SYNCTAP: TRACKING ANY POINT FROM UNSYNCHRONIZED MULTI-VIEW VIDEOS

Anonymous authors
Paper under double-blind review

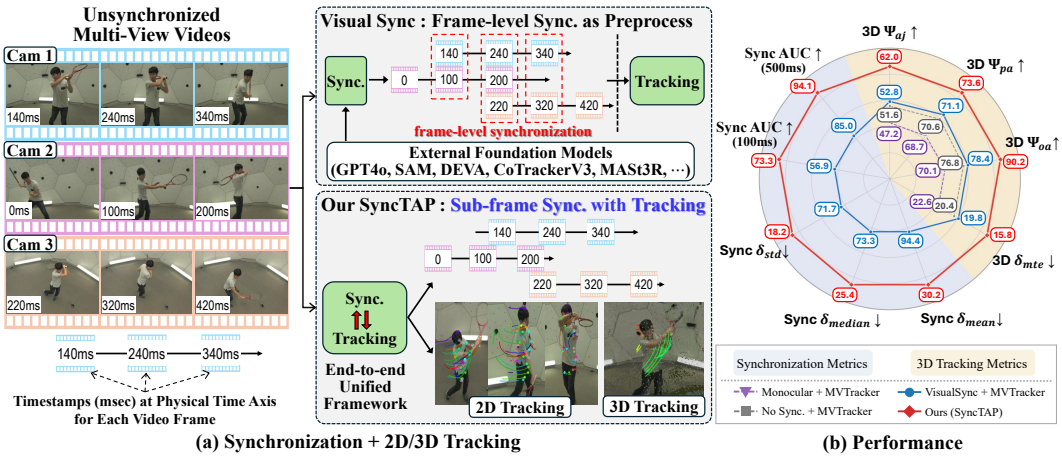

Figure 1: (a) We introduce **SyncTAP**, a novel unified framework that jointly estimates sub-frame-level time offsets and high-fidelity one-to-one correspondent 2D/3D trajectories for unsynchronized multi-view videos. (b) SyncTAP outperforms the baseline for both synchronization and 3D tracking.

ABSTRACT

Multi-view 3D point tracking is an important prior to understand dynamic 3D scenes, but typically assumes temporally aligned input videos. Even small time offsets collapse the underlying epipolar geometry, disrupting cross-view feature aggregation and severely degrading tracking performance. We propose **SyncTAP**, the first unified tracking framework that jointly estimates sub-frame per-view time offsets and high-fidelity one-to-one correspondent 2D/3D trajectories from unsynchronized multi-view videos. First, our *Epipolar-based Time Offset Initialization (EpiInit)* analytically finds initial time offsets using an epipolar constraint and a 2nd-order Taylor approximation. Then, a transformer-based refinement stage alternates two operations: (i) an *Epipolar-based Time Offset Refinement (EpiRef)* module updates time offsets via Sampson error landscapes; (ii) a *Spatio-Temporal-Cross-View (STCV) attention* module refines 2D/3D trajectories by introducing novel attention layers for robust feature aggregation under temporal misalignment. This joint optimization yields a mutually beneficial cycle between synchronization accuracy and tracking accuracy. Extensive experiments demonstrate that our SyncTAP achieves sub-frame synchronization accuracy, *significantly* outperforming SOTA visual-based synchronization method and establishing a robust new baseline for multi-view tracking under unsynchronized conditions.

1 INTRODUCTION

Tracking individual 3D points is a fundamental key to understand complex dynamic 3D scenes in real world. Many researchers have advanced the field of point tracking, from monocular 2D trajectory (Karaev et al., 2025; Doersch et al., 2024) to more challenging tasks of multi-view or 3D trajectory (Koo et al., 2026; Rajič et al., 2025; Galoaa et al., 2026; Xiao et al., 2025; Zhang et al.,

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.

2025) estimation. Among them, multi-view 3D point tracking (Rajič et al., 2025; Galoaa et al., 2026) has emerged as the most powerful geometric prior as it can incorporate diverse observations from multiple viewpoints and overcome view-specific occlusions, paving the way for future advancements in many downstream tasks such as dynamic 3D scene reconstruction.

Previous SOTA multi-view trackers heavily rely on cross-view attention (Koo et al., 2026) or geometric projection (Rajič et al., 2025; Galoaa et al., 2026) for cross-view feature aggregation, strictly assuming that all multi-view videos are synchronized. Even a few frames of time offsets easily collapse the epipolar geometry, severely degrading 2D/3D tracking performance. Therefore, aligning unsynchronized multi-view videos relying on visual cues is a highly difficult but very important task.

To overcome this challenging problem, we utilize a pretrained 2D tracker with the epipolar constraint as a geometric cue for multi-view synchronization. If the two frames from a video pair are in the same global timestamp, they can be assumed to share spatially consistent cross-view correspondences. For this, we propose **SyncTAP**, a novel unified framework that ***jointly* estimates *sub-frame-level* per-view time offsets and high-fidelity one-to-one correspondent 2D/3D trajectories for *unsynchronized* multi-view videos**. Our SyncTAP consists of a two-stage architecture with Initialization and Refinement: In the Initialization stage, our SyncTAP finds initial time offsets analytically based on the epipolar constraint resolved by the 2nd-order Taylor series approximation. In the Refinement stage, the initial time offsets and 2D/3D trajectories are iteratively refined by alternating the two operations: (i) our Epipolar-based Time Offset Refinement (EpiRef) module updates time offsets using Sampson error to capture the optimal time offsets yielding the highest cross-view consistency near the current time offset estimates; (ii) our Spatio-Temporal-Cross-View (STCV) attention network refines 2D/3D trajectories by enhancing its cross-view attention layers with temporal tolerance in order to robustly estimate trajectories under unsynchronized video inputs. This alternating refinement between the EpiRef module and STCV attention network benefits each other such that refined 2D trajectories provide more reliable epipolar landscape and more accurate time offsets lead to high-fidelity 2D/3D trajectories. Our SyncTAP shows *significantly improved* **synchronization performance** with **30.2 ms (0.36 frames)** in Mean Absolute Error on our 12 fps Kubric evaluation dataset where given input videos are unsynchronized with the mean time offset of 351.1 ms, achieving **68.0% reduction** compared to the SOTA visual-based synchronization method (Liu et al., 2025). Consequently, the **tracking performances** of our SyncTAP are superior to those of the cascaded baselines of (i) synchronization (Liu et al., 2025) and a SOTA 2D tracker (Koo et al., 2026) by **4.4%** improvement, and (ii) synchronization (Liu et al., 2025) and a SOTA 3D tracker (Rajič et al., 2025) by **17.4%** improvement in Average Jaccard. Our main contributions are:

- We propose **SyncTAP**, the first unified tracking framework for joint estimation of sub-frame per-view time offsets and one-to-one correspondent 2D/3D trajectories from unsynchronized multi-view videos.
- We present **Epipolar-based time-offset Initialization (EpiInit)**, an analytic method using an epipolar constraint and a Taylor series approximation for robust time offset initialization.
- **Epipolar-based time offset Refinement (EpiRef)** is proposed, which is a learning-based refinement that uses the Sampson error landscape for sub-frame time offset estimation.
- **Spatio-Temporal-Cross-View (STCV)** attention is proposed to robustly aggregate multi-view features and improve 2D/3D tracking under temporally misaligned multi-view videos.

2 Related Work

Monocular point tracking. Long-term point tracking has advanced rapidly since TAP-Vid (Doersch et al., 2022) and PIPs (Harley et al., 2022). Recent methods achieve highly accurate monocular 2D point tracking by introducing iterative refinement (Doersch et al., 2023), local 4D correlation volumes (Cho et al., 2024), cross-trajectory attention (Karaev et al., 2024; 2025), self-supervision (Doersch et al., 2024), high-resolution dense outputs (Harley et al., 2025), and online next-token prediction (Zholus et al., 2025). For monocular 3D point tracking, several methods lift 2D trajectories to the world-space using monocular depth (Xiao et al., 2024; 2025; Zhang et al., 2025) or achieve dense long-range tracking (Ngo et al., 2025). Reconstruction-centric approaches (Wang et al., 2025; Seidenschwarz et al., 2025; Feng et al., 2025) further integrate tracking with 4D scene reconstruc-

tion. Despite these advances, monocular point tracking suffers from unrecoverable occlusions and depth scale ambiguities for 3D tracking, necessitating multi-view camera setups.

Multi-view point tracking. For multi-view point tracking, classical pipelines rely on feature matching (Sarlin et al., 2020; Sun et al., 2021; Edstedt et al., 2024) and triangulation, but struggle on textureless surfaces. Other approaches obtain trajectories via Gaussian primitives (Luiten et al., 2024; Zheng et al., 2025). Recent advanced multi-view trackers estimate trajectories by aggregating per-view features from uplifted pixels (Rajič et al., 2025), geometric constraint masks (Galoaa et al., 2026), or cross-view attention (Koo et al., 2026). These tracking methods show good performance, but they commonly assume perfectly synchronized input videos for cross-view feature aggregation.

Multi-view synchronization. Recovering temporal alignment is crucial for unsynchronized multi-view setups. Early works utilize two-view geometric constraints (Albl et al., 2017; Rao et al., 2003). Some recent learning-based methods rely on domain-specific audio-visual cues (Iashin et al., 2024) or human pose (Choi et al., 2025). Other approaches integrate synchronization directly into per-scene optimization for 3D reconstruction with NeRF (Kim et al., 2024) or 3DGS (Xu et al., 2026; Lee et al., 2026). Other recent methods utilize geometric motion cues (Liu et al., 2025; Lee et al., 2026). Among them, VisualSync (Liu et al., 2025) estimates time offsets via exhaustive temporal search of the Sampson error relying on external foundation models (such as dynamic object recognition (Achiam et al., 2023), segmentation (Ren et al., 2024; Cheng et al., 2023), tracking (Karaev et al., 2025), and matching (Leroy et al., 2024)), which inherits the errors of those upstream models. To overcome this limitation, our SyncTAP itself analytically finds initial time offsets from an epipolar constraint resolved by 2nd-order Taylor series approximation, and iteratively refines both time offset estimation module and tracking module in a mutually beneficial alternating manner.

3 METHOD

3.1 PIPELINE OVERVIEW

Fig. 2 shows the overall architecture of our SyncTAP. Given V unsynchronized multi-view videos of a dynamic scene with known stationary camera parameters, depth maps, and 2D/3D query points, the pipeline of SyncTAP jointly estimates (i) the per-view time offsets $\{\tau_v \in \mathbb{R}\}_{v=1}^{V}$, (ii) the world-space 3D trajectories $\mathbf{T}_{3D} \in \mathbb{R}^{T\times N\times 3}$ with visibility $\mathbf{V}_{3D} \in \mathbb{R}^{T\times N\times 1}$, and (iii) the pixel-space 2D trajectories $\mathbf{T}_{2D} \in \mathbb{R}^{V\times T\times N\times 2}$ with visibility $\mathbf{V}_{2D} \in \mathbb{R}^{V\times T\times N\times 1}$, where T is the total number of each video's frames and N is the total number of trajectories in each video, through a two-stage framework with Initialization and Refinement. In the Initialization stage, EpiInit (Sec. 3.2) extracts 2D trajectories and analytically finds the initial time offset τ_v^0 for each view v using the epipolar constraint and the 2nd-order Taylor series approximation (TSA). In the Refinement stage (Sec. 3.3), time offsets τ_v and 2D/3D trajectories $\mathbf{T}_{2D}, \mathbf{T}_{3D}$ are refined in an alternating manner. At m-th iteration for $m \geq 0$, our EpiRef module (Sec. 3.3.1) predicts the time offset residual $\Delta\tau_v^{(m+1)}$ utilizing the local epipolar landscape, and our STCV attention network (Sec. 3.3.2) predicts the residual of 3D coordinates $\Delta\mathbf{T}_{3D}^{(m+1)}$, 3D feature $\Delta\mathbf{f}_{3D}^{(m+1)} \in \mathbb{R}^{T\times N\times h}$ (used to predict 3D visibility $\mathbf{V}_{3D}^{(M)}$ where h is the latent space dimension), 2D coordinates $\Delta\mathbf{T}_{2D}^{(m+1)}$, 2D visibility $\Delta\mathbf{V}_{2D}^{(m+1)}$, and 2D confidence $\Delta\mathbf{C}_{2D}^{(m+1)} \in \mathbb{R}^{V\times T\times N\times 1}$, where we newly introduce Cross-View Temporal Attention Layer and Unidirectional 2D-to-3D Attention Layer to robustly aggregate other 2D views' features under temporal misalignments. Then these residuals are added iteratively as $\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} + \Delta\mathbf{X}^{(m+1)}$ for those estimates $\mathbf{X} \in \{\tau_v, \mathbf{T}_{3D}, \mathbf{f}_{3D}, \mathbf{T}_{2D}, \mathbf{V}_{2D}, \mathbf{C}_{2D}\}$.

3.2 EPIINIT: EPIPOLAR-BASED TIME-OFFSET INITIALIZATION

Our EpiInit provides the initial per-view time offsets $\{\tau_v^0\}_{v=1}^{V}$ through an iterative two-step process: Pairwise-fitting and Global-fitting. In the Pairwise-fitting step, we obtain pairwise time offsets $T_{ij}^{n,t}$ for n-th trajectory pair between i-th video and j-th video at frame t, by applying a 2nd-order TSA to epipolar relationship to obtain a closed-form solution without an exhaustive temporal search. In the Global-fitting step, we obtain the initial per-view time offsets $\{\tau_v^0\}_{v=1}^{V}$ from all combinations of pairwise time offsets $\{T_{ij}\}_{i,j=1}^{V}$ via Laplacian solver (Barooah & Hespanha, 2008).

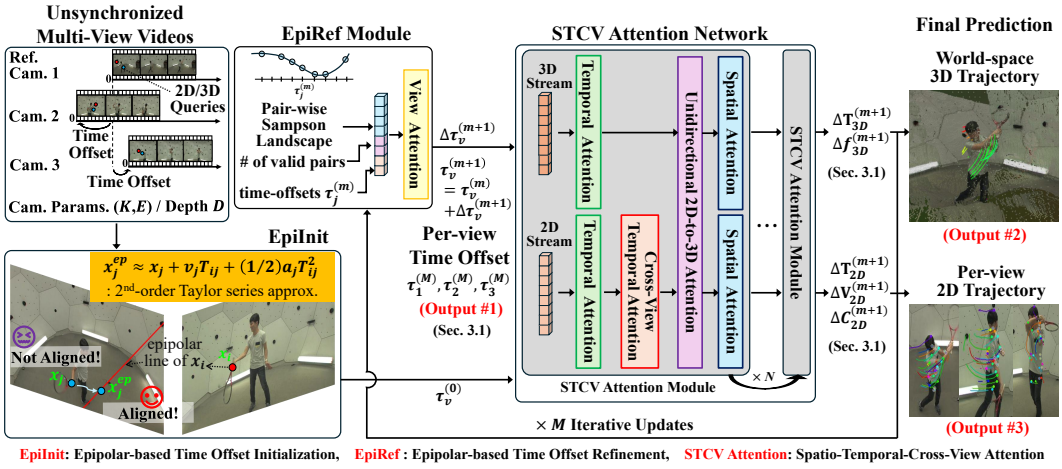

Figure 2: **Overall pipeline of SyncTAP.** For unsynchronized videos, our framework jointly estimates (i) sub-frame per-view time offsets and (ii) one-to-one correspondent 2D/3D trajectories in the two stages where the initialization stage (**EpiInit**) is described in Sec. 3.2, and the refinement stage alternates between **EpiRef** module (Sec. 3.3.1) and **STCV** attention network (Sec. 3.3.2).

Pairwise-fitting via Epipolar constraint and Taylor Series Approximation (TSA). We first find valid pairs of visible and dynamic 2D trajectory points $\{(\mathbf{x}_i^{n,t}, \mathbf{x}_j^{n,t})\}_{n=1,t=1}^{N,T}$. Since the view pair is not yet synchronized, the epipolar constraint is violated such that $\mathbf{x}_j^{n,t}$ is not on the epipolar line of $\mathbf{x}_i^{n,t}$, that is, $\mathbf{x}_j^{n,t\top} \mathbf{F}_{ij} \mathbf{x}_i^{n,t} \neq 0$, where $\mathbf{F}_{ij}$ is the fundamental matrix obtained from camera parameters of view i, j. However, we know how $\mathbf{x}_j^{n,t}$ moves by deriving velocity $\mathbf{v}_j^{n,t}$ and acceleration $\mathbf{a}_j^{n,t}$ from coordinates $\mathbf{x}_j^{n,t}$. Therefore, we can analytically predict when the tracking point will intersect the epipolar line of $\mathbf{x}_i^{n,t}$ in sub-frame precision via the following 2nd-order TSA:

$$\left(\mathbf{x}_j^{n,t} + \mathbf{v}_j^{n,t}\, T_{ij}^{n,t} + \tfrac{1}{2}\, \mathbf{a}_j^{n,t}\, (T_{ij}^{n,t})^2\right)^\top \mathbf{F}_{ij}\, \mathbf{x}_i^{n,t} \approx 0. \tag{1}$$

This quadratic equation yields a closed-form estimation of pairwise time offset $T_{ij}^{n,t}$ for each valid point pair. Detailed mathematical derivation is provided in Appendix A.1.

$$T_{ij}^{n,t} \;=\; -2C/(B \,+\, \mathrm{sign}(B)\sqrt{B^2 - 4AC}) \tag{2}$$

where $A = \frac{1}{2}\, \mathbf{a}_j^{n,t\top} \mathbf{F}_{ij}\, \mathbf{x}_i^{n,t}$, $B = \mathbf{v}_j^{n,t\top} \mathbf{F}_{ij}\, \mathbf{x}_i^{n,t}$, and $C = \mathbf{x}_j^{n,t\top} \mathbf{F}_{ij}\, \mathbf{x}_i^{n,t}$.

Global-fitting via Laplacian Solver. After the Pairwise-fitting step, a single view pair i, j has a distribution of pairwise time offsets $\{T_{ij}^{n,t}\}_{n=1,t=1}^{N,T}$. Rather than naively averaging, which can be vulnerable to outliers from inaccurate tracks, we aggregate point-pairwise time offsets $\{T_{ij}^{n,t}\}_{n=1,t=1}^{N,T}$ into a single representative view-pairwise time offset T_{ij} by taking the mode of their orthogonality-weighted histogram. The orthogonality upweights the well-posed dynamic points moving perpendicular to the epipolar line. Given these *view-pairwise* time offsets $\{T_{ij}\}$, we obtain the initial *per-view* time offsets $\{\Delta\tau_v^0\}_{v=1}^V$ by minimizing a weighted energy function for all view pairs as:

$$E(\Delta\tau^0) \;=\; \sum\nolimits_{j=1}^{V} \sum\nolimits_{i \neq j} W_{ij} \left(\Delta\tau_i^0 - \Delta\tau_j^0 - T_{ij}\right)^2, \tag{3}$$

where the confidence weight W_{ij} is derived from the number of valid point pairs and the sharpness of the pairwise time offset distribution. We can formulate this problem as Laplacian matrix equation which yields closed-form solution. Detailed mathematical derivation can be found in Appendix A.1. In Eq. 3, the Laplacian matrix becomes singular if any reference view does not exist, leading to an infinite number of solutions. To solve this, we apply Tikhonov regularization and shift the estimated per-view time offsets to set the first view (Camera 1) as the reference view ($\Delta\tau_1^0 = 0$).

Since we can obtain $\{\Delta\tau_v^0\}_{v=1}^V$ via the closed-form solutions for Eqs. 2 and 3, this eliminates the need for an exhaustive temporal search over the entire time. However, the 2nd-order TSA in Eq. 1 is

a local approximation which works well only near the current estimate with small $T_{ij}^{n,t}$. Therefore, EpiInit iteratively alternates between Pairwise-fitting and Global-fitting until convergence. We empirically found that most samples converge within nearly 4 iterations on our Kubric dataset. Detailed experiments are in Appendix A.2. Importantly, the initial sub-frame per-view time offsets $\{\Delta\tau_v^0\}_{v=1}^V$ become good starting points for the refinement of time offsets in the following Refinement stage.

3.3 Refinement Stage - Iterative Refinement of Time Offsets and Trajectories

Since the estimated 2D trajectories used in the EpiInit module are not accurate under cross-view attention of temporally misaligned features due to the absence of initially estimated time offsets, each pairwise time offset T_{ij} is often noisy from inaccurately estimated $\mathbf{v}_j$ and $\mathbf{a}_j$. So, we need to alternately refine (i) the time offset residual $\Delta\tau_v^{(m+1)}$ via the EpiRef module and (ii) the 2D/3D trajectory coordinates residuals, $\Delta\mathbf{T}_{2D}^{(m+1)}$ and $\Delta\mathbf{T}_{3D}^{(m+1)}$, via the STCV attention network. The strength of SyncTAP also lies in this alternating iterative refinement where they mutually benefit each other. That is, the improved time offset from the EpiRef module re-aligns per-view 2D information to yield more accurate trajectories, and the refined trajectories from the STCV attention network make the epipolar landscape sharper, leading to estimate the time offsets more accurately.

3.3.1 EpiRef: Epipolar-based Time Offset Refinement Module

The EpiRef module utilizes an epipolar constraint landscape as a hint for time offset estimation. This is motivated from the recent 2D point trackers (Cho et al., 2024), where they encode a local landscape of cosine similarities between the query point's features and nearby pixels' features so that transformer can recognize *which nearby pixel looks the most similar to the query point.* We apply this strategy analogously to the time offset refinement. For time offset estimate $\tau_j^{(m)}$ at m-th iteration for j-th view, we sample a set of time offset candidates, denoted as $\tau_{j,\text{cand}}^{(m)} = \tau_j^{(m)} + \delta \times s$, where $\delta \in \{-R, ..., R\}$ is an offset adjustment within a temporal range R and s is a scale factor for δ. We shift the current estimated 2D trajectory $\mathbf{x}_j^{(m)}(t_j)$ and known camera parameters $K_j(t_j), E_j(t_j)$ at frame t_j of view j by the candidate, which yields $\mathbf{x}_j^{(m)}(t_{\text{cand}})$, $K_j(t_{\text{cand}})$, and $E_j(t_{\text{cand}})$ where $t_{\text{cand}} = t_j - \delta \times s$. For error measurement, we utilize the Sampson error (Hartley & Zisserman, 2003) which approximates the distance from a point $\mathbf{x}_j^{(m)}(t_{\text{cand}})$ of view j to the epipolar line of the corresponding point $\mathbf{x}_i^{(m)}(t_i)$ of view i where $t_{\text{cand}} = t_i - (\tau_{j,\text{cand}}^{(m)} - \tau_i^{(m)})$. The Sampson error with the fundamental matrix $\mathbf{F}_{ij}$ between i-th camera at time t_i and j-th camera at time t_{cand} is:

$$E_{\text{Sam}}\left(\mathbf{x}_i^{(m)}, \mathbf{x}_j^{(m)}, \mathbf{F}_{ij}\right) = \left(\mathbf{x}_j^{(m)}(t_{\text{cand}})^\top \mathbf{F}_{ij} \mathbf{x}_i^{(m)}(t_i)\right)^2 \Bigg/ \left[\left\|\mathbf{F}_{ij}\mathbf{x}_i^{(m)}(t_i)\right\|_{1,2}^2 + \left\|\mathbf{F}_{ij}^\top \mathbf{x}_j^{(m)}(t_{\text{cand}})\right\|_{1,2}^2\right] \tag{4}$$

For each time offset candidate, we aggregate point-pairwise Sampson errors E_{Sam} into view-pairwise Sampson errors $\sum E_{\text{Sam}}^{ij}$ between view i and j. Then we concatenate the errors across all time offset candidates and map them into min-max normalized matching scores $S_{\text{Sam}}^{ij} = \exp(-\text{Norm}(\sum E_{\text{Sam}}^{ij}))$. We encode them into pair-wise embeddings Emb_{ij}, and sum across all other views $i \neq j$ to yield per-view embedding Emb_j. To construct the final input token for our EpiRef module, we concatenate Emb_j with $\tau_j^{(m)}$ and N_j^{vp}, where N_j^{vp} is the number of views sharing at least one valid (dynamic and visible) point pair with view j. Importantly, the final input token enables the module to recognize *which time offset candidate yields the best geometric consistency to minimize the Sampson error against other views.* Finally, the EpiRef module predicts time offset residual $\Delta\tau_j^{(m+1)}$ at m-th iteration by a view attention layer from the final input token.

3.3.2 Spatio-Temporal-Cross-View (STCV) Attention Network

We adopt the architectures of MV-TAP (Koo et al., 2026) and MVTracker (Rajič et al., 2025) as the backbones of 2D and 3D tracking, respectively. Fig. 3 illustrates the concept of our STCV Attention network that robustly handles unsynchronized inputs by newly adding two attention layers: Cross-View Temporal Attention Layer and Unidirectional 2D-to-3D Attention Layer.

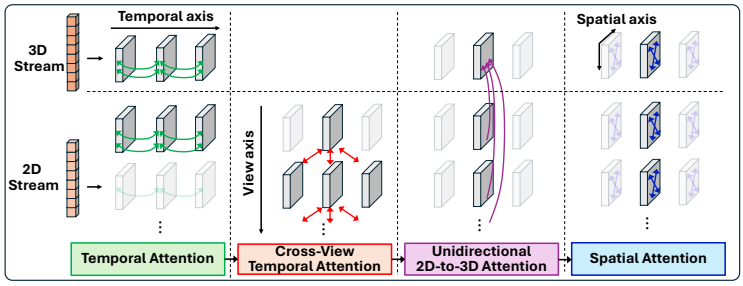

Figure 3: **Our STCV attention module.** It newly contains (i) a Cross-View Temporal Attention Layer that allows each 2D view embedding to attend to temporally nearby embeddings of other 2D views, and (ii) an Unidirectional 2D-to-3D Attention Layer allowing 3D stream to robustly aggregate 2D embeddings in latent space for high-fidelity tracking under unsynchronized multi-view inputs.

Cross-view Temporal Attention Layer (CVTAL). MV-TAP (Koo et al., 2026) applies a cross-view self-attention across corresponding frames. Given unsynchronized inputs, identical video frame indices no longer correspond to the same global timestamp. To address this temporal misalignment, we introduce CVTAL so that each 2D view can attend to a local temporal neighborhood ($\pm\rho$ frames) in other views under current $\tau_v^{(m)}$. This provides cross-view temporal tolerance, enabling the CVTAL to robustly assign higher attention weights to the most aligned frames while our EpiRef converges.

Unidirectional 2D-to-3D Attention Layer (U3DAL). Following MVTracker (Rajič et al., 2025), 3D input token aggregates per-view 2D features by computing feature correlation between 3D tracking point and K-nearest uplifted 2D pixels in the world space. This deterministic point-wise sampling relying on geometric unprojection easily fails under unsynchronized inputs. Instead, we introduce U3DAL so that 3D embedding can robustly capture multi-view 2D information in latent space. Notably, the attention layer is unidirectional (2D $\rightarrow$ 3D) to preserve the capability of 2D input tokens constructed from 4D feature correlation within each video regardless of time offsets.

3.4 Loss Functions

We jointly optimize the total loss $\mathcal{L}^j_{\text{total}}$ of j-th sliding window for time offset and 2D/3D trajectory:

$$\mathcal{L}^j_{\text{total}} = \sum_{m=1}^{M} \gamma^{M-m} \Big(\lambda^{\text{2D}}_{\text{coords}} \mathcal{L}^{\text{2D}}_{\text{coords}} + \lambda^{\text{2D}}_{\text{vis}} \mathcal{L}^{\text{2D}}_{\text{vis}} + \lambda^{\text{2D}}_{\text{conf}} \mathcal{L}^{\text{2D}}_{\text{conf}} + \lambda^{\text{3D}}_{\text{coords}} \mathcal{L}^{\text{3D}}_{\text{coords}} + \lambda^{\text{3D}}_{\text{vis}} \mathcal{L}^{\text{3D}}_{\text{vis}} + \lambda_{\text{offset}} \mathcal{L}_{\text{offset}} \Big), \tag{5}$$

where γ is an exponential decay factor for iterative updates and $\lambda^{\text{2D}}_*, \lambda^{\text{3D}}_*$ are loss weights. The 2D/3D trajectory losses ($\mathcal{L}^{\text{2D}}_{\text{coords}}, \mathcal{L}^{\text{2D}}_{\text{vis}}, \mathcal{L}^{\text{2D}}_{\text{conf}}$ / $\mathcal{L}^{\text{3D}}_{\text{coords}}, \mathcal{L}^{\text{3D}}_{\text{vis}}$) are the same as those of Koo et al. (2026) and Rajič et al. (2025). The time offset loss is defined as: $\mathcal{L}_{\text{offset}} = \sum_{v=1}^{V} |\tau_v^{(m)} - \tau_v^*|$ where $\tau_v^{(m)}$ is m-th predicted time offset and τ_v^* is GT time offset for view v. See Appendix B for details.

4 Experiments

Datasets. Recent multi-view 3D trackers rely on synthetic training data due to the need for ground-truth 3D trajectories. The MV-Kubric dataset (Rajič et al., 2025) consists of 5K videos of 24-frame length, each of which is too short to simulate unsynchronized inputs with diverse offset ranges. So, we newly generated a synthetic dataset of 5,265 scenes, named as **MV-Kubric2**, each of which consists of 10 views of 48-frame length following the MV-Kubric generation pipeline. Then we evaluate models on MV-Kubric2 test set. To prove generalizability, we also evaluate models on four real-world datasets: CMU Panoptic Studio dataset (Joo et al., 2015), DexYCB dataset (Chao et al., 2021), Harmony4D dataset (Khirodkar et al., 2024), and 3D-POP dataset (Naik et al., 2023). We simulate unsynchronized multi-view videos by sampling ground-truth per-view time offsets from the normal distribution $\mathcal{N}(0, \sigma_\tau^2)$ where the configuration for each dataset is described in Appendix C.1.

Table 1: **Multi-view synchronization performance** on MV-Kubric2, CMU Panoptic Studio, DexYCB, 3D-POP, and Harmony4D datasets. **Bold**: best, underline: second-best.

Method	MV-Kubric2					CMU Panoptic Studio					DexYCB				
	$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 ↑	A@500 ↑	$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 ↑	A@500 ↑	$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 ↑	A@500 ↑
GT offset	351.1	325.3	188.7	9.9	39.3	361.1	355.6	163.6	8.3	33.4	193.3	190.0	57.1	20.0	61.5
SyncNeRF (Kim et al., 2024)	386.4	345.8	408.0	5.4	37.2	350.3	344.9	445.3	11.6	33.8	255.0	229.1	303.8	15.4	51.5
MASt3R (Leroy et al., 2024)	157.0	147.6	79.2	40.0	74.7	479.6	511.1	209.7	14.4	42.0	233.3	230.0	112.5	18.3	62.0
VisualSync (Liu et al., 2025)	94.4	73.3	71.7	56.9	85.0	168.5	166.7	89.2	31.5	67.8	133.3	100.0	100.8	45.0	73.7
SyncTAP (Ours)	**30.2**	**25.4**	**18.2**	**73.3**	**94.1**	**29.4**	**27.5**	**18.3**	**70.9**	**94.0**	**44.7**	**32.4**	**33.3**	**70.4**	**91.4**

Method	3D-POP					Harmony4D				
	$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 ↑	A@500 ↑	$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 ↑	A@500 ↑
GT offset	483.3	475.0	163.8	6.9	25.1	262.0	227.8	150.7	22.0	47.3
MASt3R (Leroy et al., 2024)	463.9	491.7	219.4	5.6	31.4	392.6	363.9	181.6	14.4	43.0
VisualSync (Liu et al., 2025)	350.0	341.7	240.6	12.5	38.3	264.8	208.3	162.6	25.2	53.1
SyncTAP (Ours)	**37.2**	**38.9**	**27.4**	**68.2**	**93.2**	**135.6**	**125.7**	**90.3**	**49.3**	**73.6**

Table 2: **Multi-view 3D tracking performance** on MV-Kubric2, CMU Panoptic Studio, and DexYCB datasets.

Method	Sync.	MV-Kubric2				CMU Panoptic Studio				DexYCB			
		$\psi_{\text{aj}} \uparrow$	$\psi_{\text{pa}} \uparrow$	$\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$	$\psi_{\text{aj}} \uparrow$	$\psi_{\text{pa}} \uparrow$	$\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$	$\psi_{\text{aj}} \uparrow$	$\psi_{\text{pa}} \uparrow$	$\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$
MVTracker (Rajič et al., 2025)		51.6	70.6	76.8	20.4	56.9	84.1	67.6	7.8	67.4	82.7	80.6	3.22
MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	✓	50.8	69.6	77.3	20.3	53.0	81.6	66.2	7.6	66.4	82.2	79.0	3.37
VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	✓	52.8	71.1	78.4	19.8	51.7	80.6	65.0	7.0	67.2	83.4	79.7	2.98
SyncTAP (Ours)	✓	**62.0**	**73.6**	**90.2**	**15.8**	**67.7**	**86.5**	**85.4**	**4.4**	**74.4**	**85.3**	**89.6**	**2.95**

Evaluation metrics. We report per-view synchronization errors in milliseconds (*ms*) regardless of dataset-specific frame rates. **Synchronization performance** is measured in terms of **(i) Mean Absolute Error** δ_{mean}, **(ii) Median Absolute Error** δ_{median}, **(iii) Standard Deviation of Absolute Error** δ_{std}, and **(iv) Area Under the Curve (AUC)** with thresholds 100 ms (A@100) and 500 ms (A@500), excluding the reference view for a fair comparison since we set $\tau_1 = 0$. **2D/3D tracking performance** is measured in terms of **(i) Average Position Accuracy** ψ_{pa} as the Percentage of Correct Keypoints averaged over pre-defined thresholds, **(ii) Occlusion Accuracy** ψ_{oa} as the binary classification of visibility, and **(iii) Average Jaccard** ψ_{aj} as a joint evaluation of ψ_{pa} and ψ_{oa}. 3D tracking performance is also measured in terms of **(iv) Median Trajectory Error** δ_{mte} as the median spatial error in the unit of centimeters. More details about metrics are described in Appendix C.2.

Implementation details. We train our SyncTAP on MV-Kubric2 dataset for 30k iterations using two NVIDIA A6000 48GB GPUs (1 batch/GPU). Our 2D and 3D backbones are initialized from Koo et al. (2026) and Rajič et al. (2025), respectively. To process long videos, we use a sliding window of 12 frames with a stride of 8. Details about implementation details are in Appendix C.3.

Baselines. For multi-view synchronization, we compare our SyncTAP against frame-level estimation models (MASt3R (Leroy et al., 2024), VisualSync (Liu et al., 2025)) and a per-scene optimization method (SyncNeRF (Kim et al., 2024)). For multi-view point tracking, we compare our SyncTAP against (i) *single baselines* without synchronization (MV-TAP (Koo et al., 2026) for 2D, and MVTracker (Rajič et al., 2025) for 3D), (ii) *MASt3R-cascaded baselines* (MASt3R + MV-TAP/MVTracker), and (iii) *VisualSync-cascaded baselines* (VisualSync + MV-TAP/MVTracker). Details about baseline implementation are in Appendix C.4.

4.1 Quantitative evaluation

Table 1 shows the performance comparison for multi-view synchronization. As shown, our SyncTAP outperforms the recent SOTA visual synchronization methods in all the metrics for all the dataset *by significantly large margins*. While VisualSync (Liu et al., 2025) finds frame-level time offsets as pre-processing relying on external foundation models, our SyncTAP itself estimates sub-frame time offsets via our EpiInit, EpiRef, and STCV attention module. Table 2 and 9(in Appendix D) tabulate the 3D and 2D tracking performance results, respectively. As shown, our SyncTAP outperforms existing SOTA multi-view 3D/2D trackers robustly under unsynchronized videos.

4.2 QUALITATIVE EVALUATION

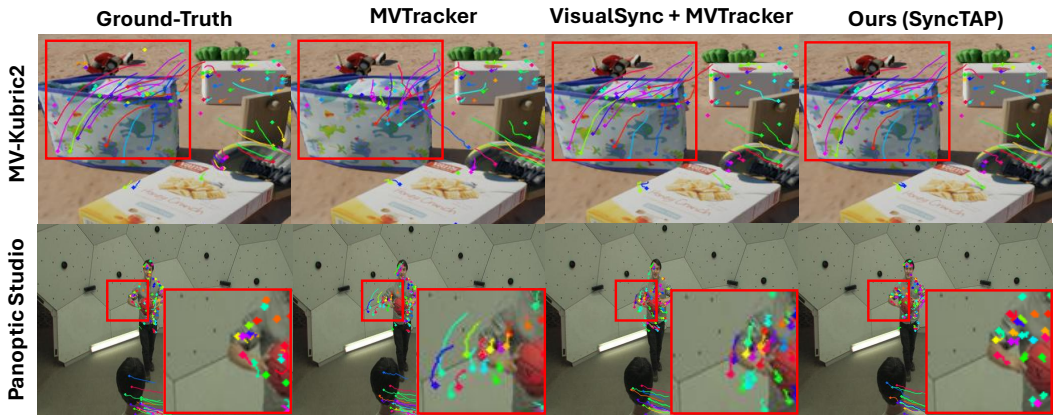

Figure 4: **Qualitative comparison of multi-view 3D point tracking for unsynchronized videos.** We visualize 3D trajectories by projecting them onto a novel viewpoint unseen in 4-view inference inputs. Our SyncTAP robustly estimates high-fidelity 3D trajectories for highly dynamic objects (1st row) and yields no ghosting artifacts of floating 3D trajectories (2nd row).

Fig. 4 shows a qualitative evaluation of 3D trajectories on the MV-Kubric2 and Panoptic Studio datasets by projecting 3D trajectories onto a novel viewpoint. High-fidelity projections onto a novel viewpoint demonstrate their strong multi-view consistency. MVTracker without synchronization shows degraded results: (i) for highly dynamic objects (e.g., falling blue basket in the 1st row), small time offsets result in larger spatial discrepancies, which leads to 3D tracking failure by aggregating spatially inconsistent 2D features; and (ii) ghosting artifacts occur (e.g., floating trajectories in the 2nd row) since 3D trajectory estimation at a specific frame is disrupted by 2D features from different timestamps, hallucinating the presence of an object in empty regions. 'VisualSync + MVTracker' may also suffer from inaccurate time offset estimation due to (i) preprocessing from external models and (ii) its frame-level accuracy. In contrast, our SyncTAP successfully estimates sub-frame time offsets and high-fidelity 3D trajectories even for highly dynamic objects without ghosting artifacts.

4.3 ABLATION STUDIES AND ANALYSIS

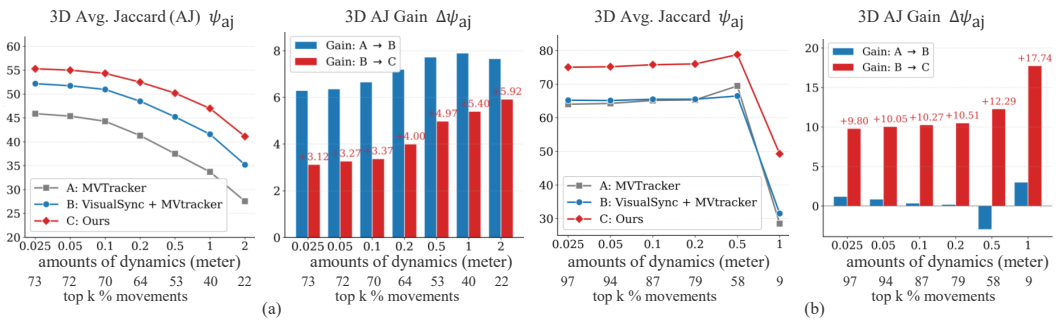

Figure 5: **3D tracking performance in average jaccard (ψ_{aj}) for dynamic 3D trajectories.** ψ_{aj} performance solely on dynamic trajectories is measured for various amounts of trajectory movements for (a) MV-Kubric2 and (b) Panoptic Studio datasets.

Importance of synchronization especially on dynamic trajectories. Fig. 5 shows the 3D tracking performance (ψ_{aj}) on only dynamic 3D trajectories across various motion thresholds. Our method becomes much more effective for highly dynamic cases than VisualSync since our approach achieves precise sub-frame synchronization. Importantly, fast-moving objects are highly vulnerable to synchronization errors since even a slight temporal misalignment leads to severe 3D spatial deviations.

Table 3: **Ablation of pipeline components on MV-Kubric2 dataset.**

Method	Modules				3D Tracking				Synchronization				
	EpiInit	EpiRef	CVTAL	U3DAL	3D $\psi_{aj}\uparrow$	3D $\psi_{pa}\uparrow$	3D $\psi_{oa}\uparrow$	3D $\delta_{mte}\downarrow$	$\delta_{mean}\downarrow$	$\delta_{median}\downarrow$	$\delta_{std}\downarrow$	A@100 $\uparrow$	A@500 $\uparrow$
Baseline (MV-TAP + MVTracker)					55.7	67.8	81.4	20.7	351.1	325.3	188.7	9.9	39.3
+ EpiRef		✓			61.9	72.8	87.4	15.2	93.8	79.0	62.9	46.6	81.2
+ EpiInit	✓	✓			65.1	75.3	88.8	14.5	31.6	27.7	18.4	72.4	93.9
+ CVTAL	✓	✓	✓		66.4	76.1	89.7	13.3	30.7	28.3	**17.6**	**73.3**	**94.4**
+ U3DAL (**SyncTAP (Ours, Full)**)	✓	✓	✓	✓	**67.9**	**76.9**	**91.9**	**12.1**	**30.2**	**25.4**	18.2	73.3	94.1

Table 4: **Robustness for varying time offset amounts σ_τ on MV-Kubric2 dataset.**

σ_τ (frames)	Tracking				Synchronization				
	3D $\psi_{aj}\uparrow$	3D $\psi_{pa}\uparrow$	3D $\psi_{oa}\uparrow$	$\delta_{mte}\downarrow$	$\delta_{mean}\downarrow$	$\delta_{median}\downarrow$	$\delta_{std}\downarrow$	A@100 $\uparrow$	A@500 $\uparrow$
0 (Sync.)	**69.8**	**79.4**	**92.7**	**10.5**	**16.9**	**16.2**	**6.5**	**85.7**	**97.1**
2	65.7	77.0	90.6	12.1	21.1	18.9	10.9	81.4	96.1
5	62.0	73.6	90.2	15.8	30.2	25.4	18.2	73.3	94.1
10	58.4	70.8	89.7	18.2	53.9	37.2	44.4	64.1	89.6
15	57.4	69.9	89.3	20.2	72.0	54.4	54.2	58.7	87.1

Table 5: **Robustness for sub-frame time offsets on MV-Kubric2 and Panoptic Studio datasets.**

Method	$S=1$ (orig. fps)					$S=2$ (orig. fps / 2)					$S=3$ (orig. fps / 3)				
	$\delta_{mean}\downarrow$	$\delta_{median}\downarrow$	$\delta_{std}\downarrow$	A@100 $\uparrow$	A@500 $\uparrow$	$\delta_{mean}\downarrow$	$\delta_{median}\downarrow$	$\delta_{std}\downarrow$	A@100 $\uparrow$	A@500 $\uparrow$	$\delta_{mean}\downarrow$	$\delta_{median}\downarrow$	$\delta_{std}\downarrow$	A@100 $\uparrow$	A@500 $\uparrow$
MV-Kubric2															
No Sync	351.1	325.3	188.7	9.9	39.3	351.1	325.3	188.7	9.9	39.3	351.1	325.3	188.7	9.9	39.3
MASt3R (Leroy et al., 2024)	157.0	147.6	79.2	40.0	74.7	336.3	298.2	164.3	14.6	46.7	300.9	249.0	184.8	17.5	53.6
VisualSync (Liu et al., 2025)	94.4	73.3	71.7	56.9	85.0	134.2	101.4	101.3	37.7	78.1	258.7	215.9	141.6	19.3	57.4
SyncTAP (Ours)	**30.2**	**25.4**	**18.2**	**73.3**	**94.1**	**53.8**	**40.0**	**36.7**	**55.5**	**89.2**	**83.3**	**77.3**	**48.9**	**40.5**	**83.6**
Panoptic Studio															
No Sync	361.1	355.6	163.6	8.3	33.4	361.1	355.6	163.6	8.3	33.4	361.1	355.6	163.6	8.3	33.4
MASt3R (Leroy et al., 2024)	479.6	511.1	209.7	14.4	42.0	627.8	600.0	210.8	4.6	29.4	498.1	494.4	267.1	3.7	22.4
VisualSync (Liu et al., 2025)	168.5	166.7	89.2	31.5	67.8	609.3	527.8	222.9	29.2	49.5	468.5	338.9	257.2	5.6	33.4
SyncTAP (Ours)	**29.4**	**27.5**	**18.3**	**70.9**	**94.0**	**21.2**	**14.8**	**11.4**	**80.0**	**96.0**	**20.8**	**20.7**	**11.2**	**79.9**	**96.0**

Ablation of pipeline components. Table 3 shows progressive 3D tracking improvements when each component is added. Meanwhile, adding EpiRef reduces δ_{mean} by 73.3%, showing that network learns synchronization from epipolar constraints. Adding EpiInit reduces δ_{mean} by 66.3%, showing that epipolar constraints with TSA provides accurate offset initialization. Adding CVTAL shows mutual benefits of tracking and synchronization under temporal tolerance. Adding U3DAL improves only 3D tracking without affecting synchronization since it updates 3D stream in unidirectional way.

Robustness for varying time offset amounts. Table 4 compares 3D tracking and synchronization performance across various time offset standard deviations σ_τ. Our SyncTAP is trained only with $\sigma_\tau = 5$, but robustly achieves sub-frame synchronization for varying σ_τ, maintaining $\delta_{\text{mean}} \leq 72ms$ (< 1 frames) on 12 fps MV-Kubric2. Smaller σ_τ improves 3D tracking since more temporal overlap across views contains richer multi-view context. Our SyncTAP also correctly estimates $\tau_v^M \approx 0$ even for perfectly synchronized videos($\sigma_\tau = 0$), ensuring robust tracking for both synchronized and unsynchronized multi-view videos. The baseline performances for various σ_τ are in Appendix D.6.

Sub-frame vs. Frame-level time offset estimation. Table 5 compares synchronization performance in sub-frame time offset scenarios for low-fps input videos. We downsample input frames with stride $S \in \{2, 3\}$ so that a 1-frame offset sampled at the original frame rate effectively becomes a 0.33-frame offset at $S = 3$. VisualSync degrades severely when $S > 1$ due to its frame-level alignment. In contrast, our SyncTAP captures sub-frame time offsets, demonstrating its applicability for real-world videos where temporal misalignments occur in a continuous physical time axis.

5 Conclusion

Our SyncTAP is a novel unified framework that deals with the challenging problem of multi-view 3D point tracking under unsynchronized multi-view videos. Our approach analytically initializes sub-frame time offsets via EpiInit, and iteratively refines both time offsets and 2D/3D trajectories via EpiRef and STCV attention. Extensive experiments demonstrate that our method achieves remarkable sub-frame synchronization and tracking accuracy, outperforming baselines by a large margin.

AI USE STATEMENT

In this work, we used generative AI tools to polish writing such as grammar check and sentence completion. We have not used generative AI tools for other tasks including research ideas, synthetic dataset generation, experiments, and analysis. We take responsibility for the final content of our work.

REPRODUCIBILITY STATEMENT

We show the reproducibility of our work in various ways. In Sec. 4 and Appendix C, we provide detailed explanations for reproducibility, including dataset descriptions, evaluation metric descriptions, and implementation details for training and evaluation. Specifically, for datasets, we utilized several public evaluation datasets, and our synthetic training dataset was also constructed by the public synthetic data generation pipeline, where we also explain how we generate the synthetic dataset MV-Kubric2 and what the dataset consists of in Sec. 4 and Appendix C.1. In Appendix A, we show the derivation of equations and the pseudo-code of convergence algorithm used in our work. Beyond the explanation, we plan to release the code of our method and the MV-Kubric2 dataset with its generation pipeline after acceptance for other researchers to reproduce our experiments. As mentioned in Appendix C.3, we fix the evaluation protocol, such as the number of views and queries, and its random seed across methods to ensure reproducibility and fair comparison. Furthermore, in Appendix D.9, we provide multiple random seed statistics on the synchronization performance and the 3D tracking performance to prove the robustness of our method and to provide confidence intervals for the reproducibility by other researchers.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Cenek Albl, Zuzana Kukelova, Andrew Fitzgibbon, Jan Heller, Matej Smid, and Tomas Pajdla. On the two-view geometry of unsynchronized cameras. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5593–5602. IEEE, 2017.

Prabir Barooah and Joao P Hespanha. Estimation from relative measurements: Electrical analogy and large graphs. *IEEE Transactions on Signal Processing*, 56(6):2181–2193, 2008.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9040–9049. IEEE, 2021.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1316–1326. IEEE, 2023.

Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. In *European conference on computer vision*, pp. 306–325. Springer, 2024.

Changwoon Choi, Jeongjun Kim, Geonho Cha, Minkwan Kim, Dongyoon Wee, and Young Min Kim. Humans as a calibration pattern: Dynamic 3d scene reconstruction from unsynchronized and uncalibrated videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6598–6608. IEEE, 2025.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022.

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 *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10027–10038. IEEE, 2023.

Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, Joao Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Asian Conference on Computer Vision*, pp. 483–500. Springer, 2024.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19790–19800. IEEE, 2024.

Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8503–8513. IEEE, 2025.

Bishoy Galoaa, Xiangyu Bai, Shayda Moezzi, Utsav Nandi, Sai Siddhartha Vivek Dhir Rangoju, Somaieh Amraee, and Sarah Ostadabbas. Look around and pay attention: Multi-camera point tracking reimagined with transformers. In *2026 International Conference on 3D Vision (3DV)*, pp. 1–17. IEEE, 2026.

Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pp. 59–75. Springer, 2022.

Adam W Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Suya You, et al. Alltracker: Efficient dense point tracking at high resolution. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5253–5262. IEEE, 2025.

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge university press Cambridge, 2003.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. Synchformer: Efficient synchronization from sparse cues. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5325–5329. IEEE, 2024.

Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, pp. 3334–3342, 2015.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European conference on computer vision*, pp. 18–35. Springer, 2024.

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.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. In *2026 International Conference on 3D Vision (3DV)*, pp. 499–509. IEEE, 2026.

Rawal Khirodkar, Jyun-Ting Song, Jinkun Cao, Zhengyi Luo, and Kris Kitani. Harmony4d: A video dataset for in-the-wild close human interactions. *Advances in Neural Information Processing Systems*, 37:107270–107285, 2024.

Seoha Kim, Jeongmin Bae, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Sync-nerf: Generalizing dynamic nerfs to unsynchronized videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2777–2785, 2024.

Jahyeok Koo, Inès Hyeonsu Kim, Mungyeom Kim, Junghyun Park, Seohyeon Park, Jaeyeong Kim, Jung Yi, Seokju Cho, and Seungryong Kim. Mv-tap: Tracking any point in multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20932–20941, 2026.

Yonghan Lee, Tsung-Wei Huang, Shiv Gehlot, Jaehoon Choi, Guan-Ming Su, and Dinesh Manocha. Synctrack4d: Cross-video motion alignment and video synchronization with multi-video 4d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 77–87, 2026.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

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. *arXiv preprint arXiv:2511.10647*, 2025.

Shaowei Liu, David Yao, Saurabh Gupta, and Shenlong Wang. Visual sync: Multi-camera synchronization via cross-view object motion. In *Advances in Neural Information Processing Systems*, volume 38, pp. 22759–22784. Curran Associates, Inc., 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pp. 800–809. IEEE, 2024.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

Hemal Naik, Alex Hoi Hang Chan, Junran Yang, Mathilde Delacoux, Iain D Couzin, Fumihiro Kano, and Máté Nagy. 3d-pop-an automated annotation approach to facilitate markerless 2d-3d tracking of freely moving birds with marker-based motion capture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21274–21284. IEEE, 2023.

Tuan Ngo, Peiye Zhuang, Evangelos Kalogerakis, Chuang Gan, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. Delta: Dense efficient long-range 3d tracking for any video. In *International Conference on Learning Representations*, volume 2025, pp. 44620–44641, 2025.

Frano Rajič, Haofei Xu, Marko Mihajlovic, Siyuan Li, Irem Demir, Emircan Gündoğdu, Lei Ke, Sergey Prokudin, Marc Pollefeys, and Siyu Tang. Multi-view 3d point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 59–68, 2025.

Rao, Gritai, and Shah. View-invariant alignment and matching of video sequences. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 939–945. IEEE, 2003.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4937–4946. IEEE, 2020.

Jenny Seidenschwarz, Qunjie Zhou, Bardienus P Duisterhof, Deva Ramanan, and Laura Leal-Taixé. Dynomo: Online point tracking by dynamic online monocular gaussian reconstruction. In *2025 International Conference on 3D Vision (3DV)*, pp. 1012–1021. IEEE, 2025.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *arXiv preprint arXiv:2104.00680*, 2021.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9660–9672. IEEE, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20406–20417. IEEE, 2024.

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, 35:38571–38584, 2022.

Zhixin Xu, Hengyu Zhou, Yuan Liu, Wenhan Xue, Hao Pan, Wenping Wang, and Bin Wang. Dynamic gaussian scene reconstruction from unsynchronized videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11469–11477, 2026.

Bowei Zhang, Lei Ke, Adam Harley, and Katerina Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. In *Advances in Neural Information Processing Systems*, volume 38, pp. 135284–135303. Curran Associates, Inc., 2025.

Chengwei Zheng, Lixin Xue, Juan Zarate, and Jie Song. Gaustar: Gaussian surface tracking and reconstruction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16543–16553. IEEE, 2025.

Artem Zholus, Carl Doersch, Yi Yang, Skanda Koppula, Viorica Patraucean, Xu Owen He, Ignacio Rocco, Mehdi SM Sajjadi, Sarath Chandar, and Ross Goroshin. Tapnext: Tracking any point (tap) as next token prediction. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9693–9703. IEEE, 2025.

APPENDIX

In this appendix, we first introduce the full mathematical derivations for our EpiInit method along with the pseudo-code and experiments for our EpiInit convergence algorithm in Sec. A. We explain training loss functions in Sec. B. Then, we provide detailed descriptions of experiment settings: datasets, evaluation metrics, implementation details, and baselines in Sec. C. Furthermore, we demonstrate quantitative ablation studies to compare our model's performance in Sec. D, and present qualitative results in Sec. E. Finally, we address the limitations of our approach in Sec. F.

A EQUATION DERIVATION AND ALGORITHM

A.1 MATHEMATICAL DERIVATION OF EQUATIONS IN EPIINIT

While our Sampson (epipolar constraint) formulation can theoretically support moving cameras, in this work, we focus on a multi-view stationary camera setup of a dynamic scene to align with the standard configuration of widely used multi-view 2D/3D tracking benchmark.

Pairwise-fitting via Epipolar constraint and Taylor Series Approximation (TSA). In Pairwise-fitting step of Sec. 3.2, for valid pairs of visible and dynamic 2D trajectory points $\{(\mathbf{x}_i^{n,t}, \mathbf{x}_j^{n,t})\}_{n=1,t=1}^{N,T}$ between view i and j, we apply the 2nd-order Taylor series approximation to the epipolar constraint as:

$$\left(\mathbf{x}_j^{n,t} + \mathbf{v}_j^{n,t}\, T_{ij}^{n,t} + \tfrac{1}{2}\,\mathbf{a}_j^{n,t}\,(T_{ij}^{n,t})^2\right)^\top \mathbf{F}_{ij}\,\mathbf{x}_i^{n,t} \approx 0. \tag{6}$$

Then, we organize Eq. 6 as:

$$A\,(T_{ij}^{n,t})^2 + B\,T_{ij}^{n,t} + C \;=\; 0, \quad \text{where} \quad A = \tfrac{1}{2}\,a_j^\top F_{ij}\,x_i, \quad B = v_j^\top F_{ij}\,x_i, \quad C = x_j^\top F_{ij}\,x_i. \tag{7}$$

The solution of Eq. 7 obtained by the naive closed-form root $T_{ij}^{n,t} = (-B \pm \sqrt{B^2 - 4AC})/(2A)$ is unstable when a dynamic point moves steadily ($a_j \approx 0$ makes $A \approx 0$). We instead use the conjugate form as:

$$T_{ij}^{n,t} \;=\; \frac{-2C}{B \;+\; \mathrm{sign}(B)\sqrt{B^2 - 4AC}}. \tag{8}$$

and fall back to the 1st-order TSA estimate $T_{ij}^{n,t} = -C/B$ if $B^2 - 4AC < 0$ due to tracker noise.

Global-fitting via Laplacian Solver. In Global-fitting step of Sec. 3.2, given view-pairwise time offsets $\{T_{ij}\}$ and nonnegative confidences $\{W_{ij}\}$, we obtain the initial *per-view* time offsets $\{\Delta\tau_v^0\}_{v=1}^V$ by minimizing a weighted energy function for all view pairs as:

$$E(\{\Delta\tau_v^0\}_{v=1}^V) \;=\; \sum\nolimits_{j=1}^{V} \sum\nolimits_{i \neq j} W_{ij}\left(\Delta\tau_i^0 - \Delta\tau_j^0 - T_{ij}\right)^2, \tag{9}$$

If we aggregate equations $\partial E(\{\Delta\tau_v^0\}_{v=1}^V)/\partial\Delta\tau_v^0 \;=\; 0$ for $v \;=\; 1, ..., V$ to find the optimal $\{\Delta\tau_v^0\}_{v=1}^V$ which minimizes $E(\{\Delta\tau_v^0\}_{v=1}^V)$, we can formulate matrix equation as:

$$\begin{aligned} \underbrace{\sum_{m \neq k} (W_{km} + W_{mk})}_{L_{kk}}\,\Delta\tau_k^0 \;+\; \sum_{m \neq k} \underbrace{\left[-(W_{km} + W_{mk})\right]}_{L_{km}}\,\Delta\tau_m^0 \\ = \underbrace{\sum_{m \neq k} \left(W_{km}T_{km} - W_{mk}T_{mk}\right)}_{b_k}. \end{aligned} \tag{10}$$

That is,

$$\mathbf{L}\boldsymbol{\Delta\tau} = \mathbf{b}, \tag{11}$$

where $\mathbf{L} \in \mathbb{R}^{V \times V}$ is the symmetric Graph Laplacian whose edge weights are pairwise confidence $W_{ij} + W_{ji}$. Because the energy is quadratic in $\Delta\tau_v^0$ where $2\mathbf{L}$ is the Hessian of $E(\{\Delta\tau_v^0\}_{v=1}^V)$, we can obtain a closed-form solution of the system in Eq. 11 without exhaustive grid search. Note that, since a global time shift of all views with the same amount is not observable from pairwise differences, $\mathbf{L}$ itself has an one-dimensional null space. We resolve this ambiguity by applying Tikhonov regularization and shifting the estimated per-view time offsets such that the first view (Camera 1) is set as the reference view with $\Delta\tau_1^0 = 0$.

A.2 Experiments on EpiInit Iteration and Convergence Algorithm

To prevent unnecessary computation while ensuring optimal initialization, our EpiInit stage iterates the Pairwise-fitting and the Global-fitting steps until the residuals of time offsets converge with one of the three criteria: (a) if the average change in time offsets is small at the beginning for early stopping for synchronized inputs, (b) if the average change in time offsets is kept small for several times, or (c) if the average change in time offsets is small and descends over the previous estimates. The detailed pseudocode is provided in Algorithm 1.

Algorithm 1 Iterative EpiInit with Robust Convergence

Input: visible, dynamic 2D trajectory point pairs $\{(\mathbf{x}_i^{n,t}, \mathbf{x}_j^{n,t})\}_{n=1,t=1}^{N,T}$,
fundamental matrices $\{F_{ij}\}$,
maximum iteration $N_{\max}$, time offset change thresholds $\epsilon_{\text{abs}}, \epsilon_{\text{first}}$,
descent ratio $\rho \in (0,1)$, persistence count threshold K_{cons}
Output: Initialized per-view time offsets $\{\tau_v^0\}$
The number of EpiInit iterations r

$\tau_v^{0,(0)} \leftarrow 0$ for all $v \in \{1, \dots, V\}$ $\quad$ $\{\tau_v^{0,(r)}$: time offset of view v at r-th EpiInit iteration$\}$
$r \leftarrow 0, \quad c \leftarrow 0, \quad \bar{\delta}_{\text{prev}} \leftarrow \texttt{None}$
while $r < N_{\max}$ **do**
  $r \leftarrow r + 1$
  $\{T_{ij}\} \leftarrow$ **Pairwise-fitting**$\big(\{(\mathbf{x}_i^{n,t}, \mathbf{x}_j^{n,t})\}, \{F_{ij}\}, \{\tau_v^{0,(r-1)}\}\big)$
  $\{\Delta\tau_v^{0,(r)}\} \leftarrow$ **Global-fitting**$\big(\{T_{ij}\}\big)$
  $\tau_v^{0,(r)} \leftarrow \tau_v^{0,(r-1)} + \Delta\tau_v^{0,(r)}$ for all v
  $\bar{\delta} \leftarrow \frac{1}{V}\sum_v |\Delta\tau_v^{0,(r)}|$
  if $r \geq 2$ **then**
    $\texttt{small} \leftarrow \big(\bar{\delta} < \epsilon_{\text{abs}}\big)$
    $\texttt{descend} \leftarrow \big(\bar{\delta}_{\text{prev}} \neq \texttt{None}\big) \wedge \big(\bar{\delta} < \rho \cdot \bar{\delta}_{\text{prev}}\big)$
    $c \leftarrow c + 1$ **if** $\texttt{small}$ **else** 0
    $\mathcal{C}_a \leftarrow (r = 2) \wedge (\bar{\delta} < \epsilon_{\text{first}})$ $\quad$ $\{$(a) already sync.$\}$
    $\mathcal{C}_b \leftarrow (c \geq K_{\text{cons}})$ $\quad$ $\{$(b) keep stably small $\bar{\delta}\}$
    $\mathcal{C}_c \leftarrow \texttt{small} \wedge \texttt{descend}$ $\quad$ $\{$(c) small $\bar{\delta}$ & descending $\bar{\delta}\}$
    if $\mathcal{C}_a \vee \mathcal{C}_b \vee \mathcal{C}_c$ **then**
      break
    end if
  end if
  $\bar{\delta}_{\text{prev}} \leftarrow \bar{\delta}$
end while
return (i) $\{\tau_v^{0,(r)}\}$ as $\{\tau_v^0\}$ and (ii) r

Tables 6 and 7 show the effect of the number of iterative steps (N_{iter}) of Pairwise-fitting step and Global-fitting step in our EpiInit and validate our convergence-based stopping criteria by comparing the 3D tracking and synchronization performance among the cases of fixed iteration ($N_{\text{iter}} \in \{1..5\}$) and our adaptive convergence algorithm.

Early saturation on MV-Kubric2 dataset with $\sigma_\tau = 5$. Table 6 compares 3D tracking and synchronization performance over the number of iterations of our EpiInit on our MV-Kubric2 dataset. Both tracking and synchronization performance improve when we iterate EpiInit more, and nearly saturate at $N_{\text{iter}} = 4 \sim 5$. Our convergence-based experiment repeats EpiInit for $N_{\text{iter}} \sim 4.2$ iterations on average, which proves that the adaptive stopping criteria in Algorithm 1 correctly identifies the saturation of performance and saves unnecessary computation.

More iterations on CMU Panoptic Studio dataset with $\sigma_\tau = 12.5$. Table 7 compares 3D tracking and synchronization performance over the number of EpiInit iterations on CMU Panoptic Studio dataset. To simulate the same synchronization error in physical time axis in the unit of [ms], we sampled ground truth time offsets from higher standard deviation σ_τ in the unit of [frame] for higher fps input videos. For harder cases with these larger ground-truth time offsets, we need to iterate more to obtain satisfiable performance. That is, a fixed iteration strategy has a bottleneck in performance

Table 6: **Ablation studies across the number of EpiInit iterations on MV-Kubric2 dataset.**

Method	3D Tracking				Synchronization				
	3D $\psi_{aj}\uparrow$	3D $\psi_{pa}\uparrow$	3D $\psi_{oa}\uparrow$	$\delta_{mte}\downarrow$	$\delta_{mean}\downarrow$	$\delta_{median}\downarrow$	$\delta_{std}\downarrow$	A@100 $\uparrow$	A@500 $\uparrow$
No Sync. + MVTracker (Rajič et al., 2025)	51.6	70.6	76.8	20.4	351.1	325.3	188.7	9.9	39.3
MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	50.8	69.6	77.3	20.3	157.0	147.6	79.2	40.0	74.7
VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	52.8	71.1	78.4	19.8	94.4	73.3	71.7	56.9	85.0
Ours ($N_{iter} = 1$)	52.6	69.9	80.5	20.7	283.8	262.1	155.0	13.3	47.7
Ours ($N_{iter} = 2$)	60.0	73.4	87.7	16.0	92.7	72.6	65.0	48.9	80.8
Ours ($N_{iter} = 3$)	61.8	**73.9**	89.6	**14.9**	45.2	36.9	31.4	68.5	90.9
Ours ($N_{iter} = 4$)	61.7	73.6	89.9	15.7	38.1	31.9	21.6	71.5	93.1
Ours ($N_{iter} = 5$)	62.0	73.7	90.2	15.6	33.0	28.1	18.3	**73.6**	94.0
Ours (with converge algorithm)	**62.0**	73.6	**90.2**	15.8	**30.2**	**25.4**	**18.2**	73.3	**94.1**

Table 7: **Ablation studies across the number of EpiInit iterations on CMU Panoptic Studio dataset.**

Method	3D Tracking				Synchronization				
	3D $\psi_{aj}\uparrow$	3D $\psi_{pa}\uparrow$	3D $\psi_{oa}\uparrow$	$\delta_{mte}\downarrow$	$\delta_{mean}\downarrow$	$\delta_{median}\downarrow$	$\delta_{std}\downarrow$	A@100 $\uparrow$	A@500 $\uparrow$
No Sync. + MVTracker (Rajič et al., 2025)	56.9	84.1	67.6	7.8	361.1	355.6	163.6	8.3	33.4
MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	53.0	81.6	66.2	7.6	479.6	511.1	209.7	14.4	42.0
VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	51.7	80.6	65.0	7.0	168.5	166.7	89.2	31.5	67.8
Ours ($N_{iter} = 1$)	61.3	83.1	73.5	8.0	322.6	314.7	147.0	12.8	38.0
Ours ($N_{iter} = 2$)	63.9	**86.9**	77.3	5.4	212.8	230.3	95.2	17.9	56.7
Ours ($N_{iter} = 3$)	64.3	86.5	81.1	4.6	112.7	120.7	31.8	36.3	76.4
Ours ($N_{iter} = 4$)	65.0	85.9	82.6	4.6	83.9	72.9	35.9	51.7	81.8
Ours ($N_{iter} = 5$)	66.3	86.8	83.0	4.4	60.6	65.5	42.6	56.7	87.9
Ours (with converge algorithm)	**67.7**	86.5	**85.4**	**4.4**	**29.4**	**27.5**	**18.3**	**70.9**	**94.0**

since it cannot deal with diverse σ_τ, and needs a manual hyperparameter decision for each scene to obtain accurate performance. In contrast, our convergence-based strategy repeats EpiInit for $N_{\text{iter}} \sim 6.5$ iterations on average and it proves that the stopping criterion also works well on higher fps input videos.

These results validate our adaptive convergence algorithm across diverse difficulties. It saves computation by early stopping on the datasets of small σ_τ, and it repeats more iterations to achieve high performance on the challenging datasets of big σ_τ.

B LOSS

As explained in Sec. 3.1, at each iteration m, the EpiRef module predicts the residual of time offsets $\Delta\tau_v^{(m+1)}$, and STCV modules predict the residuals of 2D trajectory coordinates $\Delta\mathbf{T}_{2D}^{(m+1)}$, 2D trajectory visibilities $\Delta\mathbf{V}_{2D}^{(m+1)}$, 2D trajectory confidence $\Delta\mathbf{C}_{2D}^{(m+1)}$, 3D trajectory coordinates $\Delta\mathbf{T}_{3D}^{(m+1)}$, and 3D trajectory features $\Delta\mathbf{f}_{3D}^{(m+1)}$. And we iteratively update the estimates as follows:

$$\begin{aligned}
\tau_v^{(m+1)} &= \tau_v^{(m)} + \Delta\tau_v^{(m+1)}, \\
\mathbf{T}_{2D}^{(m+1)} &= \mathbf{T}_{2D}^{(m)} + \Delta\mathbf{T}_{2D}^{(m+1)}, \\
\mathbf{V}_{2D}^{(m+1)} &= \mathbf{V}_{2D}^{(m)} + \Delta\mathbf{V}_{2D}^{(m+1)}, \\
\mathbf{C}_{2D}^{(m+1)} &= \mathbf{C}_{2D}^{(m)} + \Delta\mathbf{C}_{2D}^{(m+1)}, \\
\mathbf{T}_{3D}^{(m+1)} &= \mathbf{T}_{3D}^{(m)} + \Delta\mathbf{T}_{3D}^{(m+1)}, \\
\mathbf{f}_{3D}^{(m+1)} &= \mathbf{f}_{3D}^{(m)} + \Delta\mathbf{f}_{3D}^{(m+1)},
\end{aligned} \tag{12}$$

Then, at the final iteration $m = M - 1$, $\mathbf{f}_{3D}^{(M)}$ is used to estimate 3D visibility $\mathbf{V}_{3D}^{(M)} = \sigma(W\mathbf{f}_{3D}^{(M)})$, where $\sigma(\cdot)$ is the sigmoid function and W is a learned projection matrix.

As explained in Sec. 3.4, the total loss of j-th sliding window is a weighted sum over all M iterations with an exponential decay factor γ as:

$$\mathcal{L}^j_{\text{total}} = \sum_{m=1}^{M} \gamma^{M-m} \Big(\lambda^{\text{2D}}_{\text{coords}} \mathcal{L}^{\text{2D}}_{\text{coords}} + \lambda^{\text{2D}}_{\text{vis}} \mathcal{L}^{\text{2D}}_{\text{vis}} + \lambda^{\text{2D}}_{\text{conf}} \mathcal{L}^{\text{2D}}_{\text{conf}} + \lambda^{\text{3D}}_{\text{coords}} \mathcal{L}^{\text{3D}}_{\text{coords}} + \lambda^{\text{3D}}_{\text{vis}} \mathcal{L}^{\text{3D}}_{\text{vis}} + \lambda_{\text{offset}} \mathcal{L}_{\text{offset}} \Big), \tag{13}$$

where exponential decay factor $\gamma = 0.8$ and loss weights $\lambda^{\text{2D}}_{\text{coords}} = 0.02$, $\lambda^{\text{2D}}_{\text{vis}} = 0.5$, $\lambda^{\text{2D}}_{\text{conf}} = 1.0$, $\lambda^{\text{3D}}_{\text{coords}} = 8.0$, $\lambda^{\text{3D}}_{\text{vis}} = 1.0$, and $\lambda_{\text{offset}} = 1.5$ are empirically found by hyperparameter search.

Then we sum up the total loss $\mathcal{L}^j_{\text{total}}$ of j-th sliding window over J sliding windows to yield the final loss as: $\mathcal{L}_{\text{final}} = \sum_{j=1}^{J} \mathcal{L}^j_{\text{total}}$.

To optimize both 2D trajectory regression and occlusion prediction at m-th iteration, we construct the losses as the same with MV-TAP (Koo et al., 2026):

$$\begin{aligned} \mathcal{L}^{\text{2D}}_{\text{coords}}(\mathbf{T}^{(m)}_{2D}, \mathbf{T}^{*}_{2D}) &= \ell_{\text{Huber}}\big(\mathbf{T}^{(m)}_{2D}, \mathbf{T}^{*}_{2D}\big), \\ \mathcal{L}^{\text{2D}}_{\text{vis}}(\mathbf{V}^{(m)}_{2D}, \mathbf{V}^{*}_{2D}) &= \ell_{\text{balanced-BCE}}\big(\mathbf{V}^{(m)}_{2D}, \mathbf{V}^{*}_{2D}\big), \\ \mathcal{L}^{\text{2D}}_{\text{conf}}(\mathbf{C}^{(m)}_{2D}, \mathbf{C}^{*}_{2D}) &= \ell_{\text{BCE}}\big(\mathbf{C}^{(m)}_{2D}, \mathbf{C}^{*}_{2D}\big), \end{aligned} \tag{14}$$

where $\mathbf{T}^{*}_{2D}$ is the GT 2D trajectory coordinates, $\mathbf{V}^{*}_{2D}$ is the binary GT 2D visibility, $\mathbf{C}^{*}_{2D} = \mathbb{I}\left(\|\mathbf{T}^{(m)}_{2D} - \mathbf{T}^{*}_{2D}\|_2 < \tau\right)$ is the binary GT 2D confidence of whether m-th predicted 2D trajectory coordinates are sufficiently close to the GT 2D trajectory coordinates within a threshold τ. We use a Huber loss (Huber, 1992) for 2D trajectory coordinates in pixel coordinates, which is robust to outliers in the initial estimates. We use a balanced Binary Cross Entropy loss for 2D trajectory visibility, which is robust to class imbalance. We use a Binary Cross Entropy loss for 2D trajectory confidence.

To optimize both 3D trajectory regression at m-th iteration and occlusion prediction at the final prediction, we construct the losses as the same with MVTracker (Rajič et al., 2025):

$$\begin{aligned} \mathcal{L}^{\text{3D}}_{\text{coords}}(\mathbf{T}^{(m)}_{3D}, \mathbf{T}^{*}_{3D}) &= \ell_1\big(\mathbf{T}^{(m)}_{3D}, \mathbf{T}^{*}_{3D}\big), \\ \mathcal{L}^{\text{3D}}_{\text{vis}}(\mathbf{V}^{(M)}_{3D}, \mathbf{V}^{*}_{3D}) &= \ell_{\text{balanced-BCE}}\big(\mathbf{V}^{(M)}_{3D}, \mathbf{V}^{*}_{3D}\big), \end{aligned} \tag{15}$$

where $\mathbf{T}^{*}_{3D}$ is the GT 3D trajectory coordinates, and $\mathbf{V}^{*}_{3D}$ is the binary GT 3D visibility. We use a L1 loss for 3D trajectory coordinates in the world-space coordinates. We use a balanced Binary Cross Entropy loss for 3D trajectory visibility, which is robust to class imbalance.

To optimize time offsets at m-th iteration, we construct the supervised time offset loss as follows:

$$\mathcal{L}_{\text{offset}}(\tau^{(m)}_v) = \sum_{v=1}^{V} |\tau^{(m)}_v - \tau^{*}_v| \tag{16}$$

where τ^{*}_v is the GT time offset for view v.

C Dataset, Evaluation metrics, Implementation Details, and Baselines

C.1 Dataset

Our MV-Kubric2 dataset is constructed following the MV-Kubric generation pipeline (Rajič et al., 2025) which is built on Blender that produces photo-realistic multi-view synthetic videos with ground-truth annotations such as camera parameters, point trajectories, depths, and segmentations. To prevent data contamination such as repeated scenes across training, validation, and test sets, we use disjoint random seeds for options with randomness such as asset selection and camera placement. The MV-Kubric2 dataset consists of 5,265 scenes, and is split into 5,100, 82, and 83 scenes for training set, validation set, and test set, respectively. Following the annotation of (Rajič et al., 2025), we evaluate the methods on our MV-Kubric2 test set (83 scenes). To prove generalizability

and robustness of method, we also evaluate the methods on four real-world datasets: CMU Panoptic Studio dataset (6 scenes) (Joo et al., 2015), DexYCB dataset (10 scenes) (Chao et al., 2021), Harmony4D dataset (12 scenes) (Khirodkar et al., 2024), and 3D-POP dataset (4 scenes) (Naik et al., 2023). CMU Panoptic Studio dataset (Joo et al., 2015) is a real-world dataset captured inside the constrained dome with calibrated and synchronized cameras that record multi-person activities (e.g., basketball, juggle, and tennis) and provide optimization-based ground-truth depths. DexYCB dataset (Chao et al., 2021) is a benchmark for capturing hand grasping of objects under eight RGB-D cameras, providing MANO 3D hand poses. Since the DexYCB dataset does not have high-quality ground-truth depth maps and it only has raw sensor depths with many holes, we utilized DUSt3R-estimated depths (Wang et al., 2024) in the main tables. Harmony4D dataset (Khirodkar et al., 2024) curated by Koo et al. (2026) is a real-world video dataset for in-the-wild close contact human interactions (e.g. grappling, karate, and mma) that are captured from about 20 synchronized cameras and provide ground-truth annotations such as camera parameters and 2D/3D human poses. 3D-POP dataset (Naik et al., 2023) is a video dataset with marker-based motion capture where freely moving small distant pigeons are captured with 4 synchronized cameras to provide keypoint annotations of 2D/3D posture movements.

We simulate unsynchronized multi-view input videos by sampling ground-truth per-view time offsets from the normal distribution $\mathcal{N}(0, \sigma_\tau^2)$, where $\sigma_\tau = 5$ frames for 12 fps MV-Kubric2 dataset, and $\sigma_\tau = 3$ frames for 10 fps DexYCB dataset, and $\sigma_\tau = 12.5$ frames for 30 fps CMU Panoptic Studio, Harmony4D, and 3D-POP dataset unless otherwise specified.

C.2 Evaluation metrics

As mentioned in Sec. 4, we measure **per-view time offset estimation** performance by (i) δ_{mean}, (ii) δ_{median}, (iii) δ_{std}, and (iv) $A@100$ and $A@500$, where AUC($A@100$, $A@500$) is more robust metric to outliers due to its cumulative formulation. We exclude the reference view (e.g. Cam. 1) for the metrics δ_{mean}, δ_{median}, and δ_{std} to ensure a fair comparison since τ_1 is always 0 in our setting. We record the synchronization metrics in the unit of milliseconds [*ms*] to indicate the synchronization error regardless of the frame rate of each dataset.

We measure **2D/3D tracking** performance by (i) ψ_{pa}, (ii) ψ_{oa}, and (iii) ψ_{aj}, where we additionally measure (iv) δ_{mte} for 3D tracking performance. For 2D ψ_{pa} and 2D ψ_{aj}, they are averaged over thresholds $\mathcal{T}_{2d} = \{1, 2, 4, 8, 16\}$ in pixel coordinates. For 3D ψ_{pa} and 3D ψ_{aj}, they are averaged over distance thresholds $\mathcal{T}_{3d} = \{0.05, 0.1, 0.2, 0.4, 0.8\}$ in MV-Kubric2 dataset and $\mathcal{T}_{3d} = \{0.05, 0.1, 0.2, 0.4\}$ in CMU Panoptic Studio dataset in world-space coordinates. Following the mathematical notation of Sec. 3.1, the exact equations of each metric is shown as follows:

Evaluation Metrics for **per-view time offset estimation performance**:

$$
\begin{aligned}
\delta_{\text{mean}} &= \frac{1}{V-1} \sum_{v=1, v\neq\tilde{v}}^{V} e_v, \\
\delta_{\text{median}} &= \text{median}\{e_1, e_2, \ldots, e_v, \ldots, e_V|_{v\neq\tilde{v}}\}, \\
\delta_{\text{std}} &= \sqrt{\frac{1}{V-1} \sum_{v=1, v\neq\tilde{v}}^{V} (e_v - \delta_{\text{mean}})^2}, \\
A@\tau &= \frac{100}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \max\left(1 - \frac{e_{i,j}}{\tau}, 0\right),
\end{aligned}
\tag{17}
$$

where $e_v = |\tau_v^* - \tau_v^{(M)}|$ is per-view time offset error of view v as the difference between the ground-truth and the final predicted(at M-th iteration) value of per-view time offset, $\tilde{v}$ is the reference view (e.g. Cam. 1) to exclude at time offset metrics for a fair comparison since τ_1 is always 0 in our model, $e_{i,j} = |\tau_{i,j}^* - \tau_{i,j}^{(M)}|$ is pairwise time offset error between view i and j as the difference between the ground-truth and the final predicted(at M-th iteration) value of pairwise time offset, and $\mathcal{P}$ is a set of view pairs.

Evaluation Metrics for **2D/3D tracking performance**:

$$
\begin{aligned}
\psi_{\text{pa}} &= \frac{1}{K} \sum_{\tau \in \mathcal{T}_{2d/3d}} \left(\frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{t=1}^{T} \mathbb{I}(v^{*n,t}=1)\mathbb{I}(\|\mathbf{T}_{2D/3D}^{(M)}{}^{n,t} - \mathbf{T}_{2D/3D}^{*}{}^{n,t}\|_2 < \tau)}{\sum_{t=1}^{T} v^{*n,t}} \right), \\
\psi_{\text{oa}} &= \frac{1}{N} \sum_{n=1}^{N} \left(\frac{1}{T} \sum_{t=1}^{T} \mathbb{I}(v^{(M)n,t} = v^{*n,t}) \right), \\
\psi_{\text{aj}} &= \frac{1}{K} \sum_{\tau \in \mathcal{T}_{2d/3d}} \left(\frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{t=1}^{T} v^{*n,t} v^{(M)n,t} \alpha^{n,t,\tau}}{\sum_{t=1}^{T} \left(v^{*n,t} + (1 - v^{*n,t}) v^{(M)n,t} + v^{*n,t} v^{(M)n,t} (1 - \alpha^{n,t,\tau}) \right)} \right), \\
\delta_{\text{mte}} &= \frac{1}{N} \sum_{n=1}^{N} \text{median} \left\{ \|\mathbf{T}_{3D}^{(M)}{}^{n,t} - \mathbf{T}_{3D}^{*}{}^{n,t}\|_2 \;\middle|\; t = 1, \ldots, T,\ v^{*n,t} = 1 \right\},
\end{aligned}
\tag{18}
$$

where $K = \text{len}(\mathcal{T}_{2d/3d})$ is the number of thresholds in $\mathcal{T}_{2d/3d}$, $\mathbb{I}(\cdot)$ is a binary indicator function, $v = \mathbf{V}_{2D/3D}$ is the visibility of 2D or 3D trajectories, $\mathbf{X}^*$ and $\mathbf{X}^{(M)}$ are respectively the ground-truth and the final predicted(at M-th iteration) value for each variable $\mathbf{X} \in \{\mathbf{T}_{3D}, \mathbf{V}_{3D}, \mathbf{T}_{2D}, \mathbf{V}_{2D}\}$, and $\alpha^{n,t,\tau} := \mathbb{I}(\|\mathbf{T}_{2D/3D}^{(M)}{}^{n,t} - \mathbf{T}_{2D/3D}^{*}{}^{n,t}\|_2 < \tau)$ for n-th trajectory point, t-th frame, and $\tau \in \mathcal{T}_{2d/3d}$.

C.3 Implementation details

We provide further explanation on implementation details not covered in 4. We train our SyncTAP utilizing various techniques such as gradient checkpointing (Chen et al., 2016), flash-attention (Dao, 2024), and mixed-precision (Micikevicius et al., 2017) for memory-efficiency. We use AdamW (Loshchilov & Hutter, 2017) optimizer with base LR=1e-4, weight decay=1e-4, and OneCycleLR scheduler.

To process long videos, we use a sliding window of 12 frames with a stride of 8, starting at sufficiently negative timestamp on the reference camera's time axis to cover any view with negatively initialized time offset. The 2D query frame is set to the first visible frame for each view, and the 3D query frame is set to the reference cam's first frame. We apply masked attention to exclude empty (duplicated) edge frames in the current sliding window due to time offsets. Within model, EpiRef samples time offset candidates $\tau_j^{(m)} + \delta \times s$ for $\delta \in \{-2, \ldots, 2\}$ and $s = 2$. In CVTAL, frames attend to 5 nearby frames in other views. The refinement is performed over $M = 4$ iterations.

As a fair comparison, we established the same evaluation protocol across methods. $V = 4$ randomly sampled views are fixed across methods in each dataset, and we use $N = 300$ queries unless otherwise specified. As mentioned in Appendix D.9, we fix random seed as zero in the main tables for ensuring reproducibility and fair comparison. Due to per-view time offsets, the length of time interval with valid input frame becomes longer than T. In the main tables, the 3D tracking performance is evaluated for the whole frame interval where at least one view is valid and we can obtain 3D trajectories from the valid input frame. Exceptionally, in Table 3, the 3D tracking performance is evaluated for the overlapping frame interval where all views are valid in order to focus on each component's contribution of our model under full-view setup.

C.4 Baselines

We compare the synchronization performance on three baselines whose code is available: (i) MASt3R (Leroy et al., 2024) utilizes attention confidence scores to capture the similarity between frames, (ii) VisualSync (Liu et al., 2025) is the SOTA visual-based synchronization method which performs an exhaustive temporal search of the Sampson error using external foundation models as preprocessing, and (iii) SyncNeRF (Kim et al., 2024) in a Sync-K-Planes representation is a per-scene optimization-based synchronization method of which training took around 5 minutes per scene on average for 5k iterations. We also trained for 15k iterations, but the synchronization performance was not that different.

Table 8: **Importance of synchronization in multi-view videos.**

Video Input	Method	Tracking						
		2D ψ_{aj} ↑	2D ψ_{pa} ↑	2D ψ_{oa} ↑	3D ψ_{aj} ↑	3D ψ_{pa} ↑	3D ψ_{oa} ↑	δ_{mte} ↓
Monocular ($V=1$)	MV-TAP (Koo et al., 2026)	63.9	79.4	85.5	–	–	–	–
Monocular ($V=1$)	MVTracker (Rajič et al., 2025)	–	–	–	47.2	68.7	70.1	22.6
Unsync. ($V=4$)	MV-TAP (Koo et al., 2026)	50.9	69.1	79.6	–	–	–	–
Unsync. ($V=4$)	MVTracker (Rajič et al., 2025)	–	–	–	51.6	70.6	76.8	20.4
Sync. ($V=4$)	MV-TAP (Koo et al., 2026)	66.2	81.3	87.1	–	–	–	–
Sync. ($V=4$)	MVTracker (Rajič et al., 2025)	–	–	–	53.8	71.7	79.0	18.7

Table 9: **Multi-view 2D tracking performance** on MV-Kubric2, CMU Panoptic Studio, and DexYCB datasets.

Method	Sync.	MV-Kubric2			CMU Panoptic Studio			DexYCB		
		ψ_{aj} ↑	ψ_{pa} ↑	ψ_{oa} ↑	ψ_{aj} ↑	ψ_{pa} ↑	ψ_{oa} ↑	ψ_{aj} ↑	ψ_{pa} ↑	ψ_{oa} ↑
MV-TAP (Koo et al., 2026)		50.9	69.1	79.6	37.7	50.3	74.4	65.3	82.5	86.8
MASt3R (Leroy et al., 2024) + MV-TAP (Koo et al., 2026)	✓	57.8	75.3	83.7	35.9	51.1	73.2	64.8	83.5	83.9
VisualSync (Liu et al., 2025) + MV-TAP (Koo et al., 2026)	✓	61.6	77.7	85.9	48.8	64.8	81.6	71.3	85.8	88.3
SyncTAP (Ours)	✓	**64.3**	**79.9**	**86.6**	**60.1**	**77.2**	**84.9**	**73.0**	**86.9**	**89.6**

For 2D/3D tracking performance, we utilize the most recent multi-view 2D tracker MV-TAP (Koo et al., 2026) and 3D tracker MVTracker (Rajič et al., 2025) as the backbone of our SyncTAP and as the naive baseline. We compare the 2D/3D tracking performance on three baselines: (i) single baselines without synchronization (MV-TAP for 2D, and MVTracker for 3D) , (ii) MASt3R-cascaded baselines (MASt3R + MV-TAP/MVTracker), and (iii) VisualSync-cascaded baselines (VisualSync + MV-TAP/MVTracker).

D Additional ablation studies and analysis

D.1 Importance of Synchronization in Multi-View Videos

Table 8 validates the importance of the synchronization task for 2D/3D multi-view tracking from unsynchronized multi-view input videos. The table compares the 2D/3D tracking performance of the SOTA multi-view tracking baselines(MV-TAP (Koo et al., 2026) for 2D and MVTracker (Rajič et al., 2025) for 3D) on the MV-Kubric2 dataset of $\sigma_\tau = 5$ over different setups: (i) monocular setup, (ii) multi-view, but unsynchronized setup by setting predicted offsets as zero, and (iii) multi-view and synchronized setup by setting predicted offsets as GT offsets. Table 8 shows the severe tracking degradation for unsynchronized multi-view input videos, where it often pushes below the tracking performance of monocular setup. Multi-view point tracking is useful for dynamic 3D scene understanding since it resolves view-specific occlusions and depth ambiguities, but unsynchronized multi-view videos often do not have better advantages than just monocular video. Thus, the synchronization task is very important for unsynchronized multi-view input videos to fully utilize the advantages of multi-view inputs.

D.2 Inference Runtime

Table 10 compares the average inference runtime per scene on the MV-Kubric2 dataset under a representative setting (V=4, T=24, N=300) excluding the runtime for depth estimation (We utilize ground-truth or already estimated depth maps contained in each dataset folder for all methods). As shown in Table 10 (a), on the MV-Kubric2 dataset unsynchronized with $\sigma_\tau = 5$, the cascaded baseline of VisualSync and MVTracker requires 58.2s due to heavy preprocessing for synchronization with external foundation models such as segmentation, dense tracking, and matching, while our SyncTAP takes only 4.7s for joint estimation of both synchronization and multi-view tracking. As shown in Table 10 (b), highly unsynchronized videos with higher σ_τ results in longer runtime.

Table 10: **Inference runtime on MV-Kubric2 dataset.** Average runtime per scene (V=4, T=24, N=300). (a) Comparison of our SyncTAP with the cascaded baselines. (b) Runtime of our SyncTAP across varying σ_τ.

Method	**Runtime (s)** ↓
MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	51.0
VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	58.2
SyncTAP (Ours)	**4.7**

(a) Total runtime including synchronization.

Method	**Runtime (s)** ↓
SyncTAP (Ours) (σ_τ=0)	3.8
SyncTAP (Ours) (σ_τ=2)	4.5
SyncTAP (Ours) (σ_τ=5)	4.7
SyncTAP (Ours) (σ_τ=10)	5.0
SyncTAP (Ours) (σ_τ=15)	5.1

(b) Runtime vs. unsync. level σ_τ.

Table 11: **Synchronization performance of our SyncTAP on the Harmony4D dataset** for success case and failure case.

Scene	# Scenes	**Synchronization**				
		δ_{mean} ↓	δ_{median} ↓	δ_{std} ↓	A@100 ↑	A@500 ↑
Success ($\delta_{\text{mean}} < 100$ms)	7	**32.0**	**24.0**	**19.8**	**68.7**	**93.8**
Failure ($\delta_{\text{mean}} \geq 100$ms)	5	280.8	268.1	188.9	22.2	45.3
All scenes	12	135.6	125.7	90.3	49.3	73.6

Table 12: **Per-scene synchronization performance of our SyncTAP on the Harmony4D dataset.**

Scene	**Synchronization**				
	δ_{mean} ↓	δ_{median} ↓	δ_{std} ↓	A@100 ↑	A@500 ↑
Success ($\delta_{mean} < 100ms$)					
04_sword_part1	10.7	6.9	9.5	86.9	97.4
14_mma3	18.4	16.9	3.0	90.3	98.1
05_sword2	26.9	32.4	14.0	67.7	93.5
11_karate3	33.4	25.4	21.6	60.2	92.0
09_karate	40.2	24.5	27.9	65.0	93.0
04_sword_part2	45.8	20.8	45.7	42.6	88.8
15_mma4	48.5	41.0	16.9	68.0	93.6
Single-view failure ($100ms \leq \delta_{mean} < 200ms$)					
13_mma2	133.1	47.7	121.2	38.0	74.6
10_karate2	170.0	28.0	205.7	42.4	61.3
Failure ($\delta_{mean} \geq 200ms$)					
02_grappling	299.5	202.0	168.6	9.6	39.2
01_hugging	400.0	500.0	190.5	0.0	20.6
07_ballroom	401.3	563.1	258.4	20.8	30.8
All (12 scenes)	135.6	125.7	90.3	49.3	73.6

Since we start the first sliding window at sufficiently negative frame in terms of the reference camera to cover any view with negatively initialized time offsets, higher σ_τ widens the temporal range to estimate, leading to more sliding windows and consequently a longer runtime.

D.3 Failure cases on Harmony4D dataset

Table 1 in the main paper demonstrates that the Harmony4D dataset (Khirodkar et al., 2024) shows the degraded synchronization performance compared to other datasets. We analyzed the failure cases of our SyncTAP on the Harmony4D dataset by measuring the synchronization performance for each scene. Table 12 shows that our SyncTAP fails for three scenes: 02_grappling, 01_hugging, and 07_ballroom. For 02_grappling scene, the selected reference view with random seed 0 has almost no valid (dynamic and co-visible) trajectory pair to utilize for synchronization. For 01_hugging scene, two humans hug each other very slowly, which leads to almost no valid (dynamic and co-visible) trajectory pair to utilize for synchronization. For 07_ballroom scene, two humans rotate in place, which confuses time offset estimation. Table 11 measures the synchronization performance

Table 13: **Synchronization robustness for input query source on CMU Panoptic Studio dataset. Bold**: best and underline: second-best are marked among synchronization methods for each input query source.

Input Query	Method	N	Synchronization				Runtime (s) ↓
			δ_{mean} ↓	δ_{median} ↓	A@100 ↑	A@500 ↑	
-	GT offset	-	361.1	355.6	8.3	33.4	-
ViTPose	MASt3R (Leroy et al., 2024)	55	479.6	511.1	14.4	42.0	91.5
ViTPose	VisualSync (Liu et al., 2025)	55	259.3	144.4	27.3	53.6	6.1
ViTPose	**SyncTAP (Ours)**	55	**64.4**	**45.4**	**62.3**	**87.0**	5.4
GT	VisualSync (Liu et al., 2025)	55	181.5	138.9	41.2	65.1	6.1
GT	**SyncTAP (Ours)**	55	**53.2**	**36.5**	**62.2**	**88.5**	5.4

separately for scenes of success cases and failure cases. For scenes of success cases, our SyncTAP on the Harmony4D dataset also achieves highly accurate synchronization performance similar to other datasets in Table 1.

D.4 Robustness for Estimated Queries

Table 13 compares the synchronization performance of methods on the CMU Panoptic Studio dataset for two input query sources: GT queries or ViTPose (Xu et al., 2022) estimated queries. This experiment proves that our SyncTAP robustly works on multi-view datasets capturing humans, even when the ground-truth queries are not provided. We evaluated on the CMU Panoptic Studio dataset which contains multi-person to estimate its 2D human pose. Also, only the synchronization performance is measured since there is no ground-truth 3D trajectories corresponding to those human pose estimated queries. ViTPose (Xu et al., 2022) provides 2D human pose estimation and we utilize them for the initialization of multi-view 2D queries. The number of query points made from the human pose joints of ViTPose is 55, 28, 96, 48, 56, and 47 for basketball, boxes, football, juggle, softball, and tennis scene, respectively; the average number of detected human pose joints is then $N = 55$. We compare these results with the cases of using GT queries where the GT queries are randomly sampled as the same number of the ViTPose queries for a fair comparison. It should be noted that VisualSync (Liu et al., 2025) is originally designed to utilize dense grid queries to estimate monocular trajectories by CoTracker3 (Karaev et al., 2025). So, for a fair comparison, we modified it to utilize the given (GT or ViTPose-estimated) input queries for monocular trajectory estimation. Additionally, using given cross-view one-to-one correspondent queries instead of dense grid queries reduces VisualSync's runtime since it skips cross-view matching process with MASt3R.

With the estimated ViTPose query initialization, our SyncTAP outperforms VisualSync since our SyncTAP has much more accurate time offset estimation module: EpiInit for robust time offset initialization and EpiRef for data-driven time offset refinement. For VisualSync, the synchronization performance with the estimated(noisy) ViTPose query initialization gets worse than that with the GT query initialization. However, for our SyncTAP, both the estimated ViTPose query initialization and the GT query initialization show similar synchronization performance, which proves that our SyncTAP is robust to the quality of initialized queries. Surprisingly, our SyncTAP with the estimated ViTPose query initialization even outperforms VisualSync with GT query initialization, corresponding to approximately 64.5% reductions and 33.6% improvements in δ_{mean} and $A@500$, respectively. It's because our SyncTAP has accurate sub-frame time offset estimation modules and utilizes diverse strategies to deal with the noisy input estimates as mentioned in Appendix D.10.

D.5 Robustness for Estimated Camera Parameters and Depths

Table 14 compares the synchronization and 3D tracking performance of our SyncTAP on the DexYCB dataset for different combinations of camera and depth sources. For depth sources, we compared (i) raw sensor depth given in DexYCB dataset, (ii) DUSt3R-estimated depth (Wang et al., 2024), and (iii) VGGT-Ω-estimated depth (Wang et al., 2026). For camera parameter sources, we compared (i) Ground-Truth camera parameters given in DexYCB dataset and (ii) VGGT-Ω-estimated camera parameters. Because VGGT-Ω-estimated camera parameters and GT 3D trajec-

Table 14: **Synchronization and 3D tracking robustness for estimated camera parameters and depths on DexYCB dataset.**

Camera	Depth	Synchronization				3D Tracking			
		$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	A@100 ↑	A@500 ↑	$\psi_{\text{aj}} \uparrow$	$\psi_{\text{pa}} \uparrow$	$\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$
GT	Raw sensor	44.7	32.4	70.4	91.4	49.1	60.4	89.1	8.2
GT	DUSt3R (Estimated)	44.7	32.4	70.4	91.4	74.4	85.3	89.6	3.0
VGGT-Ω (Estimated, aligned)	VGGT-Ω (Estimated)	59.2	68.4	68.5	89.8	73.4	84.6	89.5	2.8

Table 15: **Synchronization and 3D tracking robustness for varying time offset amounts σ_τ on MV-Kubric2 dataset.** For synchronization performance, **Bold**: best and underline: second-best are marked among only synchronization methods.

σ_τ (frames)	Method	Synchronization					3D Tracking			
		$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 ↑	A@500 ↑	$\psi_{\text{aj}} \uparrow$	$\psi_{\text{pa}} \uparrow$	$\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$
0 (Sync.)	No Sync. + MVTracker (Rajič et al., 2025)	0.0	0.0	0.0	100.0	100.0	**72.0**	**82.7**	91.1	**8.9**
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	151.6	151.6	75.5	47.5	76.7	61.0	73.7	85.9	14.8
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	41.5	37.1	29.1	69.0	92.2	67.9	79.0	89.8	10.8
	SyncTAP (Ours)	**16.9**	**16.2**	**6.5**	**85.7**	**97.1**	69.8	79.4	**92.7**	10.5
2	No Sync. + MVTracker (Rajič et al., 2025)	139.9	131.5	75.6	24.2	70.4	59.0	74.9	82.5	14.2
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	138.9	123.5	77.6	48.6	78.9	57.2	72.7	82.5	15.2
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	49.9	46.2	31.7	63.3	91.0	60.0	74.9	84.1	13.6
	SyncTAP (Ours)	**21.1**	**18.9**	**10.9**	**81.4**	**96.1**	**65.7**	**77.0**	**90.6**	**12.1**
5	No Sync. + MVTracker (Rajič et al., 2025)	351.1	325.3	188.7	9.9	39.3	51.6	70.6	76.8	20.4
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	157.0	147.6	79.2	40.0	74.7	50.8	69.6	77.3	20.3
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	94.4	73.3	71.7	56.9	85.0	52.8	71.1	78.4	19.8
	SyncTAP (Ours)	**30.2**	**25.4**	**18.2**	**73.3**	**94.1**	**62.0**	**73.6**	**90.2**	**15.8**
10	No Sync. + MVTracker (Rajič et al., 2025)	594.4	612.4	254.2	7.0	23.6	45.2	66.0	72.2	27.4
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	324.0	298.2	198.1	28.2	55.4	46.2	67.9	73.5	21.9
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	195.8	142.6	163.2	42.5	70.7	47.5	68.2	75.0	20.9
	SyncTAP (Ours)	**53.9**	**37.2**	**44.4**	**64.1**	**89.6**	**58.4**	**70.8**	**89.7**	**18.2**
15	No Sync. + MVTracker (Rajič et al., 2025)	718.2	772.1	236.2	7.2	19.2	43.2	64.2	70.0	32.0
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	392.6	317.3	249.0	22.2	48.0	44.5	67.1	71.9	24.5
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	279.5	211.8	218.2	35.0	62.6	45.4	67.1	73.4	22.8
	SyncTAP (Ours)	**72.0**	**54.4**	**54.2**	**58.7**	**87.1**	**57.4**	**69.9**	**89.3**	**20.2**

tories are defined in different 3D world coordinates, we apply a 7-DoF similarity transformation to align the two coordinates in order to evaluate the quantitative 3D tracking performance.

As shown in Table 14, the synchronization performance of our SyncTAP is independent of depth sources and remains robust even with (reasonably) estimated camera parameters. On the other hand, the 3D tracking performance of our SyncTAP reflects the quality of the input depths and remains robust even with (reasonably) estimated cameras and depths. Note that replacing only depth source does not change the synchronization performance because the fundamental matrix used for synchronization depends on camera parameters but not depth. Input depth quality affects only the 3D tracking performance. Since the raw sensor depth given in DexYCB dataset contains many holes and noise, the 3D tracking performance with the raw sensor depth degrades much more than that with the estimated DUSt3R depth or the estimated VGGT-Ω depth.

D.6 Robustness for Varying Time Offset Amounts

Table 15, 16, and 17 compare the synchronization performance and the 3D tracking performance against baseline methods across various ground-truth time offset standard deviations σ_τ on the MV-Kubric2 dataset, CMU Panoptic Studio dataset, and DexYCB dataset, respectively. Although our SyncTAP is trained only with $\sigma_\tau = 5$ on the 12 fps MV-Kubric2 dataset, it achieves consistently accurate sub-frame level time offset estimation, maintaining a mean error $\delta_{\text{mean}} \leq 72$ *ms* (< 1 frame) regardless of varying levels of unsynchronization. In contrast, baseline methods such as MASt3R (Leroy et al., 2024) and VisualSync (Liu et al., 2025) suffer from severe performance degradation for larger σ_τ, where our SyncTAP reduces δ_{mean} by 74.2% against the SOTA visual-based synchronization method VisualSync (Liu et al., 2025) at an extreme unsynchronization level

Table 16: **Synchronization and 3D tracking robustness for varying time offset amounts σ_τ on CMU Panoptic Studio dataset.** For synchronization performance, **Bold**: best and underline: second-best are marked among only synchronization methods.

σ_τ (frames)	Method	Synchronization					3D Tracking			
		$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 ↑	A@500 ↑	$\psi_{\text{aj}} \uparrow$	$\psi_{\text{pa}} \uparrow$	$\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$
0 (Sync.)	No Sync. + MVTracker (Rajič et al., 2025)	0.0	0.0	0.0	100.0	100.0	81.4	**98.0**	85.6	**2.2**
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	298.1	322.2	145.3	13.4	55.1	67.4	88.3	78.0	5.2
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	161.1	155.6	48.6	55.1	77.2	72.0	91.7	79.7	3.9
	SyncTAP (Ours)	**7.0**	**5.1**	**3.4**	**94.3**	**98.9**	**84.4**	97.2	**89.1**	2.4
2	No Sync. + MVTracker (Rajič et al., 2025)	55.6	50.0	27.0	43.1	88.0	75.5	96.5	81.3	**2.6**
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	329.6	388.9	150.2	11.6	45.3	60.9	85.8	74.3	5.2
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	164.8	166.7	52.2	48.6	74.5	68.1	91.5	76.2	4.0
	SyncTAP (Ours)	**6.7**	**6.1**	**3.9**	**92.8**	**98.6**	**80.5**	**96.5**	**86.8**	2.7
5	No Sync. + MVTracker (Rajič et al., 2025)	140.7	133.3	70.6	19.0	69.2	69.5	94.8	76.8	3.2
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	285.2	316.7	137.2	16.7	58.7	62.3	90.3	72.9	4.2
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	151.9	127.8	75.0	44.9	77.3	60.1	87.8	71.9	4.9
	SyncTAP (Ours)	**9.7**	**7.1**	**5.9**	**90.7**	**98.1**	**77.8**	**95.1**	**86.3**	**3.0**
10	No Sync. + MVTracker (Rajič et al., 2025)	288.9	283.3	131.5	8.8	41.2	60.7	87.9	70.5	6.0
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	435.2	350.0	252.9	22.7	44.6	58.5	87.2	70.5	5.6
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	183.3	177.8	110.7	35.2	65.6	54.5	83.8	66.3	6.1
	SyncTAP (Ours)	**28.1**	**19.6**	**23.2**	**70.0**	**94.1**	**69.6**	**88.7**	**85.5**	**4.2**
12.5	No Sync. + MVTracker (Rajič et al., 2025)	361.1	355.6	163.6	8.3	33.4	56.9	84.1	67.6	7.8
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	479.6	511.1	209.7	14.4	42.0	53.0	81.6	66.2	7.6
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	168.5	166.7	89.2	31.5	67.8	51.7	80.6	65.0	7.0
	SyncTAP (Ours)	**29.4**	**27.5**	**18.3**	**70.9**	**94.0**	**67.7**	**86.5**	**85.4**	**4.4**
15	No Sync. + MVTracker (Rajič et al., 2025)	416.7	422.2	182.1	7.4	28.0	55.8	81.8	66.7	8.8
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	529.6	561.1	269.0	9.7	38.1	50.0	78.7	63.7	8.5
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	350.0	416.7	128.5	31.5	56.0	52.0	79.4	65.0	8.6
	SyncTAP (Ours)	**110.5**	**114.9**	**68.8**	**49.5**	**77.7**	**61.3**	**82.6**	**81.6**	**5.7**

Table 17: **Synchronization and 3D tracking robustness for varying time offset amounts σ_τ on DexYCB dataset.** For synchronization performance, **Bold**: best and underline: second-best are marked among only synchronization methods.

σ_τ (frames)	Method	Synchronization					3D Tracking			
		$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 ↑	A@500 ↑	$\psi_{\text{aj}} \uparrow$	$\psi_{\text{pa}} \uparrow$	$\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$
0 (Sync.)	No Sync. + MVTracker (Rajič et al., 2025)	0.0	0.0	0.0	100.0	100.0	**79.3**	**88.3**	89.5	**2.08**
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	133.3	130.0	64.8	31.7	80.0	75.2	84.8	87.3	3.20
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	**26.7**	**10.0**	33.0	68.3	93.8	78.3	87.4	88.9	2.53
	SyncTAP (Ours)	39.1	36.4	**16.1**	**73.8**	**94.0**	78.2	84.8	**93.0**	3.31
1	No Sync. + MVTracker (Rajič et al., 2025)	86.7	70.0	50.6	36.7	81.2	72.0	85.6	84.1	**1.88**
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	183.3	170.0	60.3	23.3	73.0	69.5	83.1	83.2	2.78
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	46.7	40.0	45.2	66.7	88.3	71.9	**85.8**	83.6	2.57
	SyncTAP (Ours)	**36.3**	**26.1**	**21.1**	**73.1**	**93.7**	**75.6**	84.2	**91.1**	3.21
2	No Sync. + MVTracker (Rajič et al., 2025)	150.0	150.0	70.4	26.7	70.0	69.6	84.3	82.3	**2.63**
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	223.3	210.0	84.1	13.3	64.5	67.2	81.7	81.1	3.60
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	86.7	**40.0**	73.8	58.3	83.8	70.4	**85.5**	82.5	2.63
	SyncTAP (Ours)	**42.2**	44.8	**24.2**	**74.3**	**92.8**	**74.4**	85.0	**89.4**	3.16
3	No Sync. + MVTracker (Rajič et al., 2025)	193.3	190.0	57.1	20.0	61.5	67.4	82.7	80.6	3.22
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	233.3	230.0	112.5	18.3	62.0	66.4	82.2	79.0	3.37
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	133.3	100.0	100.8	45.0	73.7	67.2	83.4	79.7	2.98
	SyncTAP (Ours)	**44.7**	**32.4**	**33.3**	**70.4**	**91.4**	**74.4**	**85.3**	**89.6**	**2.95**
5	No Sync. + MVTracker (Rajič et al., 2025)	216.7	220.0	50.6	18.3	57.2	65.7	82.1	78.8	3.41
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	256.7	250.0	134.4	18.3	56.8	65.0	81.7	78.7	3.65
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	126.7	100.0	111.5	48.3	75.7	64.9	82.8	78.1	**3.13**
	SyncTAP (Ours)	**53.1**	**59.2**	**26.4**	**63.8**	**89.2**	**73.0**	**84.5**	**89.0**	3.19

of $\sigma_\tau = 15$ on the MV-Kubric2 dataset. The positive effect of our SyncTAP becomes larger over baselines in highly unsynchronized settings. In the MV-Kubric2 and DexYCB dataset, other baselines show the degradation of the 3D tracking performance rapidly for larger σ_τ, but our SyncTAP robustly performs 3D tracking with less drop. Furthermore, for perfectly synchronized videos ($\sigma_\tau = 0$), our SyncTAP estimates time offsets well to be nearly zero, ensuring that our framework robustly performs 2D/3D tracking regardless of whether the multi-view input videos are initially synchronized or not.

The 2nd-order Taylor series approximation in our EpiInit module is a local approximation which works well only near the current estimate of time offsets. As shown in A.2, for large time offsets,

Table 18: **Impact of the number of query points N on MV-Kubric2 dataset.**

N (queries)	3D Tracking				Synchronization					Runtime (s) ↓
	3D ψ_{aj} ↑	3D ψ_{pa} ↑	3D ψ_{oa} ↑	δ_{mte} ↓	δ_{mean} ↓	δ_{median} ↓	δ_{std} ↓	A@100 ↑	A@500 ↑	
64	56.1	67.3	89.0	21.7	44.8	36.7	33.0	69.4	91.7	3.7
128	59.3	70.7	89.6	18.6	35.6	27.4	23.0	70.7	93.3	4.0
256	61.8	73.3	90.3	16.0	32.5	26.6	19.5	72.8	93.9	4.5
300	62.0	73.6	90.2	15.8	30.2	25.4	18.2	73.3	94.1	4.7
512	63.5	75.2	90.5	14.4	**26.9**	**25.0**	**16.6**	**75.6**	**94.8**	5.9
1024	64.3	76.3	90.3	13.4	38.4	28.9	28.1	74.3	93.4	7.7
2048	**65.1**	**76.9**	**90.6**	**13.0**	33.1	27.9	21.7	74.8	93.9	11.0

Table 19: **Impact of the number of query points N on CMU Panoptic Studio dataset.**

N (queries)	3D Tracking				Synchronization					Runtime (s) ↓
	3D ψ_{aj} ↑	3D ψ_{pa} ↑	3D ψ_{oa} ↑	δ_{mte} ↓	δ_{mean} ↓	δ_{median} ↓	δ_{std} ↓	A@100 ↑	A@500 ↑	
64	60.8	82.5	79.3	5.9	118.5	103.3	94.8	46.6	75.4	5.2
128	64.9	84.3	84.3	5.3	32.4	17.5	28.9	72.0	92.8	6.0
256	66.8	85.8	85.0	4.7	32.5	21.2	24.3	71.4	93.1	6.9
512	66.9	87.2	84.2	4.3	47.7	25.7	45.4	61.8	89.2	8.6
1024	**68.2**	**87.2**	**85.8**	**4.3**	**22.8**	**16.1**	**16.4**	**78.2**	**95.6**	12.6

Table 20: **Impact of the number of query points N on DexYCB dataset.**

N (queries)	N_{valid}	3D Tracking				Synchronization					Runtime (s) ↓
		3D ψ_{aj} ↑	3D ψ_{pa} ↑	3D ψ_{oa} ↑	δ_{mte} ↓	δ_{mean} ↓	δ_{median} ↓	δ_{std} ↓	A@100 ↑	A@500 ↑	
64	28	71.8	82.6	89.6	3.3	44.4	43.9	21.7	68.7	91.1	2.4
128	49	73.5	84.6	89.4	2.9	44.1	42.0	**19.7**	70.1	91.5	2.5
256	97	74.8	85.2	**90.6**	2.9	**39.9**	**34.3**	24.7	69.9	**92.8**	2.9
512	185	75.0	85.5	90.2	2.8	52.6	37.2	32.8	71.1	90.9	3.7
1024	360	**75.7**	**86.2**	90.1	**2.6**	41.5	36.6	25.2	**71.1**	92.7	5.1

we empirically found that time offset estimation by the 2nd-order Taylor series approximation well converges via *iterative* EpiInit procedure as long as sufficient temporal overlaps across views remain to make valid trajectory point pairs. The iterative procedure within our EpiInit module and the iterative refinement between our EpiRef and STCV attention module make our SyncTAP be robust for large σ_τ.

D.7 Impact of the Number of Queries (N)

Table 18, 19, and 20 show the impact of the number of query points N on the 3D tracking performance, the synchronization performance, and the inference runtime in our MV-Kubric2 dataset, CMU Panoptic Studio dataset, and DexYCB dataset respectively. Our SyncTAP achieves consistently accurate synchronization accuracy across various values of N even with as few as 64 trajectories ($N = 64$) where even the number of valid (dynamic and co-visible) trajectories used for time offset estimation would be less than N.

The experiments demonstrate the robustness where, once a sufficient number of valid queries is available, our SyncTAP consistently achieves high synchronization accuracy independent of N. This is because, for input queries, we adopt the random query sampling where increasing N mostly re-samples the same motions. That is, after saturation where a sufficient number of valid queries is available, increasing the number of queries brings redundancy as many trajectories describe the same motion of the same dynamic objects. Therefore, the synchronization performance based on dynamic trajectory pairs remains consistently high across all N in our experiment.

Table 21: **Impact of the number of input views V on MV-Kubric2 dataset. Bold**: best and underline: second-best are marked among synchronization methods.

V (views)	Method	Synchronization					Runtime (s) ↓
		δ_{mean} ↓	δ_{median} ↓	δ_{std} ↓	A@100 ↑	A@500 ↑	
2	GT offset	363.5	363.5	0.0	9.4	39.2	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	156.6	156.6	0.0	52.8	76.3	9.7
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	116.5	116.5	0.0	55.6	82.9	13.8
	SyncTAP (Ours)	**37.9**	**37.9**	0.0	**66.8**	**92.4**	2.3
3	GT offset	365.0	365.0	144.1	9.0	37.6	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	193.8	193.8	64.3	38.3	71.0	26.2
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	108.4	108.4	69.3	55.2	84.0	32.7
	SyncTAP (Ours)	**47.3**	**47.3**	**26.9**	**71.3**	**92.3**	3.6
4	GT offset	351.1	325.3	188.7	9.9	39.3	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	157.0	147.6	79.2	40.0	74.7	51.0
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	94.4	73.3	71.7	56.9	85.0	58.2
	SyncTAP (Ours)	**30.2**	**25.4**	**18.2**	**73.3**	**94.1**	4.7
5	GT offset	346.1	319.8	207.9	9.8	38.4	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	181.0	152.6	113.4	34.1	69.5	83.7
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	90.6	56.2	96.4	53.6	82.6	86.7
	SyncTAP (Ours)	**33.8**	**26.7**	**25.3**	**70.6**	**93.2**	5.7
6	GT offset	342.4	297.2	220.2	10.1	36.8	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	179.9	149.6	114.6	39.6	71.2	120.9
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	78.1	52.2	83.8	51.9	84.6	113.6
	SyncTAP (Ours)	**36.5**	**26.0**	**30.0**	**69.7**	**92.5**	6.8
7	GT offset	348.7	307.2	231.0	9.3	35.5	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	164.0	140.1	110.7	37.5	71.2	166.9
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	74.3	51.2	76.2	53.8	85.3	149.7
	SyncTAP (Ours)	**36.0**	**30.3**	**26.3**	**66.9**	**92.5**	7.9
8	GT offset	345.8	301.2	234.4	8.8	34.7	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	171.8	131.5	142.8	33.6	67.6	219.8
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	75.4	46.2	89.1	53.7	84.3	189.9
	SyncTAP (Ours)	**35.5**	**30.4**	**26.3**	**67.5**	**92.8**	8.9

On the other hand, unlike the synchronization performance, we found that the 3D tracking performance continues to improve with N. As noted in CoTracker (Karaev et al., 2024), spatial attention among query tracks allows reliable tracking information to be shared along spatial domain, so the 3D tracking performance keeps improving with abundant information as the number of queries increases.

D.8 Impact of the Number of Input Views (V)

Table 21, 22, and 23 show the impact of the number of input views V on the synchronization performance and the inference runtime in our MV-Kubric2 dataset, CMU Panoptic Studio dataset, and DexYCB dataset, respectively. Our SyncTAP consistently outperforms VisualSync (Liu et al., 2025) for most V cases. While all methods achieve comparable synchronization performance regardless of V, they differ significantly in computational efficiency. MASt3R (Leroy et al., 2024) and VisualSync (Liu et al., 2025) require to run external foundation models for segmentation, tracking, and matching for every view pair and those pre-processes take much time, leading to a quadratic ($\mathcal{O}(V^2)$) increase in the inference runtime. In contrast, our SyncTAP obtains time offsets from internally and jointly estimated trajectories in our STCV network at light cost without relying on external models and our view-pairwise operations in EpiInit or EpiRef do not take much time, leading to approximately linear ($\mathcal{O}(V)$) increase in the inference runtime for small $V \leq 8$ which we used for our experiments.

Table 22: **Impact of the number of input views V on CMU Panoptic Studio dataset. Bold**: best and underline: second-best are marked among synchronization methods.

V (views)	Method	Synchronization					Runtime (s) ↓
		δ_{mean} ↓	δ_{median} ↓	δ_{std} ↓	A@100 ↑	A@500 ↑	
2	GT offset	411.1	411.1	0.0	0.0	25.6	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	**166.7**	**166.7**	0.0	22.2	66.7	19.2
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	377.8	377.8	0.0	11.1	38.9	10.7
	SyncTAP (Ours)	188.7	188.7	0.0	**40.9**	**66.7**	3.2
3	GT offset	333.3	333.3	161.1	11.1	37.0	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	408.3	408.3	241.7	13.0	45.2	52.9
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	297.2	297.2	108.3	24.1	45.2	25.0
	SyncTAP (Ours)	**100.9**	**100.9**	**55.2**	**53.2**	**79.6**	5.0
4	GT offset	361.1	355.6	163.6	8.3	33.4	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	479.6	511.1	209.7	14.4	42.0	102.0
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	168.5	166.7	89.2	31.5	67.8	47.0
	SyncTAP (Ours)	**29.4**	**27.5**	**18.3**	**70.9**	**94.0**	7.3
5	GT offset	350.0	338.9	203.2	9.4	33.7	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	484.7	469.4	279.9	8.6	32.6	167.0
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	295.8	305.6	147.2	29.4	58.8	76.0
	SyncTAP (Ours)	**104.2**	**110.3**	**69.9**	**47.6**	**77.7**	8.5
6	GT offset	347.8	344.4	214.2	14.3	35.1	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	505.6	427.8	277.0	10.4	29.2	244.7
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	294.4	194.4	263.1	26.9	56.1	104.8
	SyncTAP (Ours)	**52.7**	**25.4**	**55.1**	**60.8**	**87.8**	10.4
7	GT offset	342.6	350.0	220.8	12.0	35.0	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	537.0	569.4	307.0	4.6	26.5	332.4
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	218.5	216.7	127.3	29.0	61.9	133.5
	SyncTAP (Ours)	**87.4**	**69.4**	**61.2**	**62.0**	**86.0**	12.2
8	GT offset	338.1	355.6	211.5	11.3	36.5	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	511.1	477.8	325.4	5.2	30.0	443.6
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	173.0	161.1	115.8	30.1	65.5	176.6
	SyncTAP (Ours)	**54.9**	**37.5**	**54.0**	**65.8**	**87.6**	13.5

Table 23: **Impact of the number of input views V on DexYCB dataset. Bold**: best and underline: second-best are marked among synchronization methods.

V (views)	Method	Synchronization					Runtime (s) ↓
		δ_{mean} ↓	δ_{median} ↓	δ_{std} ↓	A@100 ↑	A@500 ↑	
2	GT offset	220.0	220.0	0.0	10.0	56.0	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	190.0	190.0	0.0	20.0	62.0	3.3
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	170.0	170.0	0.0	20.0	66.0	4.2
	SyncTAP (Ours)	**38.8**	**38.8**	0.0	**61.2**	**92.2**	1.7
3	GT offset	200.0	200.0	50.0	16.7	62.3	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	195.0	195.0	95.0	26.7	66.0	8.5
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	120.0	120.0	80.0	40.0	76.3	8.7
	SyncTAP (Ours)	**63.1**	**63.1**	**31.4**	**67.0**	**89.5**	2.3
4	GT offset	193.3	190.0	57.1	20.0	61.5	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	233.3	230.0	112.5	18.3	62.0	15.3
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	133.3	100.0	100.8	45.0	73.7	14.9
	SyncTAP (Ours)	**44.7**	**32.4**	**33.3**	**70.4**	**91.4**	3.0
5	GT offset	200.0	195.0	70.5	20.0	61.2	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	195.0	205.0	105.6	26.0	65.8	24.5
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	142.5	130.0	**60.5**	37.0	74.5	23.1
	SyncTAP (Ours)	**88.1**	**60.6**	71.6	**59.7**	**83.3**	3.6
6	GT offset	206.0	200.0	77.7	20.0	57.5	-
	MASt3R (Leroy et al., 2024) + MVTracker (Rajič et al., 2025)	194.0	180.0	133.0	14.7	60.1	36.3
	VisualSync (Liu et al., 2025) + MVTracker (Rajič et al., 2025)	142.0	140.0	91.0	39.3	74.7	32.8
	SyncTAP (Ours)	**50.6**	**27.7**	**57.8**	**69.4**	**88.2**	4.5

Table 24: **Per-seed synchronization performance δ_{mean} and its mean with 95% confidence interval.**

Seed	MV-Kubric2				CMU Panoptic Studio				DexYCB			
	GT offset	MASt3R	VisualSync	Ours	GT offset	MASt3R	VisualSync	Ours	GT offset	MASt3R	VisualSync	Ours
0 (main tables)	351.1	157.0	94.4	30.2	361.1	479.6	168.5	29.4	193.3	233.3	133.3	44.7
1	344.7	170.3	82.7	30.7	368.5	357.4	327.8	129.3	176.7	183.3	103.3	49.2
2	323.0	164.3	81.0	29.2	359.3	322.2	357.4	73.4	190.0	186.7	90.0	56.6
3	317.9	179.0	104.4	29.0	311.1	338.9	277.8	121.0	196.7	176.7	116.7	23.5
4	309.2	190.1	81.7	31.6	270.4	431.5	211.1	83.8	170.0	173.3	110.0	62.1
Mean $\pm$ 95% CI	329.2 ± 22.2	172.2 ± 16.0	88.8 ± 12.8	30.1 ± 1.3	334.1 ± 52.4	385.9 ± 83.1	268.5 ± 97.7	87.4 ± 49.9	185.3 ± 14.2	190.7 ± 30.3	110.7 ± 19.9	47.2 ± 18.4

Table 25: **Multi-view synchronization performance statistics.** Each metric's mean value over five random seeds with 95% confidence interval is reported.

Method	MV-Kubric2					CMU Panoptic Studio					DexYCB				
	$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 $\uparrow$	A@500 $\uparrow$	$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 $\uparrow$	A@500 $\uparrow$	$\delta_{\text{mean}} \downarrow$	$\delta_{\text{median}} \downarrow$	$\delta_{\text{std}} \downarrow$	A@100 $\uparrow$	A@500 $\uparrow$
GT offset	329.2 ± 22.2	306.2 ± 26.1	171.4 ± 12.2	10.3 ± 1.9	41.6 ± 2.9	334.1 ± 52.4	340.0 ± 52.4	158.3 ± 38.8	10.1 ± 3.6	39.3 ± 6.6	185.3 ± 14.2	186.0 ± 25.7	79.0 ± 17.5	15.7 ± 6.3	61.9 ± 1.9
MASt3R	172.2 ± 16.0	151.0 ± 21.3	96.8 ± 14.7	39.7 ± 3.1	72.3 ± 3.1	385.9 ± 83.1	358.9 ± 110.3	202.4 ± 67.0	8.2 ± 4.8	41.3 ± 6.8	190.7 ± 30.3	190.0 ± 29.1	101.1 ± 14.0	16.0 ± 5.0	64.7 ± 2.3
VisualSync	88.8 ± 12.8	68.7 ± 7.3	63.1 ± 11.2	54.7 ± 3.4	84.7 ± 1.1	268.5 ± 97.7	230.0 ± 134.2	182.4 ± 75.2	26.9 ± 6.6	57.4 ± 8.9	110.7 ± 19.9	82.0 ± 16.2	85.5 ± 21.9	43.3 ± 13.9	77.1 ± 5.4
SyncTAP (Ours)	$\mathbf{30.1 \pm 1.3}$	$\mathbf{25.8 \pm 2.4}$	$\mathbf{17.2 \pm 3.2}$	$\mathbf{74.5 \pm 0.9}$	$\mathbf{94.4 \pm 0.3}$	$\mathbf{87.4 \pm 49.9}$	$\mathbf{58.8 \pm 55.3}$	$\mathbf{65.7 \pm 65.8}$	$\mathbf{62.9 \pm 9.3}$	$\mathbf{84.4 \pm 8.2}$	$\mathbf{47.2 \pm 18.4}$	$\mathbf{41.2 \pm 23.0}$	$\mathbf{35.7 \pm 15.1}$	$\mathbf{67.6 \pm 9.2}$	$\mathbf{90.5 \pm 4.1}$

Table 26: **Multi-view 3D tracking performance statistics.** Each metric's mean value over five random seeds with 95% confidence interval is reported.

Method	MV-Kubric2				CMU Panoptic Studio				DexYCB			
	3D $\psi_{\text{aj}} \uparrow$	3D $\psi_{\text{pa}} \uparrow$	3D $\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$	3D $\psi_{\text{aj}} \uparrow$	3D $\psi_{\text{pa}} \uparrow$	3D $\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$	3D $\psi_{\text{aj}} \uparrow$	3D $\psi_{\text{pa}} \uparrow$	3D $\psi_{\text{oa}} \uparrow$	$\delta_{\text{mte}} \downarrow$
No Sync + MVTracker	51.2 ± 0.8	70.5 ± 0.9	76.5 ± 0.5	19.1 ± 1.6	55.0 ± 4.4	81.0 ± 3.7	68.0 ± 3.5	9.4 ± 1.5	66.0 ± 2.6	81.5 ± 4.3	79.0 ± 1.3	3.4 ± 1.2
MASt3R + MVTracker	51.2 ± 0.7	69.7 ± 0.4	77.9 ± 0.5	19.1 ± 1.2	50.6 ± 6.1	77.8 ± 5.5	66.4 ± 4.4	9.0 ± 2.2	65.7 ± 2.3	81.8 ± 4.0	78.9 ± 0.8	3.1 ± 1.0
VisualSync + MVTracker	53.1 ± 0.7	71.2 ± 0.9	78.8 ± 0.7	17.8 ± 1.8	52.0 ± 6.3	78.8 ± 5.4	66.9 ± 4.5	8.8 ± 2.5	67.2 ± 1.9	83.3 ± 3.4	79.7 ± 0.5	$\mathbf{3.0 \pm 1.0}$
SyncTAP (Ours)	$\mathbf{62.3 \pm 0.6}$	$\mathbf{74.0 \pm 0.7}$	$\mathbf{90.1 \pm 0.1}$	$\mathbf{15.0 \pm 1.2}$	$\mathbf{64.1 \pm 4.1}$	$\mathbf{84.1 \pm 3.7}$	$\mathbf{81.8 \pm 3.1}$	$\mathbf{6.1 \pm 2.1}$	$\mathbf{73.1 \pm 1.9}$	$\mathbf{83.4 \pm 2.7}$	$\mathbf{90.1 \pm 1.3}$	3.1 ± 0.7

D.9 Random Seed Statistics

Table 24 shows the synchronization performance of the Mean Absolute Error δ_{mean} for five different random seeds. It also calculates the mean value of δ_{mean} for the 5 random seeds with 95% confidence interval. Each seed re-samples the options which have randomness such as the input views, the ground-truth time offsets, and the input query points, where random seed 0 corresponds to the seed value which we used to report in the main tables. Our SyncTAP consistently achieves more accurate synchronization performance compared to the SOTA baseline VisualSync (Liu et al., 2025). Even the worst value of δ_{mean} of our SyncTAP among the 5 seeds is better than the best value of δ_{mean} of VisualSync among the 5 seeds.

Table 25 shows the mean value of each synchronization metric for 5 different random seeds with 95% confidence interval. Similarly, Table 26 shows the mean value of each 3D tracking metric for 5 different random seeds with 95% confidence interval. The two tables show the robustness of our SyncTAP to estimate highly accurate sub-frame time offsets and high-fidelity trajectories under unsynchronized multi-view videos across many trials of real applications.

D.10 Robustness to the Noise of Estimates

There are several strategies used to ensure the robustness against the estimates' noise.

Filtering. Savitzky-Golay filter smooths velocity and acceleration to reduce the jitters of estimated trajectories. Trajectory pairs are filtered by co-visibility, confidence, motion magnitude, and orthogonality to utilize only well-conditioned trajectory pairs for point-pairwise time offset estimation.

Robust Estimation and Aggregation. We compute the closed-form point-pairwise time offsets by taking numerically stable conjugate form with a first-order fallback to deal with tracker noise or negligible acceleration. When aggregating them to view-pairwise time offsets, we take the mode value of orthogonality-weighted histogram of point-pairwise time offsets to downweight non-reliable tra-

jectory pairs whose motions are nearly parallel to their epipolar lines. When aggregating them to per-view time offsets, we set the view-pair confidence weight using the number of valid point pairs and the sharpness of the pairwise time offset distribution to downweight less reliable view pairs. We set the stop criteria of convergence algorithm of our EpiInit in Appendix A.2 as the persistence or the descent of small time offset estimates so that the stop criteria are not triggered by a noise at a single iteration.

Noise-aware Training. For training, we augment input query positions and camera poses by injecting Gaussian noise perturbations. This ensures the robustness to estimated queries or estimated camera parameters at inference phase as shown in D.4 and D.5.

E Qualitative Results

This section provides additional qualitative results to compare the 3D tracking performance of our SyncTAP against the baselines under unsynchronized multi-view input videos. Furthermore, we provide demo videos for the same 5 scenes of this section by the supplementary material for clearer demonstration.

In this section, we visualize 3D trajectories by projecting them onto novel viewpoints unseen in 4-view inference input views. Successfully estimating multi-view consistent 3D trajectories leads to high-fidelity trajectory visualization on novel viewpoints. All the figures of Fig. 6, 7, 8, 9, and 10 show the degraded 3D tracking performance of the baselines for highly dynamic objects, but show the high-fidelity 3D tracking performance of our SyncTAP even for highly dynamic objects. The figures commonly demonstrate that for highly dynamic objects, even slight temporal misalignment can result in large spatial difference across views, causing 3D tracking to fail even with slightly inaccurate time offset estimation.

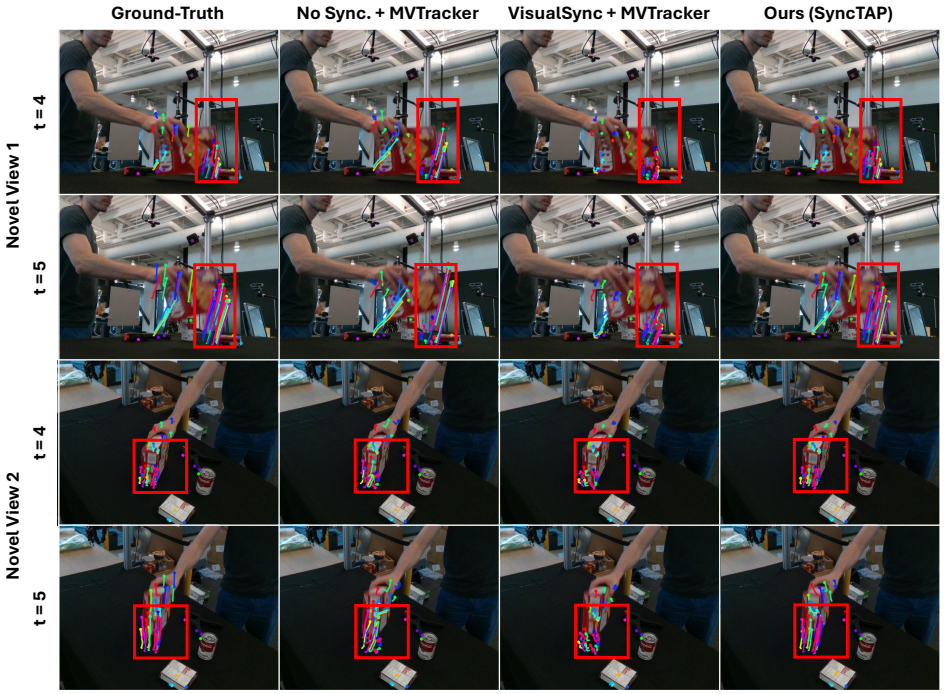

Figure 6: **Multi-view 3D point tracking for unsynchronized videos on the DexYCB dataset.** We visualize 3D trajectories by projecting them onto novel viewpoints unseen in 4-view inference inputs. Our SyncTAP robustly estimates high-fidelity 3D trajectories for highly dynamic object (e.g. cracker box in this figure).

Fig. 6 compares qualitative results of projected 3D trajectories across methods on the *20200813-subject-02__20200813_145653* scene of the DexYCB dataset by providing two frames for each of two novel viewpoints. For single baseline of MVTracker (Rajič et al., 2025) without synchronization, 3D points on the cracker box starts to move earlier since the 2D features from future frames of another view interfered due to temporal misalignments. Also, for cascaded baseline of VisualSync (Liu et al., 2025) and MVTracker (Rajič et al., 2025), it fails to track highly dynamic object, such as the cracker box, since time offset is not correctly estimated.

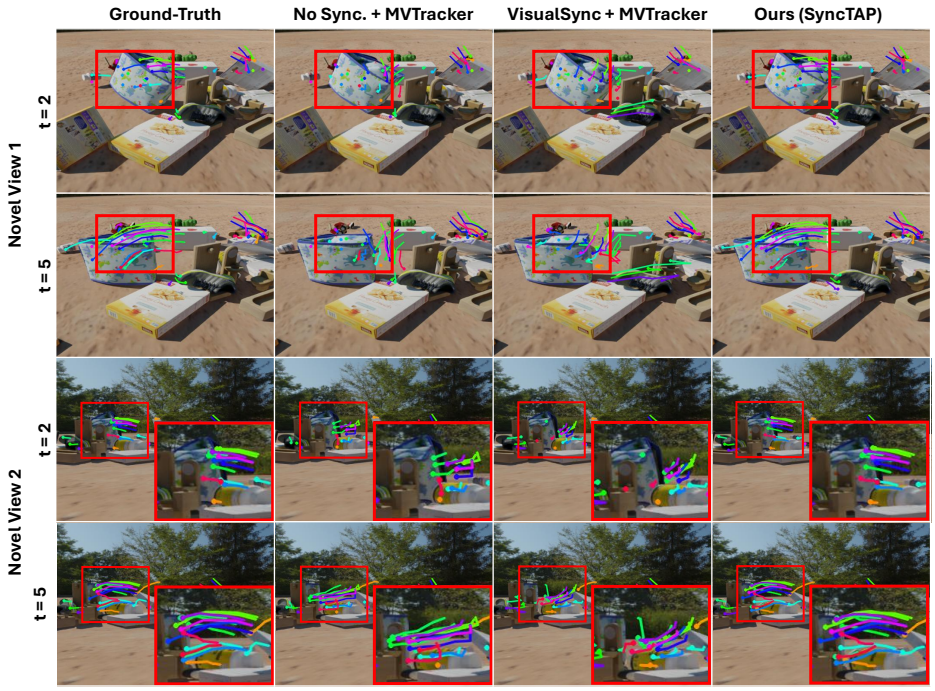

Figure 7: **Multi-view 3D point tracking for unsynchronized videos on the MV-Kubric2 dataset.** We visualize 3D trajectories by projecting them onto novel viewpoints unseen in 4-view inference inputs. Our SyncTAP robustly estimates high-fidelity 3D trajectories for highly dynamic object (e.g. blue basket in this figure).

Fig. 7 compares qualitative results of projected 3D trajectories across methods on the *1342* scene of the MV-Kubric2 dataset by providing two frames for each of two novel viewpoints. Fig. 4 in the main paper shows the projected 3D trajectories on the *1342* scene of the 12 fps MV-Kubric2 dataset by the setting of original input frames, and VisualSync (Liu et al., 2025) also performs high-quality 3D tracking with quite good synchronization. In contrast, Fig. 7 shows the projected 3D trajectories on the same *1342* scene of the 12 fps MV-Kubric2 dataset by the setting of downsampled($S = 2$) input frames, and VisualSync (Liu et al., 2025) fails to track highly dynamic objects such as blue basket.

Fig. 8 and 9 compares qualitative results of projected 3D trajectories across methods on the *juggle* scene of the CMU Panoptic Studio dataset for two different random seeds by providing two frames for each of two novel viewpoints. Both figures show that only our SyncTAP successfully tracks highly dynamic points such as hand with purple/light green trajectory and ball with pink trajectory in Fig. 8 and hand with purple/blue trajectory in Fig. 9.

Fig. 10 compares qualitative results of projected 3D trajectories across methods on the *tennis* scene of the CMU Panoptic Studio dataset by providing two frames for each of two novel viewpoints. This figure also shows that only our SyncTAP smoothly follows the 3D trajectories of highly dynamic tennis racket.

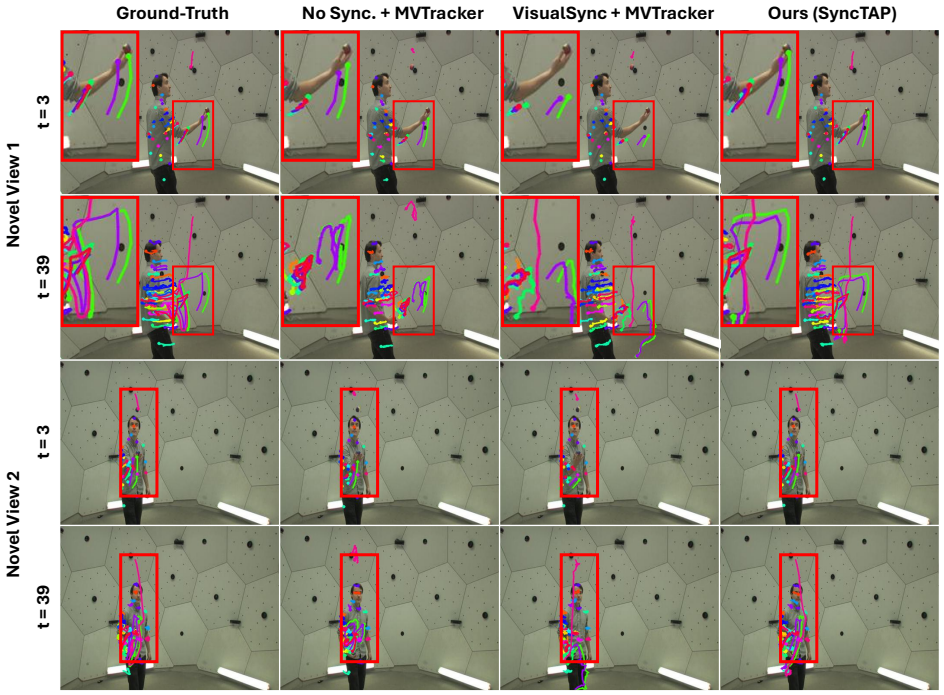

Figure 8: **Multi-view 3D point tracking for unsynchronized videos on the CMU Panoptic Studio dataset.** We visualize 3D trajectories by projecting them onto novel viewpoints unseen in 4-view inference inputs. Our SyncTAP robustly estimates high-fidelity 3D trajectories for highly dynamic object (e.g. hand with purple/light green trajectory and ball with pink trajectory).

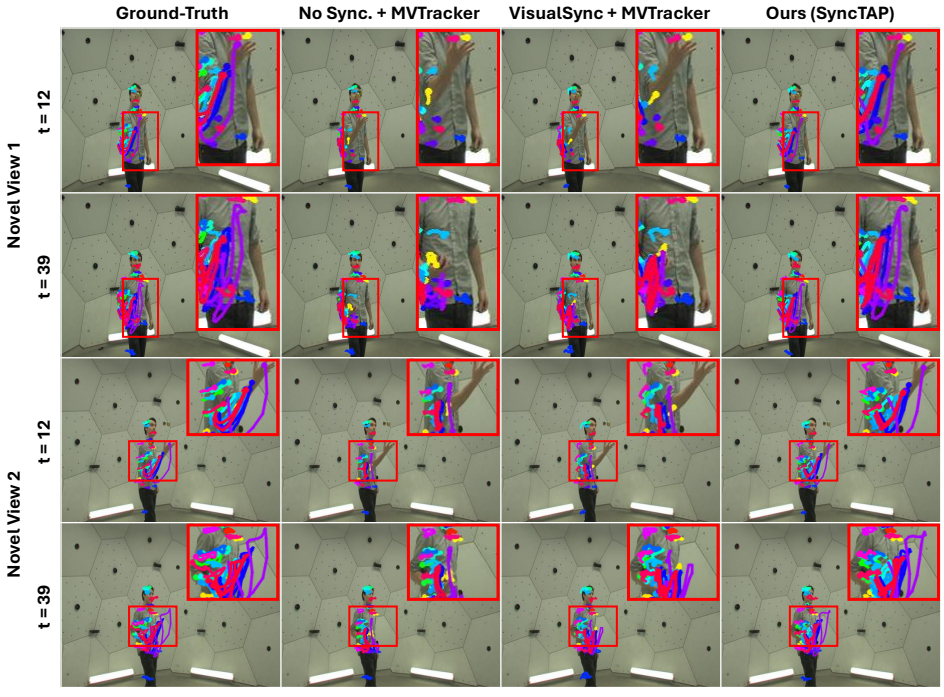

Figure 9: **Multi-view 3D point tracking for unsynchronized videos on the CMU Panoptic Studio dataset.** We visualize 3D trajectories by projecting them onto novel viewpoints unseen in 4-view inference inputs. Our SyncTAP robustly estimates high-fidelity 3D trajectories for highly dynamic object (e.g. hand with purple/blue trajectory).

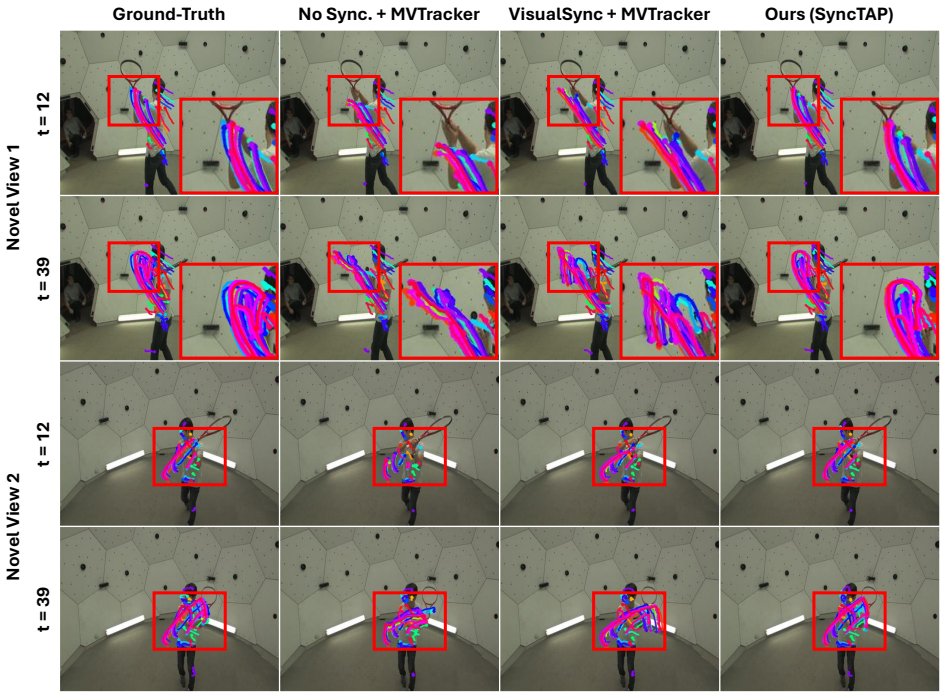

Figure 10: **Multi-view 3D point tracking for unsynchronized videos on the CMU Panoptic Studio dataset.** We visualize 3D trajectories by projecting them onto novel viewpoints unseen in 4-view inference inputs. Our SyncTAP robustly estimates high-fidelity 3D trajectories for highly dynamic object (e.g. tennis racket with pink trajectories).

F LIMITATIONS AND FUTURE WORK

While our SyncTAP achieves strong performance for joint 2D/3D tracking and synchronization under unsynchronized multi-view videos, several limitations remain. First, our framework assumes accurate camera parameters and depth maps as inputs, inherited from our backbones MV-TAP (Koo et al., 2026) and MVTracker (Rajič et al., 2025). Second, following MV-TAP (Koo et al., 2026), we assume that multi-view 2D queries are given across all views, while single-view query initialization remains an open problem. Some foundation models for feature matching (e.g., DINOv3 (Siméoni et al., 2025)) and multi-view consistent depth (e.g., DepthAnything 3 (Lin et al., 2025) and MapAnything (Keetha et al., 2026)) can provide reasonable initial estimates of multi-view queries that our framework would refine. As shown in Appendix D.4, it is proven that our SyncTAP also robustly works for human pose estimated multi-view queries. Finally, we model per-view sub-frame time offsets as scalar values, which do not reflect clock drift and rolling-shutter effects. Importantly, given unsynchronized multi-view input videos, the main novelty of our SyncTAP is to bridge the gap between time-offset estimation and dynamic multi-view 2D/3D tracking via an unified framework that utilizes pretrained 2D trackers and epipolar constraint (with 2nd-order TSA) as geometric cues. For future work, we plan to relax these input constraints by jointly estimating camera poses, depths, query correspondences, time-varying offsets, and trajectories directly from unsynchronized multi-view input videos.