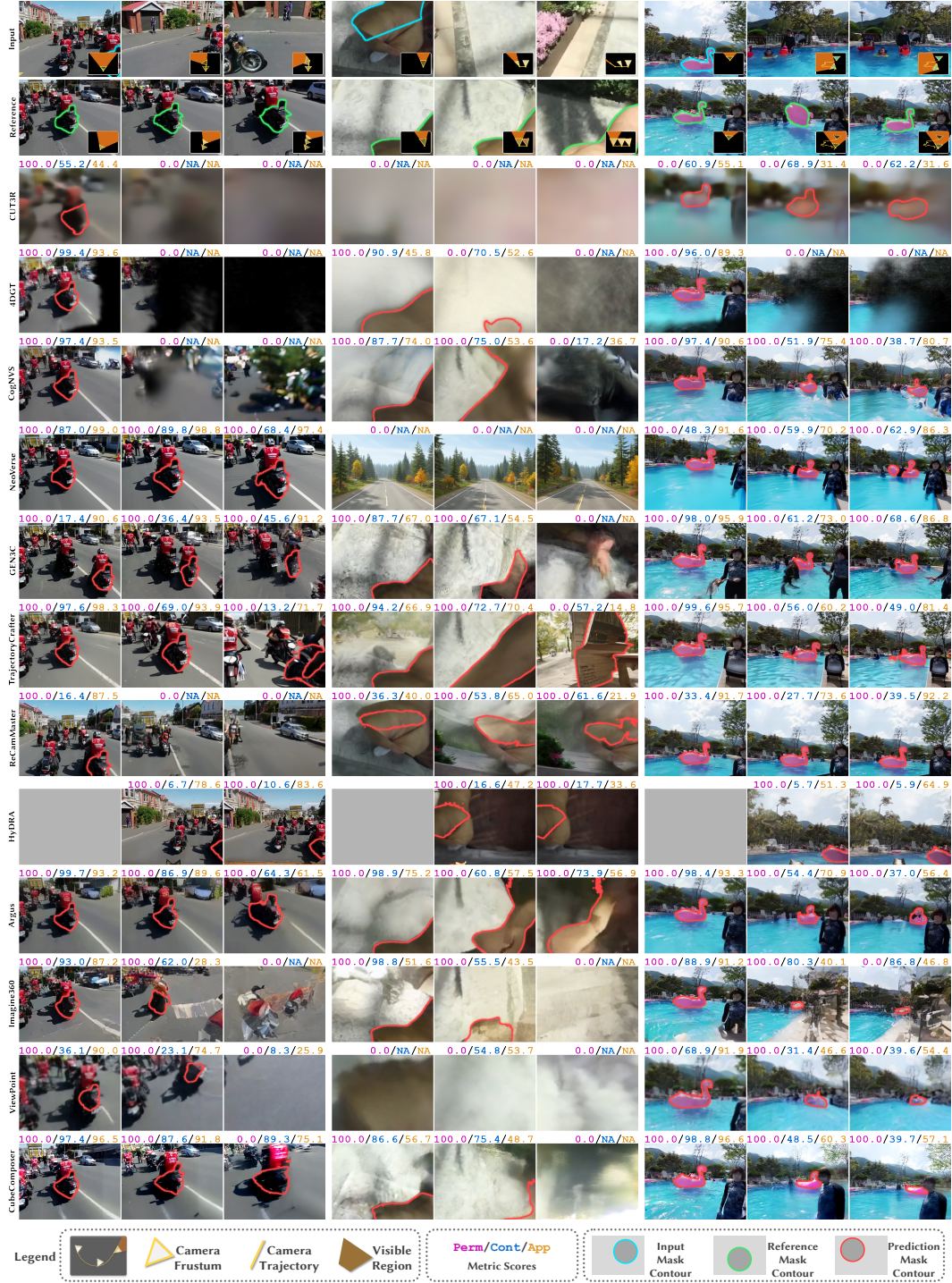


Figure 22: **Additional qualitative comparisons on the dynamic subset.**



Figure 23: **Additional qualitative comparisons on the static subset.**

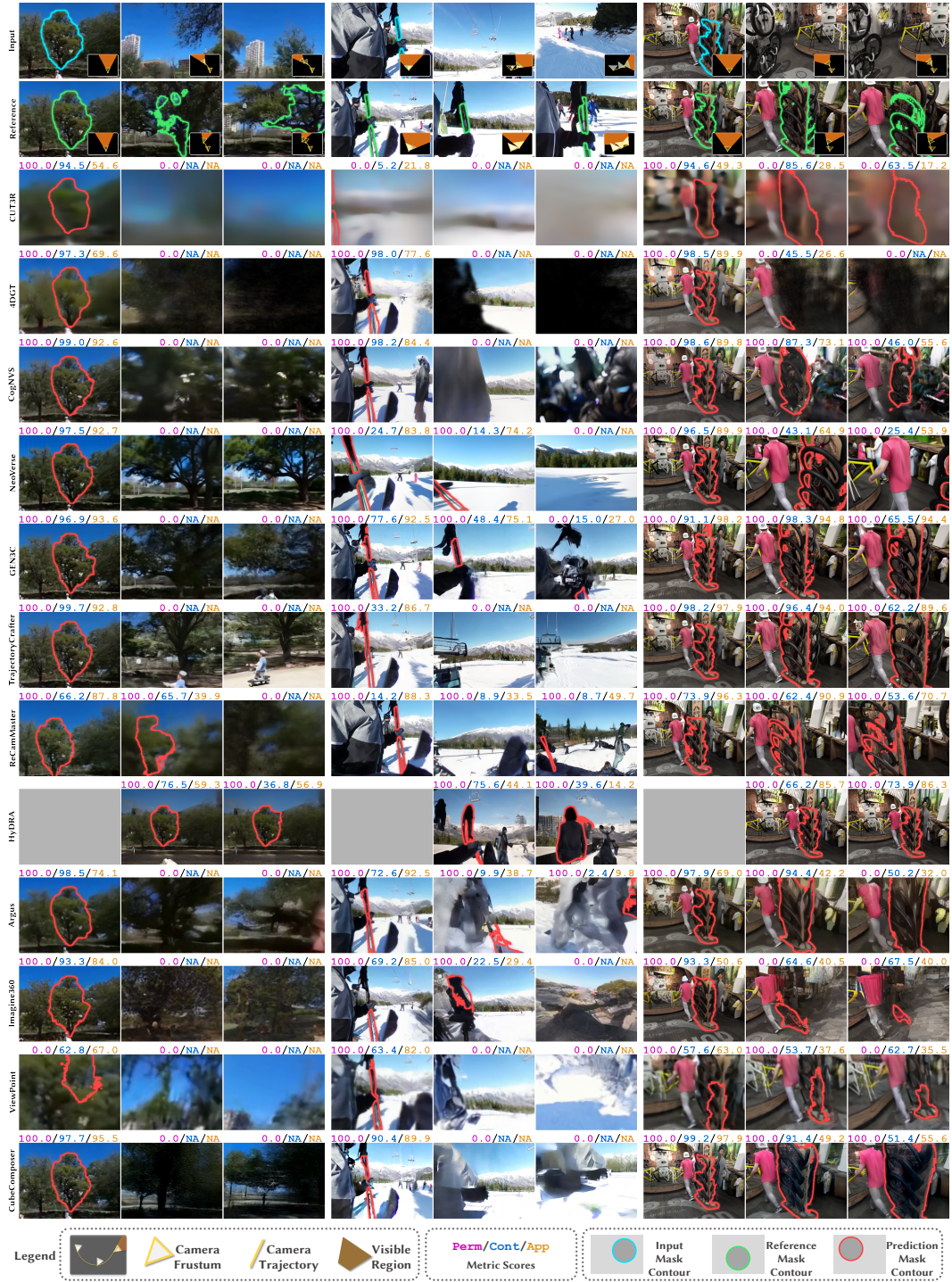


Figure 24: **Additional qualitative comparisons on the static subset.**

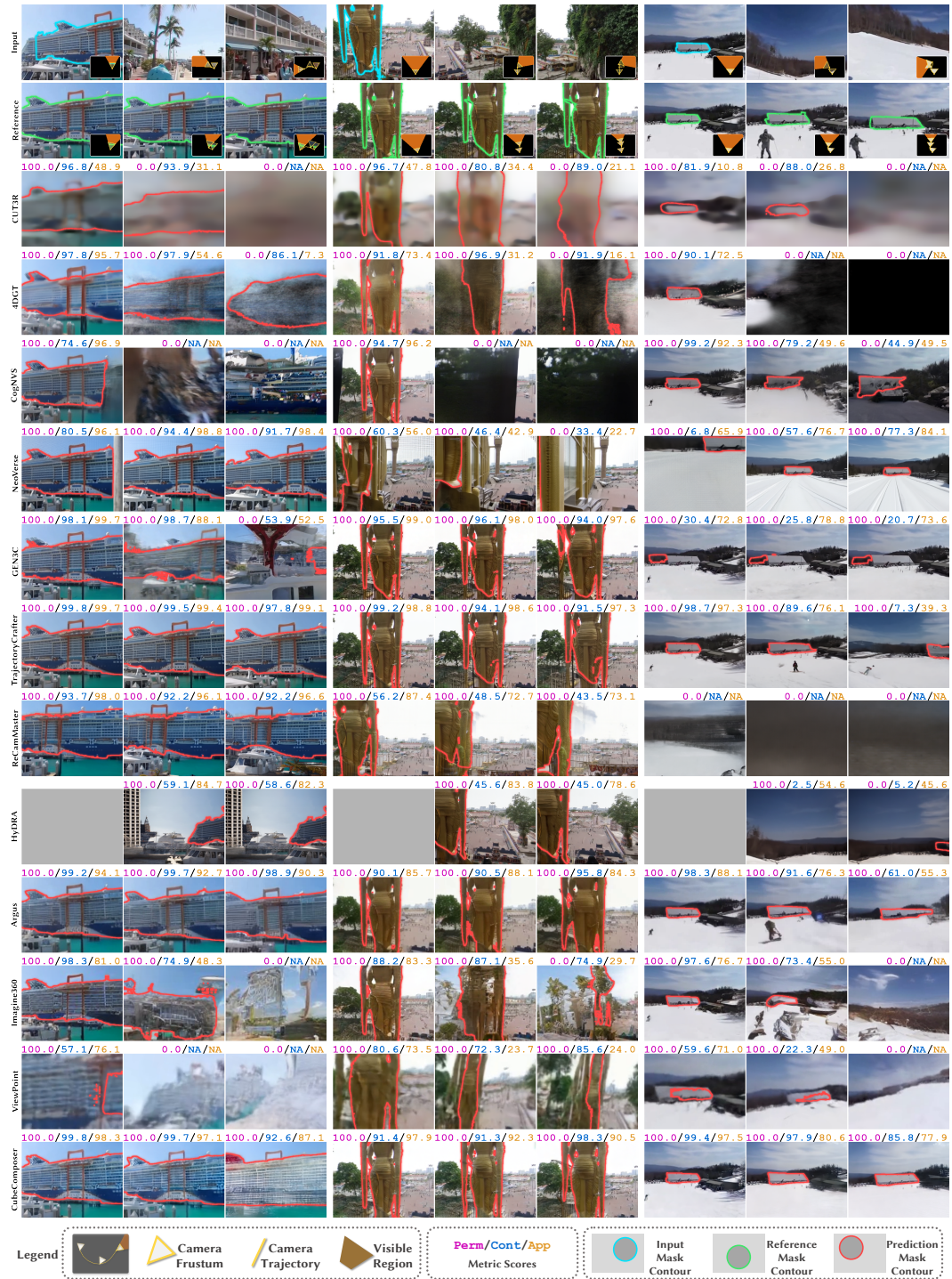


Figure 25: Additional qualitative comparisons on the static subset.