CAMERA×TIME FACTORIZATION FOR UNIFIED MULTI-CAMERA 4D RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-camera 4D reconstruction is important for embodied perception. Synchronized robot observations naturally form a camera×time structure. Cameras at the same timestamp provide complementary geometric views, while observations from the same camera capture temporal evolution. Existing methods often flatten these observations into a single sequence or process each camera independently, and thus do not explicitly model these two relations. In this paper, we first introduce a Camera×Time Formulation for embodied 4D reconstruction. According to formulation, we build **RoboCT4D**, a unified benchmark that standardizes camera×time observations and evaluation across simulated and real robot manipulation datasets. We then propose **CxT4D** with Camera×Time Factorization. It encodes camera and temporal identities separately and alternates global attention across synchronized cameras and along individual camera streams. This design explicitly models cross-camera geometry and temporal correspondence. We further propose a Long-Sequence MV4D Strategy that reduces memory cost while preserving reconstruction accuracy. Experiments on four RoboCT4D subsets show that CxT4D improves depth and cross-camera rotation accuracy over existing methods. Our long-sequence inference reduces peak memory by 51.0% at eight timestamps and 60.3% at twelve timestamps, and enables longer inputs that otherwise run out of memory.

1 INTRODUCTION

Recovering scene geometry and motion is fundamental to embodied perception. A robot observes dynamic interactions through multiple cameras with different viewpoints and motion patterns, so reliable reconstruction must reason jointly across space and time. Recent feed-forward models recover geometry from image pairs or multiple views (Wang et al., 2024; Leroy et al., 2024; Wang et al., 2025a), and dynamic methods extend reconstruction to time-varying geometry and trajectories (Zhang et al., 2025; Karhade et al., 2025; Sucar et al., 2026; Jiang et al., 2026). These advances provide a strong foundation for reconstructing robot interactions from visual observations.

However, synchronized robot observations have a structure that generic video or unordered image collections do not capture. At each timestamp, multiple cameras observe the same scene state from different viewpoints; across timestamps, each camera records a temporal stream. Together, these observations form a natural camera×time structure, as illustrated in Figure 1. The two axes encode different relations: the camera axis provides complementary geometric evidence, while the temporal axis captures scene evolution. Existing methods often flatten these observations into a single sequence or process each camera independently, which obscures the distinction between cross-camera geometry and temporal correspondence.

We therefore introduce a **Camera×Time Formulation** for embodied 4D reconstruction. The formulation treats synchronized multi-camera observations as a structured camera×time grid rather than a flat sequence of frames, making explicit the two key relations in robot reconstruction: geometry across synchronized cameras and temporal evolution within each camera stream.

According to formulation, we build **RoboCT4D**, a unified benchmark for synchronized multi-camera 4D reconstruction. RoboCT4D combines newly collected RobotWin2-4D rollouts with converted FurnitureBench and AgiBot-World recordings. Its four subsets cover clean and randomized simulation, real assembly scenes, and recordings with moving cameras, as shown in Figure 2. We

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

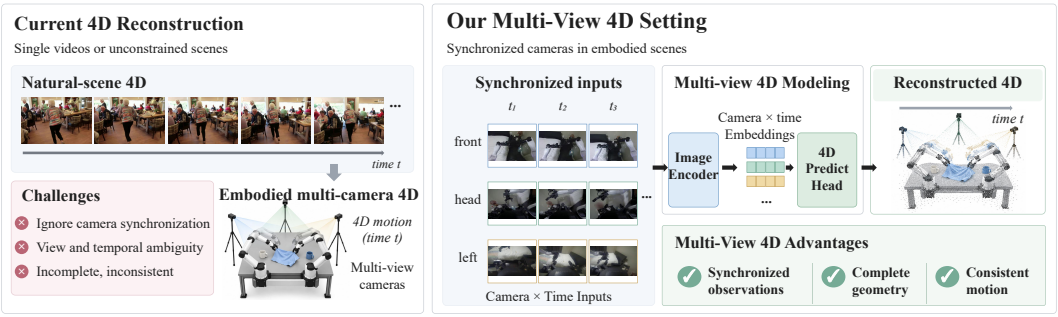

Figure 1: Conventional 4D reconstruction methods are primarily designed for generic video observations and do not explicitly exploit the synchronized multi-camera structure available in robot manipulation scenes. In contrast, **CxT4D** organizes observations in a camera×time grid, jointly leveraging complementary synchronized views for geometry and temporal observations for scene evolution, enabling complete and consistent 4D reconstruction for embodied robot manipulation.

standardize episode structure, camera conventions, temporal sampling, and depth and pose evaluation, making heterogeneous robot datasets comparable under the same camera×time setting.

We then propose **CxT4D**, a structured adaptation of OmniX (Jiang et al., 2026) with **Camera×Time Factorization**. CxT4D encodes camera identity and temporal position separately, and its global attention alternates across synchronized cameras and along individual camera streams. Cross-camera attention aggregates geometric evidence from different viewpoints at the same timestamp, while cross-time attention relates observations from the same camera over time. This factorized design directly models the two relations defined by the camera×time formulation while retaining the original prediction heads and training objectives.

Long robot sequences introduce a separate practical challenge: memory consumption grows quickly with the number of input views. We therefore introduce a **Long-Sequence MV4D Strategy** for memory-efficient inference. The strategy reduces memory cost while preserving multi-camera context and reconstruction accuracy, making longer camera×time inputs practical.

Experiments on all four RoboCT4D subsets validate the proposed formulation and model. CxT4D consistently improves depth and cross-camera rotation accuracy over existing methods. A paired study on 100 randomized RobotWin2-4D episodes further shows lower depth error for every camera under joint multi-camera inference. Our long-sequence inference reduces peak memory by 51.0% at eight timestamps and 60.3% at twelve timestamps, and enables longer inputs that otherwise run out of memory. Temporal pose improvements are less uniform, which we report in the complete results.

In summary, our contributions are:

- We introduce a **Camera×Time Formulation** for embodied 4D reconstruction and build **RoboCT4D**, a unified benchmark that standardizes synchronized multi-camera observations and evaluation across simulated and real robot manipulation datasets.
- We propose **CxT4D** with **Camera×Time Factorization**. It encodes camera and temporal identities separately and alternates global attention across cameras and time to model cross-camera geometry and temporal correspondence.
- We introduce a **Long-Sequence MV4D Strategy** for memory-efficient inference. It substantially reduces memory cost while preserving reconstruction accuracy and enables longer camera×time inputs.

2 Related Work

Our work connects three lines of research: geometric reconstruction and correspondence learning, dynamic 4D reconstruction, and robot datasets for embodied perception. We focus on how these directions relate to the camera×time structure of synchronized multi-camera robot observations.

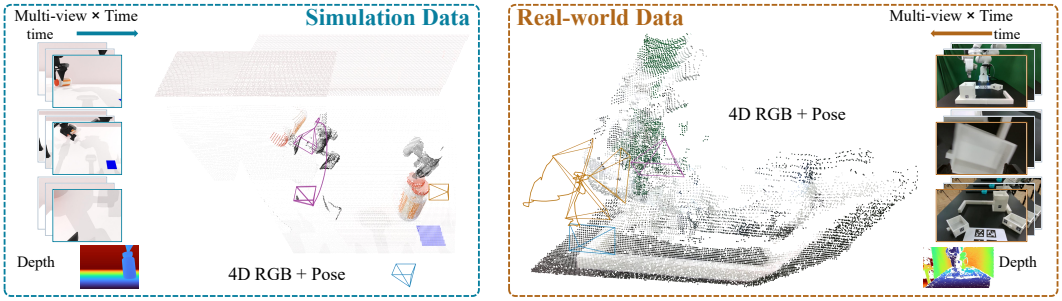

Figure 2: RoboCT4D contains four subsets which span simulated and real robot scenes. Each panel shows one episode with multi-camera RGB observations, depth, camera poses, and accumulated RGB point clouds. Stacked images indicate time.

Geometry and correspondence learning. NeRF (Mildenhall et al., 2020) and 3D Gaussian Splatting (Kerbl et al., 2023) optimize scene representations from images. Feed-forward methods instead predict geometry directly with learned models. DUSt3R (Wang et al., 2024) estimates pairwise point maps, and MASt3R (Leroy et al., 2024) extends this representation to image matching. VGGT (Wang et al., 2025a) and Depth Anything 3 (Lin et al., 2025) support multi-view geometric prediction. L2M (Liang et al., 2025) uses lifted geometry to learn dense matching features. These methods show the importance of geometric structure in visual reconstruction. Our work considers a different structure in embodied settings: synchronized observations naturally form a camera×time grid with distinct cross-camera and temporal relations.

Dynamic reconstruction and tracking. MonST3R (Zhang et al., 2025) reconstructs geometry from dynamic video. CUT3R (Wang et al., 2025b) uses a persistent state for sequential 3D perception. Any4D (Karhade et al., 2025), V-DPM (Sucar et al., 2026), and VGGT4D (Hu et al., 2026) further develop feed-forward dynamic reconstruction. OmniX (Jiang et al., 2026) predicts depth, camera pose, and trajectory fields from any-view, any-time inputs. Point trackers such as TAPIR (Doersch et al., 2023) and CoTracker3 (Karaev et al., 2025) focus on temporal correspondence. These methods provide strong models for multi-view and dynamic reconstruction, but do not explicitly distinguish synchronized camera relations from temporal relations in robot observations.

Factorized attention has been studied in video modeling. TimeSformer (Bertasius et al., 2021) separates spatial and temporal attention, while ViViT (Arnab et al., 2021) studies several factorized transformer designs. CxT4D introduces **Camera×Time Factorization** for synchronized multi-camera reconstruction. Cross-camera interaction operates among views of the same scene state, while cross-time interaction follows each camera stream over time. This design encodes the camera×time formulation directly into feature interaction. We further develop a **Long-Sequence MV4D Strategy** to make this structured reconstruction practical for longer inputs.

Robot datasets and geometric evaluation. RoboTwin 2.0 (Chen et al., 2026), FurnitureBench (Heo et al., 2023), and AgiBot-World (AgiBot-World-Contributors et al., 2025) provide robot manipulation data across simulated and real environments. CALVIN (Mees et al., 2021) and LIBERO (Liu et al., 2023) support language-conditioned and lifelong robot learning. ManiSkill3 (Tao et al., 2024) and RoboCasa365 (Nasiriany et al., 2026) further expand simulation diversity and scale. However, these datasets use different camera configurations, annotations, and evaluation protocols. **RoboCT4D** standardizes them under a common camera×time formulation for multi-camera 4D reconstruction, with unified evaluation across simulation and reality.

3 RoboCT4D: A Unified Camera×Time 4D Benchmark

RoboCT4D standardizes the **Camera×Time Formulation** for embodied 4D reconstruction across heterogeneous robot datasets. We represent an episode as

$$\mathcal{X} = \{I_{c,t}\}_{c=1,t=1}^{C,T}, \tag{1}$$

Table 1: **RoboCT4D subsets.** Camera names follow the normalized episode interface. Both simulated subsets contain 49 tasks; FurnitureBench and AgiBot-World contain four and one task groups, respectively.

Subset	Domain	Episodes	Depth cameras	Fixed cameras
RobotWin2-4D clean	Simulated	245	All	Front, head
RobotWin2-4D randomized	Simulated	245	All	Front, head
FurnitureBench	Real	12	All	Front, left
AgiBot-World	Real	69	Front	None

where c indexes cameras and t indexes synchronized timestamps. Fixing t gives multiple views of the same scene state, while fixing c gives a temporal stream. RoboCT4D provides a shared interface for these observations, together with common temporal sampling, annotation validity, and geometric evaluation. Figure 2 shows the four benchmark subsets and their multi-camera observations.

Data sources. Table 1 summarizes the four subsets. We collect clean and randomized RobotWin2-4D rollouts using RoboTwin 2.0 (Chen et al., 2026), and convert FurnitureBench (Heo et al., 2023) and AgiBot-World (AgiBot-World-Contributors et al., 2025) recordings to the same interface. Together, these subsets cover simulated and real manipulation scenes with different appearances and camera configurations. Each episode contains synchronized RGB images, camera intrinsics, camera poses, and available depth annotations. We normalize camera conventions and retain source validity masks. Pixels without valid depth are excluded from evaluation. Dynamic annotations are preserved when available, although the current benchmark evaluates depth and camera pose rather than dense object trajectories.

Camera configurations and evaluation. RobotWin2-4D and FurnitureBench include stationary cameras, while all evaluated AgiBot-World cameras move. RoboCT4D therefore covers both fixed- and moving-camera settings. The standard protocol samples three cameras and eight synchronized timestamps, starting from frame zero with an interval of eight source frames. Depth evaluation reports scale- and affine-invariant errors. Pose evaluation measures both cross-camera geometry and temporal camera motion. Detailed preprocessing, metrics, and inference settings are given in Sec. 5.1.

4 CxT4D: Camera×Time Factorization for 4D Reconstruction

Figure 3 gives an overview of CxT4D. Following the Camera×Time Formulation, CxT4D embeds the same structure into the OmniX backbone (Jiang et al., 2026). At the network level, it separates camera and temporal identities and factorizes global attention along the two axes. Cross-camera interaction aggregates synchronized geometric evidence, while cross-time interaction captures temporal evolution within each camera stream. We further introduce a Long-Sequence MV4D Strategy for memory-efficient inference. CxT4D retains the original OmniX prediction heads and training objectives.

4.1 Camera×Time Encoding

Given synchronized observations $\mathcal{X}$, CxT4D predicts per-view depth and camera pose with the OmniX backbone (Jiang et al., 2026), while retaining its trajectory head. We first encode the camera×time coordinates explicitly.

Let $\mathbf{x}_{c,t,n}$ denote token n from camera c at timestamp t. We add separate camera and temporal embeddings:

$$\tilde{\mathbf{x}}_{c,t,n} = \mathbf{x}_{c,t,n} + \mathbf{e}_c^{\mathrm{cam}} + \mathbf{e}_t^{\mathrm{time}}. \tag{2}$$

The camera embedding identifies the input stream, while the temporal embedding identifies its position in the sequence. These embeddings preserve the two axes of the camera×time structure and provide coordinates for the factorized attention.

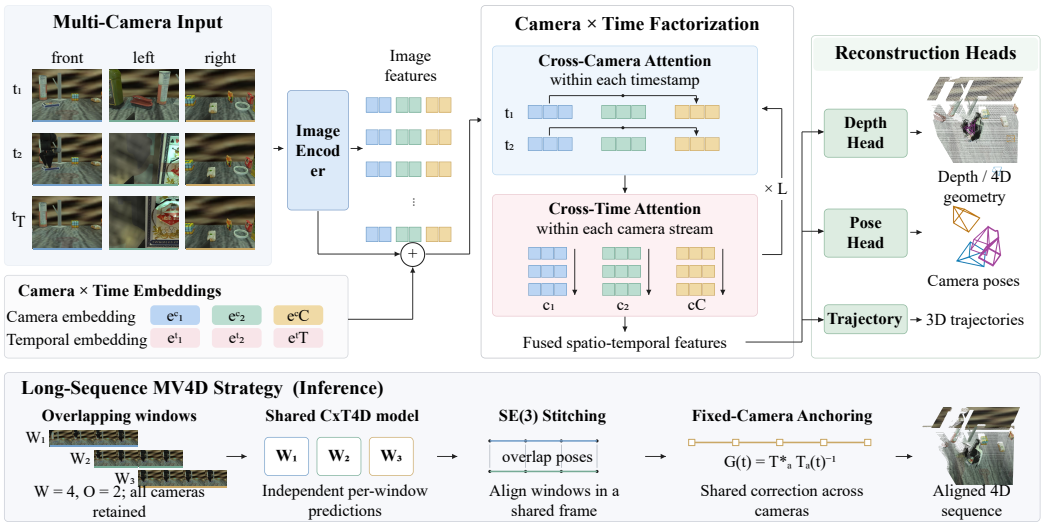

Figure 3: **CxT4D overview.** Top: camera and temporal embeddings encode the synchronized grid; global attention alternates across cameras and time before the OmniX prediction heads. Bottom: overlapping windows retain all cameras, are aligned in SE(3), and receive fixed-camera pose anchoring. Moving-camera smoothing is disabled. RGB, depth, and point-cloud examples come from a RobotWin2-4D episode. Output and window examples use ground truth for illustration, not model predictions. Attention groups and pose corrections are schematic.

4.2 Camera×Time Factorized Attention

Dense global attention mixes all CTN tokens, where N is the number of tokens per image. This places camera and temporal relations in the same attention space. CxT4D instead factorizes global attention along the two axes.

Even global layers perform **cross-camera attention** at each timestamp, while odd global layers perform **cross-time attention** within each camera stream. Their attention supports are

$$\mathcal{S}_{\text{cam}}(t) = \{(c, t, n) : 1 \le c \le C,\ 1 \le n \le N\}, \tag{3}$$

$$\mathcal{S}_{\text{time}}(c) = \{(c, t, n) : 1 \le t \le T,\ 1 \le n \le N\}. \tag{4}$$

Cross-camera attention aggregates complementary geometry from synchronized views. Cross-time attention relates observations from the same camera over time. Alternating the two operations allows information to propagate across the full camera×time grid. Figure 3 illustrates this factorization, and Sec. 5.4 analyzes the learned attention patterns.

Global layer	Fixed axis	Sequence length	Effective batch
Even	Timestamp t	CN	BT
Odd	Camera c	TN	BC

Here, B denotes batch size. Factorization reduces the longest global attention sequence from CTN to $\max(C, T)N$ tokens. We keep the alternating assignment fixed during training and inference. Nonrectangular camera×time inputs fall back to dense attention. The factorized interaction is learned during fine-tuning.

4.3 Long-Sequence MV4D Strategy

Camera×time factorization shortens global attention sequences, but memory still grows with the number of input views. In particular, activations in the depth, offset, and trajectory heads scale with sequence length. We therefore introduce a **Long-Sequence MV4D Strategy** that processes long inputs efficiently while preserving multi-camera context.

Temporal windowing and stitching. We divide the input along the temporal axis into overlapping windows. Each window retains all cameras and contains W timestamps. With an overlap of O timestamps, consecutive windows advance by $W - O$ frames. Under the standard $T{=}8$ setting, we use $W{=}4$ and $O{=}2$, giving windows $[0, 4)$, $[2, 6)$, and $[4, 8)$.

Each window is processed by the same CxT4D model. Overlapping camera poses provide correspondences between adjacent windows. We estimate an SE(3) transformation from these shared predictions and align each new window to the accumulated reconstruction. Predictions from earlier windows are retained in the overlap, while new timestamps are appended. This procedure preserves all sampled frames and requires no change to the model weights.

Fixed-camera pose anchoring. Robot camera rigs often contain stationary cameras. We use them as temporal references after window stitching. Let $\hat{\mathbf{T}}_{A,t} \in SE(3)$ denote the predicted camera-to-world pose of a fixed camera A. We define its reference pose by a robust temporal mean:

$$\mathbf{T}^*_A = \mathrm{RobustMean}_{SE(3)}\big(\{\hat{\mathbf{T}}_{A,t}\}_{t=1}^{T}\big). \tag{5}$$

For each timestamp, we compute a shared correction

$$\mathbf{G}_t = \mathbf{T}^*_A \hat{\mathbf{T}}^{-1}_{A,t}, \qquad \tilde{\mathbf{T}}_{c,t} = \mathbf{G}_t \hat{\mathbf{T}}_{c,t}. \tag{6}$$

We apply this correction to all cameras at the same timestamp. This reduces temporal pose drift while preserving cross-camera relative poses. When multiple fixed cameras are available, their temporal estimates also stabilize the reference geometry. Moving cameras receive only the shared correction and are not temporally smoothed. Depth and trajectory predictions remain unchanged. AgiBot-World contains no fixed camera, so no anchoring correction is applied.

Together, temporal windowing, SE(3) stitching, and fixed-camera anchoring form the Long-Sequence MV4D Strategy. The strategy reduces memory growth while preserving the camera×time structure used by CxT4D.

Training and implementation. Camera and temporal embeddings share a higher learning-rate group with the spatiotemporal and trajectory adapters. Training samples three cameras and two timestamps, with a random frame interval in $[1, 16]$. The training mixture includes RobotWin2-4D, FurnitureBench, AgiBot-World, CALVIN (Mees et al., 2021), LIBERO (Liu et al., 2023), ManiSkill3 (Tao et al., 2024), RoboCasa365 (Nasiriany et al., 2026), and depth-bearing RoboDojo data. We retain the OmniX camera-pose, depth, ray, 3D-trajectory, and dynamic-score objectives. All reported CxT4D results use the epoch-1 checkpoint with Camera×Time Factorization and the Long-Sequence MV4D Strategy. Multi-view KV merging and moving-camera smoothing are disabled.

5 Experiments

We evaluate three questions: whether Camera×Time Factorization improves reconstruction, whether joint multi-camera inference benefits individual views, and whether the Long-Sequence MV4D Strategy reduces memory for longer inputs. We report depth, cross-camera pose, and temporal camera errors.

5.1 Experimental Setup

Datasets and metrics. We evaluate 245 episodes from each RobotWin2-4D subset, 12 from FurnitureBench, and 69 from AgiBot-World. Unless stated otherwise, each input contains front, head, and left camera streams with eight synchronized timestamps. Sampling starts from frame zero with an interval of eight source frames. CxT4D and OmniX use $280{\times}504$ images. Depth evaluation uses valid annotations in $[0.05, 5.0]$ m with the released `pred_as_metric` setting. We report scale-invariant (SI) and affine-invariant (Aff.) depth errors. Pose metrics include cross-camera rotation and translation-direction errors (Rdeg and tdeg), and relative translation and rotation errors (RPEt and RPER). Absolute trajectory error (ATE) is retained in the released result files but omitted from the compact main tables. Lower values are better.

Inference settings. CxT4D uses Camera×Time Factorization with alternating cross-camera and cross-time global attention. For the Long-Sequence MV4D Strategy, we use temporal windows of $W{=}4$ timestamps with overlap $O{=}2$ under the standard protocol. Adjacent windows are aligned using overlapping camera poses. Fixed-camera anchoring is applied after stitching. Moving-camera smoothing is disabled. RobotWin2-4D uses front and head cameras as fixed references, while FurnitureBench uses front and left cameras. AgiBot-World has no fixed-camera constraint. Table 4 evaluates longer sequences separately.

The state-of-the-art methods for comparison. We compare CxT4D with Any4D (Karhade et al., 2025), V-DPM (Sucar et al., 2026), Depth Anything 3 (DA3) (Lin et al., 2025), VGGT4D (Hu et al., 2026), and OmniX (Jiang et al., 2026). External methods process each camera stream independently. Any4D and V-DPM use approximately 518-pixel preprocessing. DA3 uses `process_res=504`, while VGGT4D uses its tracker and two-stage refinement. All methods use the same sampled observations, but preprocessing and access to joint camera context differ. OmniX uses fixed-camera anchoring on the applicable subsets, and moving-camera smoothing on AgiBot-World. OmniX covers 243 RobotWin2-4D episodes per subset, while CxT4D covers 245 and uses no moving-camera smoothing. We therefore interpret cross-model comparisons with these implementation differences in mind. The long-sequence study keeps CxT4D weights and pose post-processing fixed, providing a controlled comparison of inference strategies.

5.2 Comparison with Existing Methods

Simulated scenes. Table 2 compares reconstruction accuracy on RoboCT4D. On RobotWin2-4D clean, CxT4D reduces SI from 0.0636 to 0.0175 relative to OmniX, a 72.5% reduction. Cross-camera rotation error decreases from 16.03° to 11.83°. On RobotWin2-4D randomized, SI decreases from 0.0259 to 0.0216 and rotation error from 18.64° to 15.91°. These results show that Camera×Time Factorization improves both depth and cross-camera geometry under clean and randomized simulation.

Real robot scenes. The gains also transfer to real observations. On FurnitureBench, SI decreases from 0.2544 to 0.1580 and rotation error from 58.24° to 19.07°. On AgiBot-World, SI decreases from 0.1120 to 0.0544 and rotation error from 10.58° to 2.71°. AgiBot-World contains no fixed camera, so these improvements do not rely on fixed-camera anchoring. Across all four subsets, CxT4D achieves the lowest reported depth and cross-camera rotation errors among the evaluated methods.

Temporal camera accuracy. Temporal camera improvements are less uniform. CxT4D improves RPER on both simulated subsets, while RPEt is slightly higher than OmniX on RobotWin2-4D randomized. These results separate the consistent gains in depth and cross-camera geometry from the remaining limitations in temporal camera estimation.

Single-view versus multi-view temporal inference. Table 3 compares joint CxT4D inference with independent OmniX inference on 100 randomized RobotWin2-4D episodes. Each pair evaluates the same camera. CxT4D lowers both depth errors for all three views. SI decreases from 0.0606 to 0.0157 on front, from 0.0756 to 0.0153 on head, and from 0.0472 to 0.0353 on left. Left-camera rotation error decreases from 28.20° to 25.22°. These results support joint multi-camera temporal inference for depth reconstruction. They compare complete systems and do not isolate multi-view context from fine-tuning or other model changes.

5.3 Long-Sequence Inference

Table 4 extends the comparison to longer inputs. Full-sequence inference processes all timestamps together and corresponds to "Dense" in the source measurements. At $T{=}12$, chunking reduces memory from 35.46 to 14.07 GiB, a 60.3% reduction. Runtime decreases from 9402 to 7495 ms, with slightly lower depth and rotation errors. At $T{=}16$, full-sequence inference runs out of memory on a 48 GiB GPU. Chunking completes the run at 19.86 GiB with an affine-invariant depth error of 0.1762. These results establish the memory benefit over the tested lengths. Sampling intervals differ between $T{=}8$ and longer inputs, so accuracy comparisons should remain within each pair.

Table 2: **Reconstruction accuracy on RoboCT4D.** Lower is better; bold marks the best value within each subset. Baselines use a single joint forward over all $C \times T$ frames (`flat_cxt`); evaluation differences are specified in Sec. 5.1.

	Depth		Cross-camera pose		Camera Trajectory	
Method	SI	Aff. ↓	Rdeg ↓	tdeg ↓	RPEt ↓	RPER ↓
RobotWin2-4D clean ($n = 245$)						
Any4D	0.1706	0.0801	25.01	53.35	0.0267	9.58
V-DPM	0.0596	0.0481	16.48	57.05	0.0263	4.43
DA3	0.0399	0.0316	15.86	55.91	0.0265	3.73
VGGT4D	0.0889	0.0599	15.26	56.86	0.0266	4.47
OmniX	0.0636	0.0282	16.03	37.87	0.0258	3.91
CxT4D	**0.0175**	**0.0143**	**11.83**	**35.37**	**0.0248**	**3.10**
RobotWin2-4D randomized ($n = 245$)						
Any4D	0.0910	0.0554	26.61	53.44	0.0421	8.70
V-DPM	0.0440	0.0452	19.42	52.17	0.0402	5.01
DA3	0.0390	0.0343	17.64	50.17	0.0414	4.32
VGGT4D	0.0654	0.0519	19.59	52.49	**0.0394**	5.44
OmniX	0.0259	0.0203	18.64	33.62	0.0414	4.63
CxT4D	**0.0216**	**0.0187**	**15.91**	**32.02**	0.0418	**3.99**
FurnitureBench ($n = 12$)						
Any4D	0.2638	0.2767	55.78	51.34	0.0207	23.76
V-DPM	0.2426	0.2634	60.16	55.21	0.0201	7.12
DA3	0.2381	0.2666	71.26	57.09	0.0198	8.72
VGGT4D	0.2954	0.3061	63.08	54.78	0.0212	7.45
OmniX	0.2544	0.2798	58.24	35.15	0.0218	6.24
CxT4D	**0.1580**	**0.1761**	**19.07**	**20.98**	**0.0148**	**4.57**
AgiBot-World ($n = 69$)						
Any4D	0.1602	0.1385	13.88	38.14	0.0146	1.67
V-DPM	0.1078	0.0997	10.77	31.63	0.0094	1.01
DA3	0.1115	0.1041	7.42	30.21	0.0083	1.09
VGGT4D	0.1236	0.1194	10.46	30.91	0.0098	1.02
OmniX	0.1120	0.1024	10.58	27.10	0.0084	1.05
CxT4D	**0.0544**	**0.0539**	**2.71**	**20.95**	**0.0078**	**0.99**

Table 3: **Single-view versus multi-view temporal inference.** Mean errors on 100 randomized RobotWin2-4D episodes. Each camera pairs independent OmniX inference with joint CxT4D inference. Bold marks lower errors; ties are both bold.

Camera	Inference	SI ↓	Aff. ↓	Rdeg ↓
Front (Fixed)	OmniX mono	0.0606	0.0377	-
	CxT4D multi-view	**0.0157**	**0.0135**	-
Head (Fixed)	OmniX mono	0.0756	0.0256	-
	CxT4D multi-view	**0.0153**	**0.0152**	-
Left (Moving)	OmniX mono	0.0472	0.0342	28.20
	CxT4D multi-view	**0.0353**	**0.0288**	**25.22**

5.4 Attention Analysis

Figure 4 visualizes context exchange along the two attention axes introduced by Camera×Time Factorization. We use eight timestamps, an interval of eight, and 280×504 images. The query token is selected from the head camera at $t = 4$. Cross-camera attention accesses synchronized front and left views at the same timestamp, while cross-time attention accesses other timestamps within the head

Table 4: **Sequence-length evaluation on FurnitureBench.** Full sequence denotes the source Dense setting. At T=16, it runs out of memory (OOM) on a 48 GiB GPU. Lower is better.

T (interval)	Inference	ms/ep ↓	Peak GiB ↓	Aff. ↓	Rdeg ↓
8 (i=8)	Full sequence	4009	19.85	0.1768	19.61
	Chunk	**3716**	**9.72**	**0.1761**	**19.07**
12 (i=4)	Full sequence	9402	35.46	0.1784	19.69
	Chunk	**7495**	**14.07**	**0.1775**	**18.74**
16 (i=4)	Full sequence	OOM	OOM	–	–
	Chunk	11969	19.86	0.1762	18.68

camera stream. These groups follow Equation 4. The visualizations show different interaction patterns along the two axes. Cross-camera attention aggregates information from complementary views observing the same scene state, with responses often focusing on shared objects and workspace regions. Cross-time attention distributes information across multiple timestamps within a camera stream, capturing temporal context under viewpoint changes and object motion.

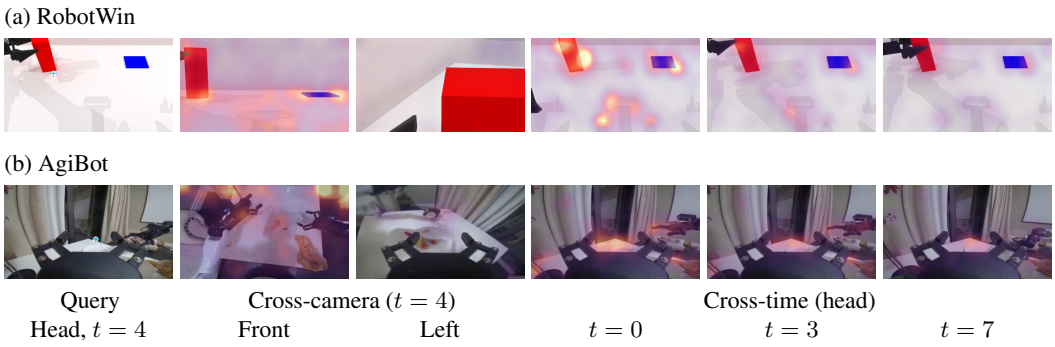

Figure 4: **Attention across cameras and time.** Final e1 attention overlays for a head-camera query at $t = 4$ (source frame 32). Warm regions indicate stronger displayed attention.

6 Conclusion

In this paper, we introduced the Camera×Time Formulation as a structured view of synchronized multi-camera observations for embodied 4D reconstruction. By explicitly separating the camera dimension, which provides complementary geometric evidence, from the temporal dimension, which captures scene evolution, the formulation reveals an important structure shared by many robot perception systems. Based on this formulation, we developed RoboCT4D, a unified benchmark that standardizes multi-camera 4D reconstruction evaluation across simulated and real robot manipulation datasets. We further proposed CxT4D with Camera×Time Factorization. By encoding camera and temporal identities separately and alternating cross-camera and cross-time attention, CxT4D enables explicit interaction between synchronized viewpoints and temporal observations while preserving the original reconstruction objectives. We also introduced the Long-Sequence MV4D Strategy, which reduces inference memory cost and enables reconstruction from longer camera×time inputs. Experiments on four RoboCT4D subsets demonstrate consistent improvements in depth estimation and cross-camera pose accuracy. Joint multi-camera inference further shows the benefit of modeling synchronized views together, while long-sequence inference substantially reduces memory consumption and extends the feasible input length. Temporal camera estimation remains a challenging direction, and future work can further explore longer-horizon reconstruction, dense dynamic object motion evaluation, and broader embodied environments. Overall, RoboCT4D and CxT4D provide a structured benchmark and modeling perspective for studying camera×time observations, offering a step toward scalable and consistent 4D reconstruction for embodied agents.

AI USE STATEMENT

Generative AI tools assisted with language editing, figure organization, and consistency checks. The authors are responsible for the manuscript and its reported results.

REPRODUCIBILITY STATEMENT

Section 5.1 specifies evaluation and inference settings. Main CxT4D results and overlap ablations use six released result JSON files. Longer-sequence and single-view comparisons use separately supplied measurements. Appendix A records their provenance and dependencies, including metrics omitted from the compact main tables.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6836–6846, 2021.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 813–824, 2021.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *International Conference on Machine Learning (ICML)*, 2026.

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

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. FurnitureBench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/rss.2023.xix.041.

Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 414–424, 2026.

Yanqin Jiang, Tengfei Wang, Zhengwei Wang, Chenjie Cao, Junta Wu, Wenhan Luo, Weiming Hu, Jin Gao, and Chunchao Guo. OmniX: Any-view and any-time 4d reconstruction via feed-forward trajectory fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6013–6022, 2025.

Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4D: Unified feed-forward metric 4d reconstruction. *arXiv preprint arXiv:2512.10935*, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with MASt3R. In *European Conference on Computer Vision (ECCV)*, pp. 71–91. Springer, 2024. doi: 10.1007/978-3-031-73220-1_5.

Yingping Liang, Yutao Hu, Wenqi Shao, and Ying Fu. Learning dense feature matching via lifting single 2d image to 3d space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6621–6631, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *arXiv preprint arXiv:2112.03227*, 2021.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR)*, 2026.

Edgar Sucar, Eldar Insafutdinov, Zihang Lai, and Andrea Vedaldi. V-DPM: 4d video reconstruction with dynamic point maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14502–14511, 2026.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. ManiSkill3: GPU parallelized robotics simulation and rendering for generalizable embodied AI. *arXiv preprint arXiv:2410.00425*, 2024.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, 2024.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations (ICLR)*, 2025.

A Final Recipe and Result Provenance

The recipe uses factored global attention, W=4, O=2, and `cxt_anchor`, with `smooth_moving=false`. AgiBot-World has no fixed cameras and receives no anchoring correction. Main results use the `mean_chunked` aggregates in the four subset JSON files. The full-window ablation uses `mean_full`; the overlap-1 ablation uses `mean_chunked`. Earlier checkpoint scans used different post-processing and do not form a controlled epoch ablation. Unavailable matched component controls remain empty. Table 4 incorporates the separately supplied sequence-length measurements; the $T = 12$ and $T = 16$ rows are not contained in the six original release JSON files. The 100-episode single-view comparison is supplied as an episode-mean summary. Its values are preserved in Table 3. The supplied package contains no model weights, training implementation, or complete wrapper dependencies. Reproducing inference requires the corresponding OmniX/CxT4D environment, checkpoint, and datasets.

B Supplementary Experiments

B.1 Qualitative Results

Figure 5 shows depth predictions for randomized RobotWin2-4D and AgiBot-World. CxT4D separates the foreground bottle from the workspace in the randomized scene and preserves arm silhouettes in the AgiBot view. Surface smoothness and local boundaries still vary across methods. Each depth map is independently normalized, so the visualizations support structural comparison rather than metric-scale agreement. Table 2 provides the quantitative evidence.

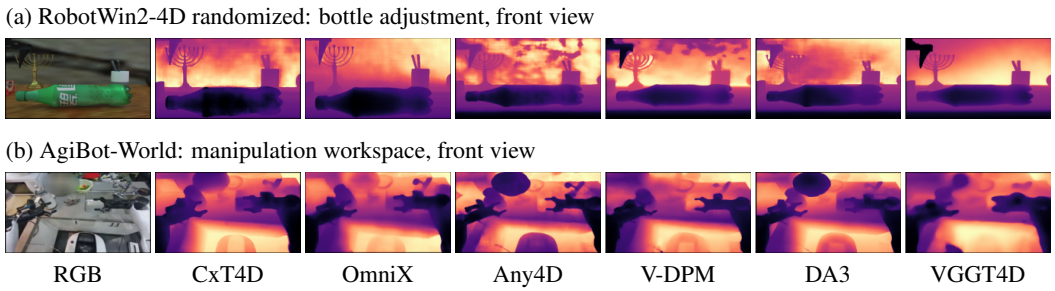

Figure 5: **Qualitative depth comparison.** Selected views from the supplied comparison archive at timestamp index $t = 4$ (source frame 32). All methods are shown for the same view. Depth maps are normalized independently, as in the source visualizations; colors do not share a metric scale. No ground-truth depth is shown.

B.2 Ablation and Efficiency Details

Table 5 summarizes the available component comparisons. Full-sequence factored inference and chunked inference share model weights and pose post-processing, so their difference isolates the inference schedule. Matched OmniX and dense fine-tuning controls are unavailable and remain blank.

Table 7 summarizes the available FurnitureBench efficiency measurements. Timing scopes differ across methods, so the controlled windowing study in Table 6 provides stronger evidence for the memory benefit.

C Additional Qualitative Comparisons

The following panels preserve all three camera views and all supplied methods for the selected AgiBot-World episode. Image panels are reproduced from the comparison archive without recoloring. The source label "ours" is displayed as CxT4D. Independent depth normalization prevents direct metric-scale comparison. These panels provide context for the selected views in Figure 5.

Table 5: **Component study on FurnitureBench.** FT: camera×time fine-tuning; F: factored attention; C: chunking. Completed rows share weights and pose post-processing. Empty cells denote unavailable controls.

	Components			Accuracy			Efficiency	
Variant	FT	F	C	SI ↓	Aff. ↓	Rdeg ↓	ms/ep ↓	GiB ↓
Base Model	–	–	–	0.1991	0.2463	27.75	4261	19.96
+ C×T Encoding	✓	–	–	0.1622	0.2120	22.54	4288	20.01
+ Factored Attention	✓	✓	–	0.1588	0.1768	19.61	4009	19.85
+ Chunking	✓	✓	✓	**0.1580**	**0.1761**	**19.07**	**3716**	**9.72**

Table 6: **Temporal chunking on FurnitureBench.** Identical weights and pose post-processing, C=3, T=8. Lower is better; bold marks column minima.

Setting	SI ↓	Aff. ↓	Rdeg ↓	ms/ep ↓	Peak GiB ↓
Full T=8	0.1588	0.1768	19.61	4009	19.85
W=4, O=1	**0.1578**	**0.1757**	**18.79**	**3633**	**9.72**
W=4, O=2	0.1580	0.1761	19.07	3716	**9.72**

Table 7: **Released efficiency measurements on FurnitureBench.** Baselines use `flat_cxt`; timing scopes differ across methods.

Method	ms/ep ↓	Peak GiB ↓	SI ↓	Rdeg ↓
Any4D	**941**	10.43	0.2638	55.78
DA3	1663	12.00	0.2381	71.26
V-DPM	35258	17.30	0.2426	60.16
VGGT4D	21295	9.93	0.2954	63.08
OmniX	∼3261	19.96	0.2544	58.24
CxT4D	3716	**9.72**	**0.1580**	**19.07**

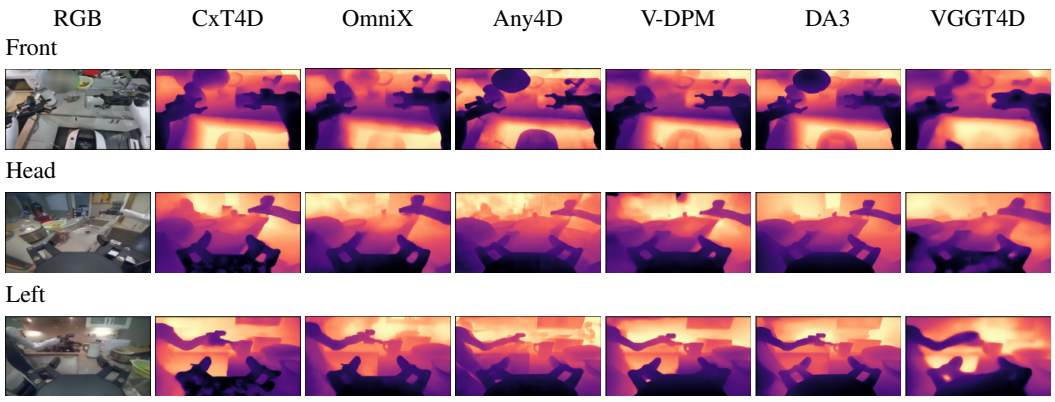

Figure 6: **AgiBot-World camera views.** Episode 0000 from task group 4713, timestamp index 4. Rows show front, head, and left views. Each depth map uses its own normalization.

D FRAME-LEVEL ATTENTION DISTRIBUTIONS

Figure 7 summarizes three episodes from the supplied Final e1 token visualizations: FurnitureBench drawer (episode 0000), RobotWin hammer (episode 0000), and AgiBot task 4713 (episode 0000). All use the standard eight-timestamp sampling protocol. For each query frame, attention is averaged over query patches and summed over key patches. Camera tokens are excluded, so rows need not sum to one. These matrices retain self-frame attention, unlike the RGB overlays in Figure 4.

The source uses block indices 13 and 17 for early cross-camera attention, and 29, 33, and 37 for late cross-camera attention. Cross-time groups use blocks 15 and 19, and 31, 35, and 39, respectively. These are source backbone block indices; even and odd in Sec. 4.2 refer to the alternating global-attention schedule. Each early/late pair shares a color scale within an episode and axis. The figures are plotted directly from the supplied attention arrays, without changing their values.

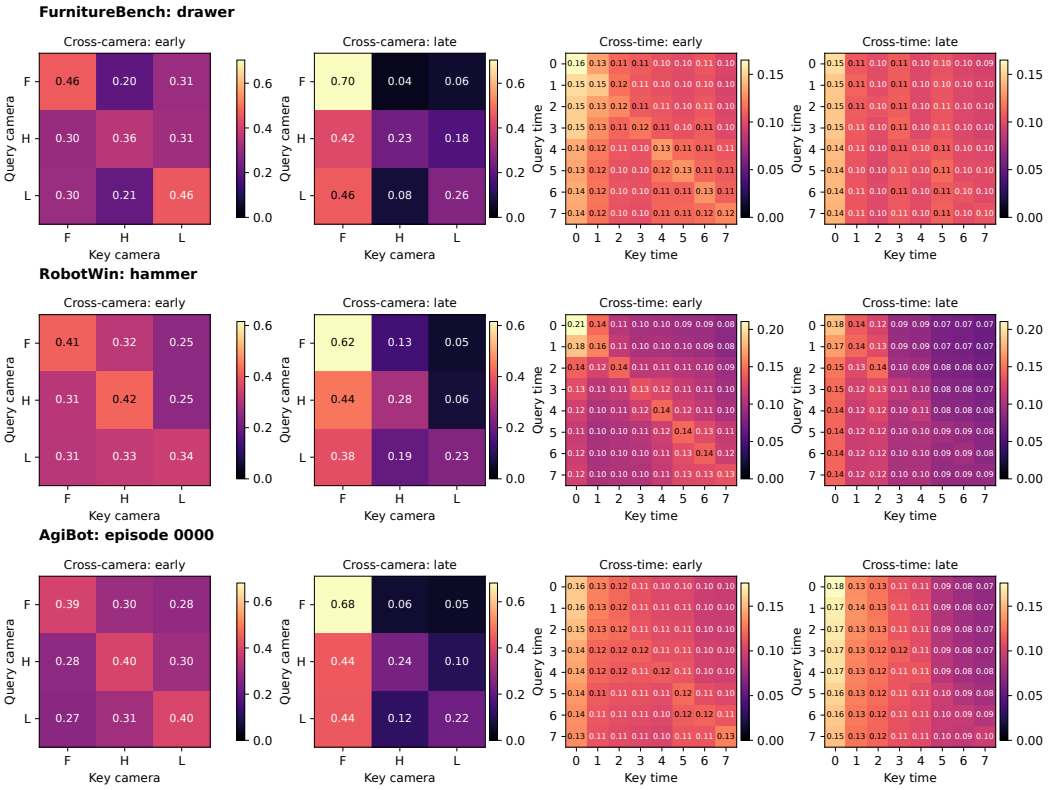

Figure 7: **Frame-level attention mass.** Rows show FurnitureBench, RobotWin, and AgiBot examples. Columns compare early and late cross-camera and cross-time attention. Matrix rows index queries; columns index keys. F, H, and L denote front, head, and left cameras. Nonzero off-diagonal entries indicate attention to other views or timestamps. Camera and temporal distributions differ across episodes and layers.