TrackFish3D: Self-Supervised 3D Tracking of Schooling Fish from Multi-view Videos

Patt Phurtivilai[1], **Zhiyang Dou**[1,†], **Yifan Wu**[1], **Kinfung Chu**[4]
Yuan Liu[2], **Lei Yang**[1], **Wenping Wang**[3], **Taku Komura**[1,†]

[1] The University of Hong Kong, Hong Kong, China
[2] Hong Kong University of Science and Technology, Hong Kong, China
[3] Texas A&M University, College Station, TX, US
[4] Centre for Transformative Garment Production

Abstract

Quantifying collective fish behavior requires accurate trajectories, yet multi-view 3D tracking remains challenging due to frequent occlusions, visually similar individuals, and the long-standing scarcity of identity annotations. We present TrackFish3D, a geometry-driven self-supervised framework for dense multi-camera 3D tracking of schooling fish. Instead of relying on appearance-based re-identification or manually annotated identities, TrackFish3D turns calibrated multi-view geometry into supervision: triangulation and reprojection consistency provide pseudo-associations, while a geometric encoder and global association transformer learn all-to-all cross-view correspondence within each frame. To make these associations identity-aware, TrackFish3D introduces a self-supervised contrastive objective that separates co-visible individuals in the embedding space, together with a temporal predictor that preserves identities and bridges short occlusions across frames. The resulting model is trained once on unlabeled footage and applied directly to unseen test videos, requiring no cross-view identity labels, temporal annotations, 3D ground truth, appearance features, or test-time optimization. On our benchmark, TrackFish3D improves 3D Multi-Object Tracking Accuracy from 87.7% for the strongest baseline to 95.8%. On the 3D-ZeF zebrafish benchmark, it achieves 81.1% MOTA, compared with 77.4% for the best geometric baseline. TrackFish3D also generalizes beyond fish, achieving strong results on real-world bird tracking. Project page.

1 Introduction

Collective animal behaviour, from fish schools to bird flocks and insect swarms, emerges from local interaction rules and is central to behavioural ecology and computational biology [1, 11, 16, 52, 57]. Understanding these rules informs models of decision-making, predator evasion, and swarm robotics [15, 17, 26, 28, 41, 61]. Among them, *schooling fish* provide a tractable system because their group structure and interactions vary with density, group size, and environmental context [8, 16, 17, 31, 60]. Yet naturalistic 3D schooling remains incompletely understood, requiring dense 3D trajectories for every individual over time [3, 10, 18, 29]. Vision is the only scalable, non-invasive way to recover such trajectories, while 2D analysis



Fig. 1: Dense fish school in the wild share a nearly identical appearance with mutual occlusion.[1]

[†]Corresponding authors.
[1]Image sources: Unsplash (`unsplash.com`); Freepik (`freepik.com`).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

is insufficient for mechanistic modelling of spatial organisation and interactions [8, 28, 48, 49, 53].

Recovering dense *3D trajectories* from video therefore remains a central bottleneck. A practical system must solve two coupled problems: *spatial association*, which fuses multi-view 2D observations into 3D positions, and *temporal association*, which links these positions into identity-consistent trajectories. For schooling fish, both are especially difficult because supervision is scarce and appearance cues are weak: cross-view identities, temporal annotations, and 3D ground truth are rarely available, while individuals look nearly identical and frequently occlude one another (Fig. 1) [20, 30, 44].

Existing vision approaches only partially address these requirements. *Single-view multi-object trackers* (SVMOT) [6, 22, 33, 38, 54] model temporal association within one camera, but cannot fuse observations across views. *Multi-view association* (MVA) methods solve the complementary spatial problem by matching detections across synchronized cameras within each frame, but do not model temporal continuity; supervised variants such as MessyTable [7] and ViTP3DE [55] require cross-view identity labels, while self-supervised variants such as Self-MVA [12] and MvMHAT [21, 58] still rely on visual features. *Multi-view multi-object trackers* (MVMOT) such as ReST [13], MCTR [43], LMGP [42], and DyGLIP [50] jointly address spatial and temporal association, but require identity supervision for training. Thus, existing method families still depend on appearance features and/or identity supervision (Tab. 1)—precisely the resources unavailable for dense fish schools.

Tab. 1: Method families for dense 3D tracking. *Temp./Spat.*: temporal/spatial association. The central columns indicate whether a family can operate without cross-view identity labels (No XV-ID), temporal identity labels (No T-ID), or 3D ground-truth trajectories (No 3D GT). *No Vis.*: no visual appearance features.

			No labels / GT required			
	Temp.	Spat.	No XV-ID	No T-ID	No 3D GT	No Vis.
SVMOT	✓	✗	✓	✗	✓	✗
Supervised MVA	✗	✓	✗	✓	✓	✗
Self-supervised MVA	✗	✓	✓	✓	✓	✗
MVMOT	✓	✓	✗	✗	✓	✗
TrackFish3D	✓	✓	✓	✓	✓	✓

In this paper, we propose **TrackFish3D**, a self-supervised framework for the 3D association and tracking of visually homogeneous fish schools (Fig. 2). Our key insight is that, when individuals are nearly indistinguishable in appearance, reliable identity cues should come not from visual re-identification, but from the geometric and temporal consistency induced by calibrated multi-view observations. Accordingly, TrackFish3D learns both cross-view correspondence and temporal identity consistency directly from multi-view geometry. Within each frame, it uses geometric cues to associate detections across cameras, triangulate candidate 3D hypotheses, and learn an embedding space in which detections of the same individual are pulled together while different individuals are separated. Across frames, it further learns temporal predictions that propagate identity embeddings through dense overlap and occlusions, reducing identity drift when instantaneous geometric matching becomes ambiguous. Triangulation and reprojection consistency provide the training signal throughout, turning multi-view consistency into supervision and eliminating the need for cross-view identity labels, temporal identity annotations, appearance-based re-identification, or ground-truth trajectories. Here, self-supervised refers specifically to the 3D association and tracking stage.

To facilitate evaluation, we introduce **SynFish**, a realistic Unreal Engine 5 synthetic benchmark with dense ground-truth 3D trajectories for visually homogeneous fish schools. We further evaluate on the real-world 3D-ZeF20 zebrafish benchmark [47]. Under a unified protocol that converts representative baselines into end-to-end 3D tracking pipelines, TrackFish3D outperforms all methods on both datasets, achieving higher 3D tracking accuracy and fewer identity switches.

In summary, our main contributions are:

- A self-supervised multi-camera 3D tracking framework **TrackFish3D** that jointly learns cross-view association and temporal identity consistency from calibrated multi-view geometry, requiring no cross-view identity labels, temporal identity annotations, appearance-based re-identification, or ground-truth 3D trajectories.
- **SynFish**, a realistic synthetic benchmark rendered in Unreal Engine 5, providing dense ground-truth 3D trajectories for visually homogeneous fish schools across varying group sizes, occlusion levels, and camera configurations.

2 Related Work

Dense multi-camera 3D tracking requires both *spatial association* across views and *temporal association* across frames. Existing methods address only part of this problem or rely on supervision and appearance cues, as summarized in Tab. 1.

Multi-view association from 2D detections. Given per-camera detections, classical multi-view association matches observations across cameras and triangulates them into 3D [24]. These methods typically solve bipartite matching with Hungarian [34] or greedy algorithms using epipolar and reprojection costs [23], but become fragile in dense scenes with occlusions, missed detections, and localization noise. Learned MVA methods replace hand-crafted costs with affinity models: supervised methods such as MessyTable [7] and ViTP3DE [55] require cross-view identity labels, while self-supervised methods such as Self-MVA [12], MvMHAT [21], and Vo et al. [58] reduce annotation needs. However, MVA methods only solve *spatial* association within each frame and still rely on visual features, leaving temporal identity consistency to downstream heuristics; they do not learn a full 3D tracking pipeline over time.

Single-view tracking followed by 3D lifting. Single-view multi-object trackers (SVMOT), including MeMOT [6], TrackFormer [38], CoTracker [33], SambaMOTR [54], and MOTIP [22], model temporal association within one camera using appearance, motion, or long-range tracklet cues. To obtain 3D trajectories, their per-view tracks must still be associated across cameras. For visually interchangeable animals, this produces fragmented 2D tracklets and reintroduces the same cross-view ambiguity faced by MVA methods.

Joint multi-view multi-object tracking. MVMOT methods such as TRACTA [25], DyGLIP [50], LMGP [42], ReST [13], and MCTR [43] jointly model spatial and temporal association using cross-camera re-identification with graph or transformer architectures. They are closest to full multi-camera tracking, but typically require identity supervision and discriminative appearance features. ADA-Track++ [19] further performs end-to-end multi-camera 3D tracking, but assumes 3D object detections and targets autonomous-driving scenes with visually distinct objects.

In contrast to these methods, TrackFish3D targets the underexplored intersection of geometry-only identity association and self-supervised temporal 3D tracking for visually homogeneous animals by jointly learning cross-view association and temporal identity consistency from calibrated multi-view geometry, without identity labels, appearance-based re-identification, or ground-truth 3D trajectories.

3 Method

TrackFish3D addresses dense multi-camera 3D multi-object tracking of schooling fish from synchronized videos captured by C calibrated cameras. At each time step t, each camera provides 2D instance detections, each with a bounding box, an instance centroid $p = (u, v)$, and a detector confidence score $s \in [0, 1]$; camera intrinsics K_c and extrinsics (R_c, t_c) are assumed known. TrackFish3D is agnostic to the upstream 2D perception model: it only requires per-frame, per-camera 2D detections, which can come from a detector–segmenter pipeline such as YOLO26+SAM2 [32, 51] or a promptable segmentation model such as SAM 3 [9]. It does not require temporal tracking output, clean detections, cross-view identity labels, or temporal identity labels from the 2D perception stage. In our implementation, each per-camera, per-frame 2D instance is represented by a centroid, which is derived from an instance segmentation mask. Overall, the method only assumes per-camera, per-frame 2D detections and calibrated cameras as input.

As illustrated in Fig. 2, the 3D association and tracking stage of TrackFish3D reconstructs trajectories self-supervisedly by learning three coupled components: cross-view association within each frame, embedding-space identity separation across views, and temporal identity consistency across frames. The complete training procedure is given in Algorithm 1 (Appendix A). Crucially, TrackFish3D requires no cross-view identity annotations, no temporal identity annotations, and no ground-truth 3D trajectories.

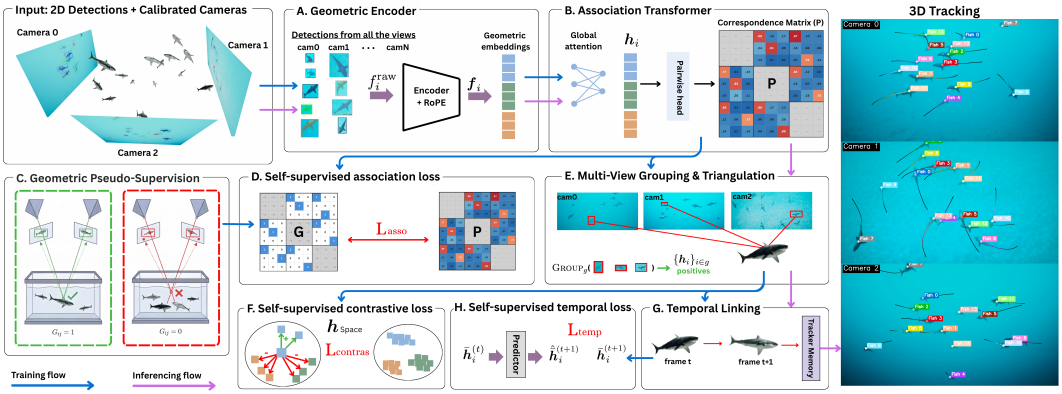


Fig. 2: Overview of the TrackFish3D framework. Synchronised multi-view video is processed by (A) a *Geometric Encoder* and (B) an *Association Transformer* that predicts cross-view pairwise scores, supervised by (C) *Geometric Pseudo-Supervision* from triangulation consistency through (D) a *Self-Supervised Association Loss*. Detections are fused via (E) *Multi-View Grouping & Validation* to produce 3D hypotheses, whose embeddings are separated by (F) a *Self-Supervised Contrastive Loss*. Across frames, (G) *Temporal Linking* matches identities and (H) a *Self-Supervised Temporal Loss* enforces cross-frame consistency, enabling stable 3D trajectory recovery at inference.

3.1 Geometric encoder

Each raw detection feature, comprising the 2D centroid, bounding-box size, camera index, confidence score, and a back-projected world ray, is encoded by a lightweight MLP into a compact embedding that captures multi-view geometric constraints while remaining entirely agnostic to visual appearance.

Concretely, for a detection with mask-derived centroid $p = (u, v)$, bounding box size (w, h), camera index c, and detector confidence $s \in [0, 1]$, we construct a raw feature vector: $f_i^{\text{raw}} = [\tilde{u}_i, \tilde{v}_i, \tilde{w}_i, \tilde{h}_i, \tilde{c}_i, s_i, r_{w,x}, r_{w,y}, r_{w,z}]$, where centroid coordinates are divided by image width and height respectively, bounding-box dimensions are similarly normalised by the corresponding image dimension, and the camera index is divided by $(C-1)$, mapping all 2D quantities to approximately $[0, 1]$. The components $(r_{w,x}, r_{w,y}, r_{w,z})$ denote the unit world-ray direction obtained by back-projecting the centroid through the camera intrinsics and rotating into the world frame. The geometric encoder ϕ, a lightweight two-layer MLP with ReLU [40] and LayerNorm [2] layers, maps this raw feature into a d=128-dimensional embedding: $\boldsymbol{f}_i = \phi(f_i^{\text{raw}}) \in \mathbb{R}^d$. This representation encodes geometric constraints while remaining agnostic to appearance.

3.2 Association transformer

For each frame, the encoder outputs $\{\boldsymbol{f}_i\}$ are processed jointly by an association transformer ψ, producing context-enriched embeddings $\{\boldsymbol{h}_i\}$. We adopt a *global* self-attention design over all N detections from all C cameras in a frame, similar in spirit to prior geometric token processing in VGGT [59], rather than applying cross-attention between camera pairs. Detections from all cameras are processed jointly so that correspondence decisions can exploit global cross-view context within each frame. This global self-attention design captures all-to-all relationships in one pass, supports transitive reasoning across cameras, and avoids separate pair-specific computations. We inject 2D spatial information using rotary positional embeddings (RoPE) [27] derived from normalised centroid coordinates, which encode relative spatial structure without fixed resolution assumptions. This scene-level context is especially important in dense scenes, where multiple cross-view candidates may satisfy similar pairwise geometric costs. Rather than deciding each match independently, the transformer conditions each detection on the full set of observations in the frame before multi-view grouping, helping disambiguate geometrically plausible but incorrect matches.

A pairwise association head estimates whether detections from different cameras correspond to the same individual. For each ordered pair of detections (i, j) from different cameras, the head outputs a probability $P_{ij} \in (0, 1)$ indicating whether the two detections correspond to the same fish. Pairs from the same camera are ignored by construction.

Concretely, let $\boldsymbol{x}_i = \psi_{\text{proj}}(\boldsymbol{f}_i)$ denote the projected token for detection i, and let $\boldsymbol{p}_i = (c_{y,i},\, c_{x,i}) \in [0, 1]^2$ be its normalised centroid. In the implementation, RoPE is applied to the projected token

before the transformer encoder layers: $\boldsymbol{x}'_i = R(\boldsymbol{p}_i)\,\boldsymbol{x}_i$, where the vertical and horizontal centroid axes are encoded independently in separate halves of the embedding dimension. The rotated tokens $\{\boldsymbol{x}'_i\}$ are then processed by the standard transformer self-attention stack to produce contextualised embeddings $\{\boldsymbol{h}_i\}$. Because the rotary transform injects position information in a relative, parameter-free manner, the model can distinguish spatially nearby detections from distant ones without a fixed learned positional embedding table, which is particularly important for disambiguating nearby fish in dense scenes.

3.3 Geometric pseudo-supervision

Because cross-view identity annotations are unavailable, TrackFish3D generates pseudo-labels through multi-view geometric consistency. The contribution here is not a new triangulation primitive, but using calibrated multi-view geometry as supervision for cross-view association without identity labels. For each pair of detections from different cameras within the same timestamp , we compute a tentative 3D point by triangulation [23] and reproject it back into the image planes. If two detections correspond to the same fish, triangulation is geometrically consistent and reprojections align with the original observations. Based on this geometric consistency criterion, we construct a binary pseudo ground-truth association matrix $G \in \{0,1\}^{N\times N}$. For a cross-camera detection pair (i,j), we assign $G_{ij} = 1$ if the triangulated point reprojects inside both bounding boxes, and $G_{ij} = 0$ otherwise. Pairs from the same camera and self-pairs are excluded from supervision and masked out during training.

To account for geometric noise, we additionally compute a confidence score for positive pairs based on reprojection error $C_{ij} = \frac{1}{1+\mathrm{err}_{ij}}$, where err_{ij} denotes the average reprojection error *in pixels* across the two views (pseudo-GT computation operates in raw pixel coordinates; the normalisation described in Section 3.1 applies only inside the feature encoder). This confidence modulates the strength of supervision without changing the binary nature of the pseudo labels.

3.4 Self-supervised association loss

The association loss supervises the pairwise correspondence head using the geometry-derived binary pseudo labels. For each frame, we consider ordered detection pairs (i,j) such that $i \neq j$ and $c_i \neq c_j$. Pairs excluded during pseudo-GT construction are masked out and do not contribute to the loss.

Let $P_{ij} \in (0,1)$ denote the predicted correspondence probability and $G_{ij} \in \{0,1\}$ the pseudo label. We apply a confidence-weighted binary cross-entropy loss in which positive and negative terms are normalised independently: $\mathcal{L}_{\text{asso}} = -\frac{\sum_{G_{ij}=1} C_{ij}\,\log P_{ij}}{\sum_{G_{ij}=1} C_{ij}} - \frac{\sum_{G_{ij}=0}\log(1-P_{ij})}{|\{G_{ij}=0\}|}$, where C_{ij} is the pseudo-GT confidence (Section 3.3). Positive pairs are weighted by their confidence and averaged by the total confidence mass, while negative pairs receive uniform weight and are averaged by count. This independent normalisation prevents negative-pair domination when positive pairs are scarce, ensuring balanced gradient contributions regardless of the positive-to-negative ratio.

This loss is applied independently to frames t and $t{+}1$ during training. It encourages high correspondence probability for geometrically consistent cross-view pairs and low probability otherwise, providing a self-supervised signal for learning multi-view association without identity labels or ground-truth 3D trajectories.

3.5 Multi-view grouping & triangulation

Predicted correspondence probabilities are converted into multi-view fish hypotheses through a compact grouping pipeline. For cross-camera pairs, we combine learned correspondence and embedding agreement with $A_{ij} = w_p\,P_{ij} + w_f\,\frac{\cos(\boldsymbol{h}_i,\boldsymbol{h}_j)+1}{2}$, then apply Hungarian matching within each camera pair, merge accepted matches transitively across cameras, and triangulate each candidate group. Groups are kept only when the triangulated 3D point reprojects consistently into the supporting detections. For each valid group, we compute the group embedding $\bar{\boldsymbol{h}}_g = \ell_2\text{-norm}\big(\frac{1}{|g|}\sum_{i\in g}\boldsymbol{h}_i\big)$, During training, grouping is not a supervision target. Instead, it produces two outputs used downstream: the individual member embeddings $\{\boldsymbol{h}_i\}_{i\in g}$, which define positive pairs for $\mathcal{L}_{\text{ctr}}$ (Section 3.6), and the group embedding $\bar{\boldsymbol{h}}_g$, which serves as the per-fish identity token for temporal linking (Section 3.7). Full grouping, thresholding, and geometric validation details are provided in Appendix M.1.

3.6 Self-supervised contrastive loss

Within each frame, detections from the same multi-view group are treated as positives and all others as negatives. We apply an InfoNCE-style contrastive loss [45] to the normalised transformer embeddings $\boldsymbol{z}_i = \boldsymbol{h}_i/\|\boldsymbol{h}_i\|$, with similarity $S_{ij} = \boldsymbol{z}_i^\top \boldsymbol{z}_j/\tau$. For each anchor detection i with positive set $\mathcal{P}_i$, the loss is: $\mathcal{L}_{\text{ctr}}^{(i)} = -\log \frac{\sum_{j\in\mathcal{P}_i} \exp(S_{ij})}{\sum_{\substack{k=1\\k\neq i}}^{N} \exp(S_{ik})}$, and the frame-level loss averages over anchors with at least one positive. This objective pulls same-fish detections together across views and pushes different identities apart, improving grouping in dense scenes (Fig. 3). Full positive-mining and implementation details are provided in Appendix M.2.

3.7 Temporal linking and self-supervised temporal loss

To encourage identity stability over time, TrackFish3D links triangulated group hypotheses between consecutive frames and trains a temporal predictor on the matched group embeddings. Given groups from frames t and $t{+}1$ (each with a 3D position $\mathbf{X}$ and a group embedding $\bar{\boldsymbol{h}}$), we construct the matching cost $\text{cost}(i,j) = \alpha \frac{\|\mathbf{X}_j^{(t+1)} - \mathbf{X}_i^{(t)}\|}{d_{\max}} + \beta\left(1 - \cos(\bar{\boldsymbol{h}}_i^{(t)}, \bar{\boldsymbol{h}}_j^{(t+1)})\right)$, where $\alpha{=}0.6$ and $\beta{=}0.4$ balance 3D proximity and embedding similarity. Hungarian assignment [34] is applied with distance and similarity gates, and each accepted match receives a weight $w_{ij} = \min(\text{conf}_i, \text{conf}_j)$. For each matched pair (i,j), the predictor takes the observed group embedding $\bar{\boldsymbol{h}}_i^{(t)}$ and predicts $\hat{\bar{\boldsymbol{h}}}_j^{(t+1)} = \rho(\bar{\boldsymbol{h}}_i^{(t)})$. The temporal loss is a weight-normalized mean squared error: $\mathcal{L}_{\text{temp}} = \frac{\sum_{(i,j)} w_{ij} \|\hat{\bar{\boldsymbol{h}}}_j - \bar{\boldsymbol{h}}_j^{(t+1)}\|^2}{\sum_{(i,j)} w_{ij}}$, which encourages identity-consistent embeddings across time and improves linking under occlusion (Fig. 4). Full fallback matching, predictor, and training details are provided in Appendix M.3.

3.8 Training and Inference

The total training loss is a weighted combination of the three complementary objectives:

$$\mathcal{L} = \mathcal{L}_{\text{asso}} + \lambda_{\text{ctr}}\,\mathcal{L}_{\text{ctr}} + \lambda_{\text{temp}}\,\mathcal{L}_{\text{temp}}.$$

Training begins with the association loss alone and gradually introduces the contrastive and temporal terms to stabilize optimization. Each component targets a distinct failure mode: $\mathcal{L}_{\text{asso}}$ resolves cross-view ambiguity, $\mathcal{L}_{\text{ctr}}$ reduces within-frame identity confusion, and $\mathcal{L}_{\text{temp}}$ prevents temporal identity drift. No component requires identity annotations or ground-truth 3D data; all supervision is derived from multi-view geometric consistency.

At inference, TrackFish3D reconstructs per-frame 3D hypotheses using the learned association scores P_{ij} and embeddings, then links them over time with the same distance–similarity cost and the learned predictor ρ. Missing tracks are bridged by rolling embeddings forward with ρ, and non-overlapping fragments are stitched using embedding similarity and 3D proximity. Optional interpolation and short-track removal are applied only as post-processing. Thus, inference is a direct continuation of training and requires no identity labels, appearance re-identification, or privileged scene information. Full tracker and stitching details are provided in Appendix M.4.

4 Experimental Results

4.1 Datasets and difficulty characterisation

We evaluate TrackFish3D on the synthetic **SynFish** benchmark and the real-world zebrafish dataset 3D-ZeF [47]. All datasets provide calibrated cameras; SynFish additionally provides ground-truth 3D trajectories for quantitative evaluation.

SynFish dataset. SynFish contains 14 scenes with 4–16 visually near-indistinguishable fish, rendered at $1920{\times}1080$ with $C{=}3$ synchronized cameras and 7 camera-rig configurations. We train on 8 scenes and evaluate on 6 held-out scenes, including one unseen rig (Appendix F).

Real-world zebrafish (3D-ZeF). 3D-ZeF [47] provides calibrated two-camera real-world sequences at $2704{\times}1520$ resolution with 2 or 5 fish. We use zebra1 and zebra3 for training and zebra2 and zebra4 for testing.

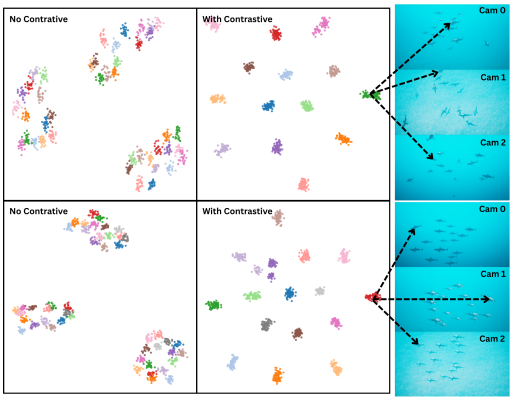


Fig. 3: Effect of the contrastive objective on cross-view embeddings. **Left:** Without contrastive training, same-fish detections overlap other identities. **Right:** With contrastive training, they form compact, well-separated clusters.

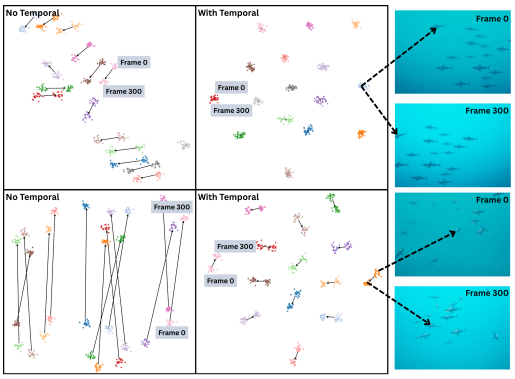


Fig. 4: Effect of the temporal prediction objective on cross-frame identity consistency. **Left:** Without temporal training, same-fish embeddings drift across frames and cause identity switches. **Right:** With temporal training, they stay close across frames for stable linking.

To characterise difficulty beyond fish count, we report an occlusion score (OccScore), defined as the average maximum pairwise 2D bounding-box IoU over frames and views, together with %Overlap, the percentage of frame-view instances containing any overlap (Appendix Table 8). Camera calibration parameters are listed in Appendix F. Please refer to Appendix N.1–N.3 for more details.

4.2 Baselines, Protocol, and Metrics

No prior method solves this setting end-to-end without identity supervision, so we compare against 13 complete 3D tracking pipelines spanning four families. Detection-based baselines combine SAM 3 [9] or YOLO26+SAM2 [32, 51] detections with geometric MVA (Hungarian [34] or greedy) or learned MVA (*Self-MVA* [12], *ASNet* [7]), followed by triangulation and nearest-neighbour 3D linking; **3D-SORT** instead adds constant-velocity Kalman tracking [4] after SAM3 + Hungarian (Appendix Q). We also evaluate single-view MOT methods, *SambaMOTR* [54] and *MOTIP* [22], lifted to 3D via first-frame cross-view association, and multi-camera MOT methods, *ReST* [13] and *MCTR* [43], whose cross-view-consistent 2D tracks are triangulated directly.

For fairness, all external components are finetuned or retrained on the SynFish and 3D-ZeF training splits with default configurations, while TrackFish3D is trained self-supervisedly on the same splits using SAM 3 or YOLO26+SAM2 detections. All pipelines use identical detections, camera calibrations, and post-processing, including short-gap interpolation between already-associated observations and removal of short-lived ghost tracks; these steps use no privileged information and introduce no new identities. We evaluate reconstructed trajectories using standard 3D MOT metrics: MOTA, mostly-tracked (MT%), mostly-lost (ML%), identity switches (IDSw), fragmentations (Frag), and mean time between failures (MTBF). Predicted 3D points are matched to ground truth under a fixed 3D distance threshold at each frame, from which trajectory-level metrics are computed.

4.3 Results on SynFish test set

Table 2 presents 3D tracking results averaged over the 6 held-out SynFish test scenes (4–16 fish). Per-scene breakdowns are provided in Appendix Table 12.

Across six held-out SynFish scenes, TrackFish3D achieves the best overall performance, reaching 96.6% MOTA with YOLO26+SAM2 inputs and 95.8% with SAM3, while also obtaining the highest MT% (95.8%) and strongest identity stability (IDSw 0.0–0.2, MTBF 266–297 frames). It consistently outperforms all baselines, including supervised methods retrained on the same data. The strongest baseline, 3D-SORT, reaches 87.7% MOTA; *MCTR* reaches 83.7%, and direct geometric MVA pipelines achieve 82–84%, while TrackFish3D remains above 86% on dense 16-fish scenes. In contrast, appearance-driven and single-view trackers fail to transfer reliably to visually homogeneous fish. Fig. 5 further shows that TrackFish3D degrades gracefully with fish density, whereas baselines drop sharply beyond 8 fish.

Tab. 2: 3D tracking results on the SynFish test set, averaged over 6 held-out scenes (4–16 fish). Best result in each column among methods with MOTA>0 is **bolded**; per-scene breakdown is in Appendix Table 12.

Method	MOTA↑	MT%↑	ML%↓	IDSw↓	Frag↓	MTBF↑
Detection + geometric MVA + 3D nearest-neighbour linking						
SAM3 + Hungarian	83.5%	71.9%	0.0%	7.5	12.8	183.8
SAM3 + Greedy	83.5%	71.9%	0.0%	7.5	12.8	183.8
SAM3 + 3D-SORT	87.7%	77.1%	0.0%	4.0	9.0	209.4
YOLO26+SAM2 + Hungarian	82.2%	69.8%	0.0%	8.7	12.2	183.1
YOLO26+SAM2 + Greedy	82.2%	69.8%	0.0%	8.7	12.2	183.1
Detection + learned MVA + 3D nearest-neighbour linking						
Self-MVA [12]	−64.6%	0.0%	96.9%	1.7	2.2	15.9
ASNet [7]	33.3%	49.6%	11.5%	20.8	29.0	60.7
Single-view 2D MOT → MVA (first frame) → 3D						
SambaMOTR [54]	−10.5%	11.7%	73.8%	0.0	7.8	51.8
MOTIP [22]	14.7%	21.4%	42.9%	0.0	25.7	41.2
Multi-camera MOT → triangulate						
ReST [13]	61.3%	74.8%	0.0%	173.7	78.5	17.5
MCTR [43]	83.7%	82.7%	0.0%	1.7	14.2	160.6
Ours (self-supervised 3D tracking)						
TrackFish3D (SAM3)	95.8%	**95.8%**	**0.0%**	**0.0**	6.5	266.3
TrackFish3D (YOLO26+SAM2)	**96.6%**	**95.8%**	**0.0%**	0.2	**3.7**	**296.5**

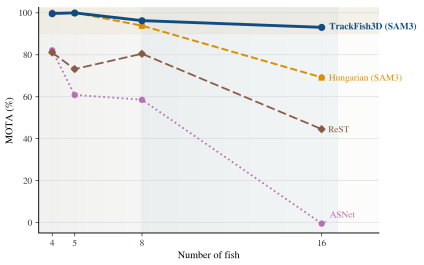


Fig. 5: MOTA vs. fish count on SynFish test scenes. TrackFish3D degrades gracefully with density, whereas baselines drop sharply beyond 8 fish.

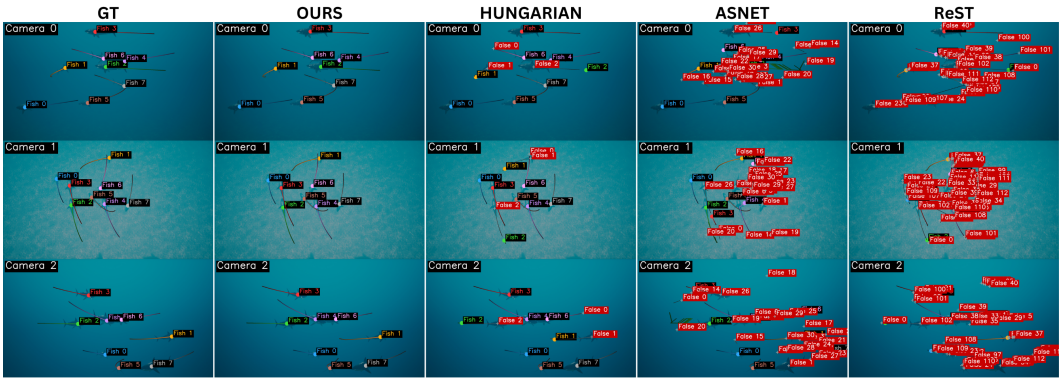


Fig. 6: Qualitative 3D trajectory comparison on SynFish test scenes. Columns: ground truth, TrackFish3D, and representative baselines. Colour encodes fish identity; trajectory breaks indicate identity switches or fragmentations.

4.4 Results on 3D-ZeF test set

Tab. 3 reports results on the real 3D-ZeF test set, where TrackFish3D is trained self-supervisedly on zebra1/3 and evaluated on zebra2/4 using the same geometric objectives as on SynFish; per-scene results are in Appendix Table 14. TrackFish3D achieves 81.1% MOTA, outperforming all baselines. The gap over geometric MVA baselines is modest in MOTA (77.4% for Hungarian, 76.9% for Greedy) because 3D-ZeF contains only 5 fish, making per-frame cross-view association relatively sparse.

The advantage is clearer in identity stability: TrackFish3D reduces fragmentation from 73–76 to 30.5, improves MTBF from 38–43 to 93.6 frames, and lowers ID switches from 6–10 to 4.0. This shows the benefit of learned temporal linking under real occlusions, where per-frame geometric matching alone is insufficient. Appearance-based baselines remain ineffective after retraining, with most producing near-zero or negative MOTA and Self-MVA reaching only 50.8%, again indicating that geometry is the more reliable cue for visually homogeneous animals.

4.5 Cross-Species Generalization to Real-World Pigeons

On the real-world Pigeon10 subset from 3D-POP [39], TrackFish3D achieves the best MOTA (91.9%), recall (94.5%), and fragmentation (4) among evaluated methods without bird identity labels. We emphasise that this subset mainly captures pigeons walking on the ground, with only limited flying in or out of view, rather than sustained 3D flocking. We therefore present it as preliminary evidence that the geometry-driven formulation can transfer beyond fish to another visually homogeneous animal setting, not as validation on fully 3D flying-bird tracking; evaluating on harder datasets with sustained bird flight is left for future work. Full setup and results are provided in Appendix P.

Tab. 3: 3D tracking results on the 3D-ZeF test set, averaged over 2 sequences (zebra2: 5 fish, zebra4: 5 fish). Best result in each column (among methods with MOTA>0) is **bolded**.

Method	MOTA ↑	MT% ↑	ML% ↓	IDSw ↓	Frag ↓	MTBF ↑
Hungarian + nearest	77.4%	**90.0%**	**0.0%**	6.0	73.0	42.5
Greedy + nearest	76.9%	**90.0%**	**0.0%**	10.0	76.0	38.7
3D-SORT	75.6%	**90.0%**	**0.0%**	13.5	65.5	43.4
Self-MVA [12]	50.8%	60.0%	**0.0%**	21.5	73.5	48.5
ASNet [7]	0.3%	0.0%	**0.0%**	72.5	175.0	10.6
SambaMOTR [54]	−26.4%	0.0%	100.0%	0.5	1.0	6.5
MOTIP [22]	−45.9%	0.0%	100.0%	1.5	6.5	2.7
ReST [13]	0.0%	0.0%	100.0%	0.0	0.0	0.0
MCTR [43]	0.1%	0.0%	40.0%	21.0	57.5	8.8
TrackFish3D	**81.1%**	**90.0%**	**0.0%**	**4.0**	**30.5**	**93.6**



Fig. 7: Qualitative 3D trajectory comparison on 3D-ZeF test sequences. TrackFish3D maintains identity consistency through real-world occlusions that fragment or swap baseline trajectories. ReST produces no trajectories because its cross-view association fails to match detections correctly.

4.6 Ablation Studies

We conduct ablation studies on SynFish to isolate (i) the contribution of each self-supervised loss component and (ii) the impact of visual appearance features. Results are reported in Tab. 4.

Tab. 4: Ablation studies on SynFish. *Top:* removing contrastive ($\mathcal{L}_{\text{ctr}}$) and/or temporal ($\mathcal{L}_{\text{temp}}$) losses. *Bottom:* concatenating DINOv3 [56] visual features with geometric features.

Variant	MOTA ↑	MT% ↑	ML% ↓	IDSw ↓	Frag ↓	MTBF ↑
Ablation 1: loss components						
TrackFish3D (full)	**97.7%**	**98.3%**	**0.0%**	**0.5**	**3.1**	**283.1**
w/o $\mathcal{L}_{\text{temp}}$	86.0%	85.2%	8.0%	2.0	5.2	243.1
w/o $\mathcal{L}_{\text{ctr}}$	36.8%	23.8%	36.4%	3.0	8.0	99.1
w/o both	36.9%	25.6%	34.2%	3.8	9.8	88.8
Ablation 2: visual features						
TrackFish3D (geometry only)	97.7%	98.3%	0.0%	**0.5**	**3.1**	**283.1**
TrackFish3D + DINOv3	**97.9%**	**99.2%**	0.0%	1.1	3.2	270.2

Different Loss Components. Removing the contrastive loss ($\mathcal{L}_{\text{ctr}}$) causes a significant drop from 97.7% to 36.8% MOTA, with ML% rising from 0% to 36.4%. Removing the temporal loss ($\mathcal{L}_{\text{temp}}$) reduces MOTA from 97.7% to 86.0% and roughly quadruples identity switches (0.5→2.0); spatial association still functions, but occlusion-bridging degrades noticeably, leading to higher fragmentation and lower MTBF. Removing both auxiliary losses yields 36.9% MOTA, nearly identical to removing $\mathcal{L}_{\text{ctr}}$ alone, indicating that without view-consistent identity separation, temporal prediction has little to anchor to.

Influence of Visual Features. Concatenating pretrained DINOv3 [56] visual features with the geometric features yields 97.9% MOTA versus 97.7% for geometry alone, a negligible difference. MTBF actually decreases slightly (270.2 vs. 283.1), and IDSw doubles (1.1 vs. 0.5), indicating no practical gain from adding appearance cues.

Controlled Association Transformer Ablation. To isolate the role of scene-level self-attention, we replace the Association Transformer with an MLP-only encoder while retaining the same training data and all three losses. Results are reported in Tab. 5.

Tab. 5: Controlled Association Transformer ablation on SynFish.

Variant	MOTA ↑	MT% ↑	IDSw ↓	Frag ↓	MTBF ↑
TrackFish3D (SAM3)	**95.8%**	**95.8%**	**0.0**	**6.5**	**266.3**
MLP-only + all losses	78.4%	64.2%	2.0	16.3	172.9
SAM3 + Hungarian	83.5%	71.9%	7.5	12.8	183.8

Removing global self-attention reduces MOTA by 17.4 percentage points, lowers mostly-tracked trajectories from 95.8% to 64.2%, and increases fragmentation from 6.5 to 16.3. The MLP-only model also falls below SAM3 + Hungarian in MOTA, indicating that independent token processing cannot substitute for scene-level context in dense scenes.

5 Conclusion

We presented TrackFish3D, a self-supervised framework for dense multi-camera 3D tracking that learns cross-view association and temporal identity linking from calibrated geometry alone, without identity labels, appearance-based re-identification, or ground-truth 3D trajectories. TrackFish3D achieves 96.6% MOTA on held-out SynFish scenes and 81.1% on real 3D-ZeF zebrafish, outperforming strong baselines while substantially improving identity stability. Ablations show that contrastive embedding learning is essential for separating co-visible individuals, and the recovered trajectories are accurate enough to support downstream abnormal-behaviour detection (Appendix O). The same formulation also transfers to real-world pigeons. Additional appendix experiments evaluate 30-fish scaling (Appendix R), audit pseudo-label noise (Appendix S), and test training-time calibration perturbations (Appendix T). The method nevertheless depends on calibration, multi-view visibility, and reliable detections. Future work could improve robustness to calibration errors and missed detections, and collect dense real-world multi-camera animal datasets with ground-truth 3D trajectories.

Acknowledgments and Disclosure of Funding

This work was partially funded by the Research Grants Council of Hong Kong (Ref: 17216826), the Innovation and Technology Commission of the HKSAR Government under the ITSP-Platform grant (Ref: ITS/335/23FP, ITS/469/24FP), and a contract research project with SoftBank Corp.

References

[1] Ichiro AOKI. A simulation study on the schooling mechanism in fish. *NIPPON SUISAN GAKKAISHI*, 48(8):1081–1088, 1982. doi: 10.2331/suisan.48.1081.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Michele Ballerini, Nicola Cabibbo, Raphael Candelier, Andrea Cavagna, Evaristo Cisbani, Irene Giardina, Vivien Lecomte, Alberto Orlandi, Giorgio Parisi, Andrea Procaccini, et al. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the national academy of sciences*, 105(4):1232–1237, 2008.

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. Ieee, 2016.

[5] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[6] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8080–8090. IEEE, 2022.

[7] Zhongang Cai, Junzhe Zhang, Daxuan Ren, Cunjun Yu, Haiyu Zhao, Shuai Yi, Chai Kiat Yeo, and Chen Change Loy. Messytable: Instance association in multiple camera views. In *European Conference on Computer Vision*, pages 1–16. Springer, 2020.

[8] Daniel S Calovi, Ugo Lopez, Paul Schuhmacher, Hugues Chaté, Clément Sire, and Guy Theraulaz. Collective response to perturbations in a data-driven fish school model. *Journal of The Royal Society Interface*, 12(104):20141362, 2015.

[9] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pages 138846–138923, 2026.

[10] Andrea Cavagna, Alessio Cimarelli, Irene Giardina, Giorgio Parisi, Raffaele Santagati, Fabio Stefanini, and Massimiliano Viale. Scale-free correlations in starling flocks. *Proceedings of the National Academy of Sciences*, 107(26):11865–11870, 2010.

[11] Andrea Cavagna, Irene Giardina, and Tomás S Grigera. The physics of flocking: Correlation as a compass from experiments to theory. *Physics Reports*, 728:1–62, 2018.

[12] Keqi Chen, Vinkle Srivastav, Didier Mutter, and Nicolas Padoy. Learning from synchronization: self-supervised uncalibrated multi-view person association in challenging scenes. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24419–24428. IEEE, 2025.

[13] Cheng-Che Cheng, Min-Xuan Qiu, Chen-Kuo Chiang, and Shang-Hong Lai. Rest: A reconfigurable spatial-temporal graph model for multi-camera multi-object tracking. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10017–10026. IEEE, 2023.

[14] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *International symposium on neural networks*, pages 189–196. Springer, 2017.

[15] Soon-Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A survey on aerial swarm robotics. *IEEE Transactions on robotics*, 34(4):837–855, 2018.

[16] Iain D Couzin, Jens Krause, Richard James, Graeme D Ruxton, and Nigel R Franks. Collective memory and spatial sorting in animal groups. *Journal of theoretical biology*, 218(1):1–11, 2002.

[17] Iain D Couzin, Jens Krause, Nigel R Franks, and Simon A Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, 2005.

[18] Anthony I Dell, John A Bender, Kristin Branson, Iain D Couzin, Gonzalo G de Polavieja, Lucas PJJ Noldus, Alfonso Pérez-Escudero, Pietro Perona, Andrew D Straw, Martin Wikelski, et al. Automated image-based tracking and its application in ecology. *Trends in ecology & evolution*, 29(7):417–428, 2014.

[19] Shuxiao Ding, Lukas Schneider, Marius Cordts, and Juergen Gall. Ada-track++: End-to-end multi-camera 3d multi-object tracking with alternating detection and association. *IEEE transactions on pattern analysis and machine intelligence*, 48(1):482—499, January 2026. ISSN 0162-8828. doi: 10.1109/tpami.2025.3613269. URL `https://doi.org/10.1109/TPAMI.2025.3613269`.

[20] Audrey Filella, François Nadal, Clément Sire, Eva Kanso, and Christophe Eloy. Model of collective fish behavior with hydrodynamic interactions. *Physical review letters*, 120(19): 198101, 2018.

[21] Yiyang Gan, Ruize Han, Liqiang Yin, Wei Feng, and Song Wang. Self-supervised multi-view multi-human association and tracking. In *Proceedings of the 29th ACM international conference on multimedia*, pages 282–290, 2021.

[22] Ruopeng Gao, Ji Qi, and Limin Wang. Multiple object tracking as id prediction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27883–27893. IEEE, 2025.

[23] Richard Hartley, Andrew Zisserman, et al. *Multiple view geometry in computer vision*, volume 2. Cambridge university press Cambridge, 2003.

[24] Richard I Hartley and Peter Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[25] Yuhang He, Xing Wei, Xiaopeng Hong, Weiwei Shi, and Yihong Gong. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE Transactions on Image Processing*, 29: 5191–5205, 2020. doi: 10.1109/TIP.2020.2980070.

[26] Conor Heins, Beren Millidge, Lancelot Da Costa, Richard P Mann, Karl J Friston, and Iain D Couzin. Collective behavior from surprise minimization. *Proceedings of the National Academy of Sciences*, 121(17):e2320239121, 2024.

[27] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.

[28] James E Herbert-Read, Andrea Perna, Richard P Mann, Timothy M Schaerf, David JT Sumpter, and Ashley JW Ward. Inferring the rules of interaction of shoaling fish. *Proceedings of the National Academy of Sciences*, 108(46):18726–18731, 2011.

[29] Hans A Hofmann, Annaliese K Beery, Daniel T Blumstein, Iain D Couzin, Ryan L Earley, Loren D Hayes, Peter L Hurd, Eileen A Lacey, Steven M Phelps, Nancy G Solomon, et al. An evolutionary framework for studying mechanisms of social behavior. *Trends in ecology & evolution*, 29(10):581–589, 2014.

[30] Yaroslav Ispolatov. Computing in fish schools. *ELife*, 5:e12852, 2016.

[31] Mingjie Jiang, Anyu Zhou, Runping Chen, Yuqin Yang, Hao Dong, and Wei Wang. Collective motions of fish originate from balanced local perceptual interactions and individual stochastics. *Physical Review E*, 107(2):024411, 2023.

[32] Glenn Jocher, Jing Qiu, Mengyu Liu, Shuai Lyu, Fatih Cagatay Akyon, and Muhammet Esat Kalfaoglu. Ultralytics yolo26: unified real-time end-to-end vision models. *arXiv preprint arXiv:2606.03748*, 2026.

[33] 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*, pages 18–35. Springer, 2024.

[34] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[35] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng. Anomaly detection with generative adversarial networks for multivariate time series. *arXiv preprint arXiv:1809.04758*, 2018.

[36] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. Ieee, 2008.

[37] Catarina IM Martins, Leonor Galhardo, Chris Noble, Børge Damsgård, Maria T Spedicato, Walter Zupa, Marilyn Beauchaud, Ewa Kulczykowska, Jean-Charles Massabuau, Toby Carter, et al. Behavioural indicators of welfare in farmed fish. *Fish Physiology and Biochemistry*, 38 (1):17–41, 2012.

[38] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 8834–8844. IEEE, 2022.

[39] 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)*, pages 21274–21284. IEEE, 2023.

[40] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[41] Joel W Newbolt, Jun Zhang, and Leif Ristroph. Flow interactions between uncoordinated flapping swimmers give rise to group cohesion. *Proceedings of the National Academy of Sciences*, 116(7):2419–2424, 2019.

[42] Duy MH Nguyen, Roberto Henschel, Bodo Rosenhahn, Daniel Sonntag, and Paul Swoboda. Lmgp: Lifted multicut meets geometry projections for multi-camera multi-object tracking. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8856–8865. IEEE, 2022.

[43] Alexandru Niculescu-Mizil, Deep Patel, and Iain Melvin. Mctr: Multi camera tracking transformer. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 816–826. IEEE, 2025.

[44] Hiro-Sato Niwa. Newtonian dynamical approach to fish schooling. *Journal of Theoretical Biology*, 181(1):47–63, 1996.

[45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[46] Vassilis M Papadakis, Ioannis E Papadakis, Fani Lamprianidou, Alexios Glaropoulos, and Maroudio Kentouri. A computer-vision system and methodology for the analysis of fish behavior. *Aquacultural engineering*, 46:53–59, 2012.

[47] Malte Pedersen, Joakim Bruslund Haurum, Stefan Hein Bengtson, and Thomas B Moeslund. 3d-zef: A 3d zebrafish tracking benchmark dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2423–2433. IEEE, 2020.

[48] Alfonso Pérez-Escudero and Gonzalo G. de Polavieja. Collective animal behavior from bayesian estimation and probability matching. *PLOS Computational Biology*, 7(11):1–14, 11 2011. doi: 10.1371/journal.pcbi.1002282. URL `https://doi.org/10.1371/journal.pcbi.1002282`.

[49] Alfonso Pérez-Escudero, Julián Vicente-Page, Robert Hinz, Sara Arganda, and Gonzalo Polavieja. Idtracker: Tracking individuals in a group by automatic identification of unmarked animals. *Nature methods*, 11, 06 2014. doi: 10.1038/nmeth.2994.

[50] Kha Gia Quach, Pha Nguyen, Huu Le, Thanh-Dat Truong, Chi Nhan Duong, Minh-Triet Tran, and Khoa Luu. Dyglip: A dynamic graph model with link prediction for accurate multi-camera multiple object tracking. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13779–13788. IEEE, 2021.

[51] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pages 28085–28128, 2025.

[52] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.

[53] Francisco Romero-Ferrero, Mattia Bergomi, Robert Hinz, Francisco Heras, and Gonzalo Polavieja. idtracker.ai: Tracking all individuals in large collectives of unmarked animals. *Nature Methods*, 16, 02 2019. doi: 10.1038/s41592-018-0295-5.

[54] Mattia Segu, Luigi Piccinelli, Siyuan Li, Yung-Hsu Yang, Luc Van Gool, and Bernt Schiele. Samba: Synchronized set-of-sequences modeling for multiple object tracking. In *International Conference on Learning Representations*, volume 2025, pages 30057–30070, 2025.

[55] Minseok Seo, Hyuk-Jae Lee, and Xuan Truong Nguyen. ViT-P3DE*: Vision transformer based multi-camera instance association with pseudo 3D position embedding. In *IJCAI*, pages 1340–1350, 2023.

[56] 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.

[57] Tamás Vicsek and Anna Zafeiris. Collective motion. *Physics reports*, 517(3-4):71–140, 2012.

[58] Minh Vo, Ersin Yumer, Kalyan Sunkavalli, Sunil Hadap, Yaser Sheikh, and Srinivasa G Narasimhan. Self-supervised multi-view person association and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2794–2808, 2020.

[59] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5294–5306. IEEE, 2025.

[60] Yifan Wu, Zhiyang Dou, Yuko Ishiwaka, Shun Ogawa, Yuke Lou, Wenping Wang, Lingjie Liu, and Taku Komura. Cbil: collective behavior imitation learning for fish from real videos. *ACM Transactions on Graphics (TOG)*, 43(6):1–17, 2024.

[61] Xin Zhou, Xiangyong Wen, Zhepei Wang, Yuman Gao, Haojia Li, Qianhao Wang, Tiankai Yang, Haojian Lu, Yanjun Cao, Chao Xu, et al. Swarm of micro flying robots in the wild. *Science robotics*, 7(66):eabm5954, 2022.

Appendix Contents

Appendix	Page
A. TrackFish3D Training Algorithm	16
B. Network Architecture	16
C. Training Details	18
D. Inference Implementation	19
E. Dataset Statistics	21
F. Camera Calibration Parameters	21
G. Per-Scene Results: SynFish Test Set	23
H. Per-Scene Results: 3D-ZeF Test Set	24
I. Temporal Linking Weight Ablation	24
J. Robustness to Camera Calibration Noise	24
K. Robustness to Detection Noise	25
L. Runtime Performance	26
M. Extended Method	27
N. Extended Experiment	30
O. Proof-of-principle downstream analysis on Abnormal Behaviour Detection	32
P. Cross-species generalisation to real-world pigeons	34
Q. 3D-SORT Kinematic Baseline	34
R. Density and Computational Scaling	35
S. Pseudo-Label Noise Audit	35
T. Training-Time Calibration Noise	36

A TrackFish3D Training Algorithm

Algorithm 1: TrackFish3D Training

Input: Scenes $\mathcal{S}$ with calibrated cameras $\{K_c, R_c, t_c\}_{c=1}^{C}$ and 2D detections
Output: Encoder ϕ, association transformer ψ, temporal predictor ρ

for epoch $= 1, \ldots, E$ **do**
  `full` $\leftarrow$ (epoch $> E_{\text{warmup}}$)
  for each scene $\mathcal{S}_k$ **do**
    $\mathcal{M} \leftarrow \emptyset$ `// track memory (sliding window)`
    for each consecutive pair $(t,\ t{+}1)$ **do**
      `// Stage 1: Encode & Associate (Sec. 3.3-3.4)`
      $F_\bullet \leftarrow \phi(\text{dets}_\bullet, \text{cams})$; $H_\bullet \leftarrow \psi_{\text{tf}}(\text{RoPE}(\psi_{\text{proj}}(F_\bullet)))$ for $\bullet \in \{t, t{+}1\}$
      $P_\bullet \leftarrow \sigma\big(\psi_{\text{pw}}([H_\bullet^i \| H_\bullet^j])\big)\ \forall\ c_i \neq c_j$
      `// Stage 2: Pseudo-Supervision & Association Loss (Sec. 3.5-3.6)`
      $(G_\bullet, C_\bullet) \leftarrow \text{PseudoGT}(\text{dets}_\bullet, \text{cams})$
      $\mathcal{L}_{\text{asso}} \leftarrow \sum_\bullet \text{WeightedBCE}(P_\bullet, G_\bullet, C_\bullet)$
      if `full` **then**
        `// Stage 3: Grouping & Triangulation (Sec. 3.7)`
        $A_\bullet \leftarrow 0.6 P_\bullet + 0.4\ \frac{\cos(H_\bullet, H_\bullet)+1}{2}$
        $\mathcal{G}_\bullet \leftarrow \text{GroupTriangulate}(A_\bullet, \text{dets}_\bullet, \text{cams})$;
        prune by confidence
        $\bar{\boldsymbol{h}}_g \leftarrow \ell_2\text{-norm}(\frac{1}{|g|}\sum_{i \in g} \boldsymbol{h}_i)$ for each group g
        `// Stage 4: Contrastive Learning (Sec. 3.8)`
        $\mathcal{L}_{\text{ctr}} \leftarrow \sum_\bullet \text{InfoNCE}(H_\bullet, \{(i,j) \mid i, j \in g\}, \tau)$
        `// Stage 5: Temporal Linking & Prediction (Sec. 3.9)`
        $\mathcal{T} \leftarrow \text{Hungarian}\big(\alpha\, d_{3\text{D}} + \beta\,(1 - \cos)\big)$ gated by thresholds
        $\mathcal{T}_{\text{fb}} \leftarrow \text{FallbackMatch}(\mathcal{G}_{t+1}, \mathcal{M})$
        $\mathcal{L}_{\text{temp}} \leftarrow \text{WeightedMSE}\big(\rho(\bar{\boldsymbol{h}}_{g_t}),\ \bar{\boldsymbol{h}}_{g_{t+1}},\ \mathcal{T} \cup \mathcal{T}_{\text{fb}}\big)$
      else
        $\mathcal{L}_{\text{ctr}}, \mathcal{L}_{\text{temp}} \leftarrow 0$ `// warmup: only` $\mathcal{L}_{\text{asso}}$
      end
      $\mathcal{L} \leftarrow \lambda_{\text{asso}}\mathcal{L}_{\text{asso}} + \lambda_{\text{ctr}}\mathcal{L}_{\text{ctr}} + \lambda_{\text{temp}}\mathcal{L}_{\text{temp}}$
      Update ϕ, ψ, ρ via AdamW (grad clip); append $\mathcal{G}_t$ to $\mathcal{M}$
    end
  end
end

B Network Architecture

TrackFish3D consists of three learned modules: a *Geometric Feature Encoder* ϕ, an *Association Transformer* ψ, and a *Temporal Predictor* ρ. All modules are trained jointly end-to-end. This section provides the full architectural specification of each component, including layer dimensions and parameter counts.

Geometric Feature Encoder ϕ. The encoder maps each 2D detection into a geometric embedding that captures its spatial location, scale, camera assignment, detection confidence, and world-space ray direction. For each detection with bounding box (x_1, y_1, x_2, y_2), segmentation centroid (c_x, c_y), camera ID c, and detection score s, we construct a 9-dimensional raw feature vector f_i^{raw}:

$$f_i^{\text{raw}} = \Big[\ \tfrac{c_x}{W},\ \tfrac{c_y}{H},\ \tfrac{x_2 - x_1}{W},\ \tfrac{y_2 - y_1}{H},\ \tfrac{c}{C-1},\ s,\ r_x,\ r_y,\ r_z\ \Big] \tag{1}$$

where (W, H) is the image resolution, C is the number of cameras, and (r_x, r_y, r_z) is the unit-length world-space ray obtained by unprojecting the centroid through the camera intrinsic inverse K^{-1} and rotating by $R^\top$.

This 9-D vector is processed by a 2-layer MLP with a hidden dimension of 64, followed by LayerNorm, producing a $d{=}128$ dimensional geometric embedding $\boldsymbol{f}_i = \phi(f_i^{\text{raw}})$:

$$\boldsymbol{f}_i = \text{LayerNorm}\big(\text{Linear}_{64 \to 128}\big(\text{ReLU}\big(\text{Linear}_{9 \to 64}(f_i^{\text{raw}})\big)\big)\big) \in \mathbb{R}^{128} \tag{2}$$

Parameter count: $\text{Linear}_{9\to64}$: $9 \times 64 + 64 = 640$; $\text{Linear}_{64\to128}$: $64 \times 128 + 128 = 8{,}320$; LayerNorm_{128}: $2 \times 128 = 256$. **Total encoder MLP:** $9{,}216$ parameters.

Association Transformer ψ. The association transformer takes the set of per-detection embeddings from ϕ and produces both contextualized embeddings $\mathbf{h}_i$ and pairwise cross-camera association probabilities P_{ij}. It consists of three sub-modules:

Input projection. A single linear layer projects the encoder output to the transformer dimension: $\psi_{\text{proj}} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{128}$ where $d_{\text{in}} = 128$ (geometry-only mode).

2D Rotary Position Embedding (RoPE). After input projection and before entering the transformer layers, projected detection tokens receive 2D RoPE [27] based on their normalised bounding-box centres $(c_y, c_x) \in [0, 1]^2$. The 128-dimensional embedding is split into two 64-dimensional halves: the first half encodes the vertical position c_y and the second half encodes the horizontal position c_x. Within each half, we form 32 pairs and apply the standard rotary transform:

$$\theta_k = \frac{c_{\text{axis}}}{10000^{2k/64}} \cdot \pi, \quad k = 0, \ldots, 31 \tag{3}$$

$$(x_1', x_2') = (x_1 \cos\theta_k - x_2 \sin\theta_k,\ x_1 \sin\theta_k + x_2 \cos\theta_k) \tag{4}$$

RoPE has no learnable parameters; it enables the transformer to reason about detection spatial relationships in a translation-equivariant manner.

Transformer encoder. A standard 4-layer Transformer encoder with pre-norm architecture:

- Layers: 4, attention heads: 4, head dimension: 32
- Model dimension $d = 128$, feed-forward dimension $d_{\text{ff}} = 256$
- Dropout: 0.1, activation: ReLU
- Pre-LayerNorm (norm-first) variant

Each layer applies multi-head self-attention followed by a position-wise feed-forward network. The transformer operates on the full set of N detection tokens from all cameras simultaneously, enabling cross-camera reasoning.

Per-layer parameter count: Self-attention QKV: $3\times(128\times128+128) = 49{,}536$; Output projection: $128 \times 128 + 128 = 16{,}512$; FFN: $(128 \times 256 + 256) + (256 \times 128 + 128) = 65{,}920$; LayerNorm $\times2$: $2 \times 256 = 512$. **Per layer:** $132{,}480$**. Four layers:** $529{,}920$**.**

Pairwise MLP. For every pair of detections (i, j) from *different* cameras ($c_i \neq c_j$), we concatenate their transformer outputs and predict an association probability:

$$P_{ij} = \sigma\big(\text{Linear}_{128\to1}\big(\text{ReLU}\big(\text{Linear}_{256\to128}([\mathbf{h}_i \| \mathbf{h}_j])\big)\big)\big) \tag{5}$$

Pairwise MLP parameters: $(256 \times 128 + 128) + (128 \times 1 + 1) = 33{,}025$.

Total association transformer: Input projection (16,512) + transformer layers (529,920) + pairwise MLP (33,025) = ≈**579K** parameters. Note that 2D RoPE contributes **zero learnable parameters**—it stores only a fixed frequency buffer computed from deterministic sinusoidal formulas and applies parameter-free rotations to the embeddings.

Temporal Predictor ρ. The temporal predictor is a small MLP that maps a group embedding $\bar{\boldsymbol{h}}_g$ at frame t to its predicted embedding at frame $t{+}1$:

$$\rho(\bar{\boldsymbol{h}}_g) = \text{Linear}_{256\to128}\big(\text{LayerNorm}_{256}\big(\text{ReLU}\big(\text{Linear}_{128\to256}(\bar{\boldsymbol{h}}_g)\big)\big)\big) \in \mathbb{R}^{128} \tag{6}$$

Parameter count: $\text{Linear}_{128\to256}$: $128 \times 256 + 256 = 33{,}024$; LayerNorm_{256}: $2 \times 256 = 512$; $\text{Linear}_{256\to128}$: $256 \times 128 + 128 = 32{,}896$. **Total predictor:** $66{,}432$ parameters.

At inference, when a track has been unobserved for g frames, ρ is applied iteratively (up to 5 times) to roll the embedding forward, approximating what the track's representation would look like after g frames of motion: $\hat{\bar{\boldsymbol{h}}}^{(t+g)} = \rho^{(g)}(\bar{\boldsymbol{h}}^{(t)})$.

Tab. 6: TrackFish3D parameter budget (geometry-only mode, no visual backbone).

Module	Parameters	%
Geometric Feature Encoder ϕ	9,216	1.4%
Association Transformer ψ	579,457	88.5%
Temporal Predictor ρ	66,432	10.1%
Total (trainable)	**≈655K**	100%

Total model size. The entire TrackFish3D model has approximately **655K trainable parameters**—orders of magnitude smaller than typical vision transformer trackers (>50M parameters). This compact architecture is enabled by the geometry-only design: since all fish are visually indistinguishable, a large visual backbone is unnecessary.

C Training Details

Training protocol. TrackFish3D is trained for **400 epochs** with a **100-epoch warmup** on the SynFish training set (8 scenes, 4–16 fish, ∼350 frames each). During the warmup phase (epochs 1–100), only the association loss $\mathcal{L}_{\text{asso}}$ is active; grouping, contrastive learning, and temporal linking are disabled. This allows the encoder and association transformer to learn basic cross-camera correspondence before the more demanding grouping and tracking losses are introduced. After warmup (epochs 101–400), all three losses ($\mathcal{L}_{\text{asso}}$, $\mathcal{L}_{\text{ctr}}$, $\mathcal{L}_{\text{temp}}$) are jointly optimised.

Training iterates over scenes sequentially within each epoch; within each scene, consecutive frame pairs $(t, t{+}1)$ are processed in temporal order. This sequential processing is essential because the temporal prediction loss requires a sliding-window track memory ($W{=}10$ past frames) that accumulates as the sequence progresses. Each frame pair constitutes one gradient step: the total loss is computed and backpropagated, and all parameters are updated via AdamW before proceeding to the next pair.

Tab. 7: Complete TrackFish3D training hyperparameters.

Category	Hyperparameter	Value
Optimiser	Algorithm	AdamW
	Learning rate	1×10^{-4}
	Weight decay	1×10^{-4}
	Gradient clipping (max norm)	5.0
Loss weights	λ_{asso} (association)	1.0
	λ_{ctr} (contrastive)	0.5
	λ_{temp} (temporal)	0.25
Grouping	Association threshold θ_{assoc}	0.6
	Affinity weights (w_P, w_F)	(0.6, 0.4)
	Minimum group confidence θ_{conf}	0.3
	Two-camera acceptance threshold	0.8
Temporal linking	Distance weight α	0.6
	Appearance weight β	0.4
	Distance threshold θ_{dist}	0.5
	Similarity threshold θ_{sim}	0.6
Fallback matching	Gap distance scale	0.15
	Gap similarity drop	0.04
Post-hoc merge	Merge distance threshold θ_{merge}	2.0
	Merge similarity threshold	0.45
Contrastive	Temperature τ	0.07
	InfoNCE	detection-level
Schedule	Total epochs	400
	Warmup epochs E_{warmup}	100
	Track memory window W	10

Optimiser and hyperparameters.

Pseudo ground-truth caching. Pseudo ground-truth labels (G, C) are computed by pairwise triangulation and bidirectional reprojection-containment checks (see Sec. 3.5), then cached to disk as compressed `.npz` files before the first epoch. This one-time precomputation eliminates redundant geometric computation during training and ensures deterministic supervision across epochs. For each frame, the cached files store the binary assignment matrix $G \in \{-1, 0, 1\}^{N \times N}$ and the continuous confidence matrix $C \in [0, 1]^{N \times N}$.

Feature reuse across frame pairs. To reduce redundant computation, when processing consecutive frame pairs $(t, t{+}1)$ and $(t{+}1, t{+}2)$, the encoder output for frame $t{+}1$ computed in the first pair is *detached* from the computational graph and reused as the frame t input for the next pair. This halves the per-scene encoding cost with no effect on gradient quality (since only the current pair's computation graph is retained for backpropagation).

Checkpoint selection. Checkpoints are saved every 100 epochs and at the final epoch. For the reported SAM3 and YOLO26+SAM2 models, `best_ckpt.pth` is selected by the lowest training objective; the resulting checkpoints are from epochs 396 and 397, respectively. The held-out split is therefore not used for model selection.

D Inference Implementation

At inference, TrackFish3D processes a scene in three sequential passes followed by post-processing. The inference pipeline uses the trained encoder, association transformer, and temporal predictor without oracle information such as the true number of fish. It retains the training affinity and temporal-cost forms, with an inference association threshold of 0.5 (and a stricter 0.8 acceptance threshold for two-camera groups).

Pass 1: Per-frame encoding and grouping. For each frame t in the sequence:

1. **Load and encode:** All 2D detections across C cameras are encoded by the frozen encoder ϕ, producing embeddings $F \in \mathbb{R}^{N \times 128}$.
2. **RoPE and transformer:** Embeddings are projected, augmented with 2D RoPE based on normalised bounding-box centres, and processed by the frozen association transformer ψ, yielding contextualised representations $H \in \mathbb{R}^{N \times 128}$.
3. **Pairwise prediction:** For all ordered cross-camera pairs (i, j) with $c_i \neq c_j$, the pairwise MLP outputs directional association probabilities P_{ij}.
4. **Group and triangulate:** The hybrid affinity matrix $A = 0.6\,P + 0.4\,\frac{\cos(H,H)+1}{2}$ is computed. For each camera pair (c_a, c_b), Hungarian matching uses the directed sub-matrix with rows from camera c_a and columns from camera c_b, without symmetrising P_{ij} and P_{ji}. Connected components form groups. Each group is pruned to one detection per camera (keeping highest detection score), and multi-view triangulation via DLT yields a 3D position $\mathbf{X}_g$. Groups whose reprojected 3D point falls outside any member's bounding box are rejected. Overlapping groups (sharing detections) are resolved by keeping the higher-confidence group.
5. **Group embedding:** Each group's embedding is the ℓ_2-normalised mean of its members' transformer outputs: $\mathbf{z}_g = \frac{\overline{H}_{\text{members}}}{\|\overline{H}_{\text{members}}\|_2}$.

Pass 1 runs entirely under `torch.no_grad()` with optional FP16 mixed precision on GPU.

Pass 2: Temporal tracking. Active tracks are maintained in a list, each storing: label, last 3D position, last embedding (both NumPy and Torch), velocity estimate, gap count, and birth/last-seen timestamps. For each frame:

Track-to-group matching. Active tracks are linked to current-frame groups via Hungarian matching with the same weighted distance-and-similarity cost form used during training:

$$\text{cost}(i, j) = \alpha \cdot \frac{d_{\text{3D}}(\hat{\mathbf{X}}_i, \mathbf{X}_j)}{\theta^{\text{local}}_{\text{dist}}} + \beta \cdot (1 - \cos(\hat{\mathbf{z}}_i, \mathbf{z}_j)) \tag{7}$$

where $\hat{\mathbf{X}}_i$ and $\hat{\mathbf{z}}_i$ are the *predicted* position and embedding for track i.

PredictorM embedding rollforward. When a track has been unobserved for $g > 0$ frames and the trained temporal predictor ρ is available, the track's embedding is rolled forward by iteratively applying ρ (capped at 5 iterations to limit drift): $\hat{\mathbf{z}} = \ell_2\text{-norm}(\rho^{(\min(g,5))}(\mathbf{z}_{\text{last}}))$. This is the direct inference-time payoff of the temporal prediction loss $\mathcal{L}_{\text{temp}}$.

Constant-velocity position prediction. Position is predicted using exponentially smoothed velocity: $\hat{\mathbf{X}} = \mathbf{X}_{\text{last}} + \mathbf{v} \cdot (g + 1)$, where $\mathbf{v}$ is updated at each observation as $\mathbf{v} \leftarrow 0.7\,\mathbf{v}_{\text{new}} + 0.3\,\mathbf{v}_{\text{old}}$.

Gate relaxation. For tracks with gap $g > 0$, the distance and similarity gates are relaxed using the same formulas as training's fallback matching:

$$\theta_{\text{dist}}^{\text{local}} = \theta_{\text{dist}} \cdot (1 + 0.15 \cdot g), \quad \theta_{\text{sim}}^{\text{local}} = \max(0.1,\ \theta_{\text{sim}} - 0.04 \cdot g) \tag{8}$$

Track birth and death. Groups that remain unmatched after Hungarian assignment spawn new tracks. Tracks that have not been observed for more than `max_gap` $= 15$ consecutive frames are killed and moved to a graveyard for potential post-hoc merging.

Pass 2.5: Post-hoc track merge. After temporal tracking, short-lived track fragments may correspond to the same fish observed before and after a long occlusion. Post-hoc merging identifies such fragments and consolidates them:

1. All dead tracks (including those still active at the end of the sequence) are sorted chronologically by birth time.
2. For each fragment B, we search all earlier-ending fragments A where A's last frame precedes B's first frame (non-overlapping).
3. A merge is considered if: (a) the 3D distance between A's last position and B's first position is ≤ 2.0, and (b) the cosine similarity between A's last embedding and B's first embedding exceeds 0.45 (either raw or after rolling A's embedding forward with ρ).
4. The best merge candidate maximises: $\text{score} = \text{sim} - 0.1 \cdot \frac{\text{gap}}{100} - 0.05 \cdot \frac{d_{3\text{D}}}{\theta_{\text{merge}}}$, where $\theta_{\text{merge}} = 2.0$ is the merge distance threshold.
5. Accepted merges relabel all of B's observations to A's label.

Post-processing. Two post-processing steps are applied to all methods (TrackFish3D and baselines) identically for fairness:

Gap filling. For each track, if two consecutive observations are separated by $1 < \Delta t \leq$ `max_gap` frames, intermediate frames are filled by linear interpolation of the 3D position.

Ghost filtering. Tracks with fewer than `min_obs` $= 2$ total observations are removed as likely detection noise or spurious groupings.

After post-processing, tracks are relabelled to consecutive identifiers (A, B, C, ...) sorted by first appearance time, and the output is written as a CSV file with columns: `object, Timestamp, X, Y, Z, group`.

E Dataset Statistics

Tab. 8: Dataset summary and difficulty statistics. Occlusion level is derived from the OccScore, which is computed per scene by taking, for each (frame, camera) pair, the maximum pairwise 2D bounding-box IoU among all tracked objects, then averaging these maxima across all frames and camera views. %Overlap is the fraction of (frame, camera) instances with any overlap (IoU>0.01).

Split	Scene	#Fish	#Frames	#Cams	Resolution	OcclLevel	%Overlap
SynFish (train)	train1	4	359	3	1920×1080	Low	0.2%
SynFish (train)	train2	6	349	3	1920×1080	Mild	10.2%
SynFish (train)	train3	8	362	3	1920×1080	Moderate	81.1%
SynFish (train)	train4	10	359	3	1920×1080	Moderate	75.1%
SynFish (train)	train5	14	359	3	1920×1080	Heavy	91.9%
SynFish (train)	train6	14	362	3	1920×1080	Mild	66.9%
SynFish (train)	train7	16	362	3	1920×1080	Mild	74.1%
SynFish (train)	train8	16	362	3	1920×1080	Moderate	99.7%
SynFish (test)	test1	4	359	3	1920×1080	Low	0.2%
SynFish (test)	test2	5	359	3	1920×1080	Low	0.3%
SynFish (test)	test3	8	359	3	1920×1080	Moderate	68.0%
SynFish (test)	test4	16	362	3	1920×1080	Heavy	100.0%
SynFish (test)	test5	16	362	3	1920×1080	Heavy	99.9%
SynFish (test)	test6	16	362	3	1920×1080	Mild	99.1%
3D-ZeF (train)	zebra01	2	800	2	2704×1520	Moderate	69.4%
3D-ZeF (train)	zebra03	2	800	2	2704×1520	Low	16.8%
3D-ZeF (test)	zebra02	5	701	2	2704×1520	Moderate	83.8%
3D-ZeF (test)	zebra04	5	910	2	2704×1520	Heavy	86.8%

F Camera Calibration Parameters

This section lists the complete intrinsic and extrinsic camera calibration for every dataset scene. Extrinsics are reported as the rotation matrix $R \in \mathrm{SO}(3)$ and translation vector $\mathbf{t} \in \mathbb{R}^3$ such that a world point $\mathbf{X}$ projects to pixel $\mathbf{u} = K[R|\mathbf{t}]\mathbf{X}$.

SynFish Synthetic Scenes. All 14 SynFish scenes are rendered at 1920×1080 with $C{=}3$ synchronised pinhole cameras sharing identical intrinsics:

$$K_{\mathrm{syn}} = \begin{bmatrix} 1866.67 & 0 & 960.0 \\ 0 & 1866.67 & 540.0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\mathrm{HFoV} \approx 54.3^\circ) \tag{9}$$

Camera extrinsics are defined by **7 distinct rig configurations** (numbered 0–6). Scenes sharing the same rig use identical camera placements. The training set spans 6 rig configurations; the test set uses 4, of which one (rig 0) is unseen during training. Table 9 maps each scene to its rig, and Table 10 lists the full extrinsic parameters.

Tab. 9: SynFish scene-to-rig mapping. Scenes sharing a rig use identical camera placements (same R and $\mathbf{t}$ for all 3 cameras). Rig 0 appears only in the test set.

Rig	Train scenes	Test scenes
0	—	test1, test2
1	train1	—
2	train2	—
3	train4	—
4	train5, train6	—
5	train3	test3
6	train7, train8	test4, test5, test6

Tab. 10: SynFish camera rig extrinsics. Each rig defines 3 cameras (0, 1, 2) with rotation R (shown row-by-row) and translation $\mathbf{t}$. All cameras share the intrinsic K_{syn} above.

Rig	Cam	R (row-major)			$\mathbf{t}$		
0	0	−0.6129	0.7902	−0.0000	7.48	18.22	43.67
		0.1589	0.1232	−0.9796			
		−0.7740	−0.6004	−0.2011			
	1	−0.9866	−0.1633	0.0000	−1.35	13.66	73.22
		−0.1450	0.8759	−0.4602			
		0.0752	−0.4540	−0.8878			
	2	−0.9282	0.3721	−0.0000	5.41	24.25	34.57
		−0.0298	−0.0744	−0.9968			
		−0.3709	−0.9252	0.0802			
1	0	−0.6129	0.7902	−0.0000	8.20	19.80	43.94
		0.1589	0.1232	−0.9796			
		−0.7740	−0.6004	−0.2011			
	1	−0.9866	−0.1633	0.0000	−0.54	13.67	66.57
		−0.1450	0.8759	−0.4602			
		0.0752	−0.4540	−0.8878			
	2	−0.9282	0.3721	−0.0000	11.51	23.96	42.23
		−0.0298	−0.0744	−0.9968			
		−0.3709	−0.9252	0.0802			
2	0	−0.7547	0.6561	0.0000	13.00	18.33	29.30
		0.0936	0.1076	−0.9898			
		−0.6494	−0.7470	−0.1426			
	1	−0.9511	0.3090	−0.0000	4.26	26.58	59.48
		0.2654	0.8169	−0.5120			
		−0.1582	−0.4870	−0.8590			
	2	0.9511	−0.3090	0.0000	−6.22	24.97	51.34
		0.1026	0.3159	−0.9432			
		0.2915	0.8971	0.3322			
3	0	−0.6129	0.7902	−0.0000	7.48	18.22	43.67
		0.1589	0.1232	−0.9796			
		−0.7740	−0.6004	−0.2011			
	1	−0.9866	−0.1633	0.0000	−1.35	13.66	73.22
		−0.1450	0.8759	−0.4602			
		0.0752	−0.4540	−0.8878			
	2	−0.9311	0.3649	−0.0000	5.70	24.25	34.52
		−0.0293	−0.0747	−0.9968			
		−0.3637	−0.9281	0.0802			
4	0	0.9499	0.3126	−0.0000	−2.72	10.39	72.20
		0.1146	−0.3482	−0.9304			
		−0.2908	0.8838	−0.3666			
	1	−0.9866	−0.1633	0.0000	−0.36	15.77	70.33
		−0.1450	0.8759	−0.4602			
		0.0752	−0.4540	−0.8878			
	2	−0.9282	0.3721	−0.0000	5.98	19.98	45.08
		−0.0298	−0.0744	−0.9968			
		−0.3709	−0.9252	0.0802			
5	0	−0.6129	0.7902	−0.0000	4.87	17.82	45.65
		0.1589	0.1232	−0.9796			
		−0.7740	−0.6004	−0.2011			
	1	−0.9866	−0.1633	0.0000	−1.42	13.00	71.09
		−0.1450	0.8759	−0.4602			
		0.0752	−0.4540	−0.8878			
	2	−0.5210	−0.8536	0.0000	−1.94	22.44	31.22
		0.0685	−0.0418	−0.9968			
		0.8508	−0.5193	0.0802			
6	0	0.9499	0.3126	−0.0000	−0.59	13.00	68.85
		0.1146	−0.3482	−0.9304			
		−0.2908	0.8838	−0.3666			
	1	−0.9866	−0.1633	0.0000	−0.54	16.15	70.02
		−0.1450	0.8759	−0.4602			
		0.0752	−0.4540	−0.8878			
	2	−0.9311	0.3649	−0.0000	6.34	17.98	45.20
		−0.0293	−0.0747	−0.9968			
		−0.3637	−0.9281	0.0802			

3D-ZeF Zebrafish Sequences. The 3D-ZeF benchmark [47] uses $C{=}2$ cameras at 2704×1520 resolution. Both cameras retain *fixed intrinsics* across all four sequences (the physical cameras are not changed between recordings):

$$K_0 = \begin{bmatrix} 1490.78 & 0 & 1343.87 \\ 0 & 1463.64 & 781.86 \\ 0 & 0 & 1 \end{bmatrix}, \quad K_1 = \begin{bmatrix} 1460.43 & 0 & 1348.20 \\ 0 & 1457.88 & 774.27 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Note that unlike the SynFish cameras, K_0 and K_1 have different focal lengths and principal points because they are two distinct physical cameras. The extrinsics vary per sequence because the cameras are repositioned slightly between recording sessions (Table 11).

Tab. 11: 3D-ZeF camera extrinsics per sequence. All sequences share the intrinsics K_0, K_1 above.

Sequence	**Cam**	R (row-major)			**t**		
zebra01 (train)	0	0.9999	−0.0143	0.0054	−14.68	−16.34	31.24
		0.0139	0.9974	0.0708			
		−0.0064	−0.0707	0.9975			
	1	0.9999	−0.0077	−0.0067	−14.63	−9.80	48.50
		0.0079	0.1591	0.9872			
		−0.0065	−0.9872	0.1591			
zebra03 (train)	0	0.9989	−0.0109	0.0453	−15.03	−15.83	33.80
		0.0076	0.9973	0.0726			
		−0.0460	−0.0722	0.9963			
	1	0.9987	−0.0487	−0.0128	−13.18	−10.62	48.76
		0.0216	0.1857	0.9824			
		−0.0455	−0.9814	0.1866			
zebra02 (test)	0	0.9990	−0.0115	0.0430	−14.97	−15.95	32.52
		0.0092	0.9986	0.0528			
		−0.0435	−0.0524	0.9977			
	1	0.9994	−0.0332	−0.0100	−13.73	−10.09	48.67
		0.0154	0.1669	0.9859			
		−0.0311	−0.9854	0.1673			
zebra04 (test)	0	0.9999	−0.0159	0.0011	−15.09	−14.85	33.05
		0.0158	0.9981	0.0594			
		−0.0020	−0.0594	0.9982			
	1	0.9988	0.0458	−0.0148	−15.74	−11.04	46.73
		0.0048	0.2121	0.9772			
		0.0479	−0.9762	0.2117			

G Per-Scene Results: SynFish Test Set

Tab. 12: Per-scene accuracy metrics on the SynFish test set (6 held-out scenes). Scene names correspond to Table 8: test1 (4 fish), test2 (5 fish), test3 (8 fish), test4–test6 (16 fish each).

	MOTA (%) ↑							**MT (%) ↑**							**ML (%) ↓**						
Method	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**	**Avg**	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**	**Avg**	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**	**Avg**
TrackFish3D (SAM3)	99.6	99.9	96.2	86.2	94.8	98.0	95.8	100	100	100	81.2	93.8	100	95.8	0	0	0	0	0	0	0.0
TrackFish3D (YOLO26+SAM2)	100	100	97.7	86.3	96.7	99.1	96.6	100	100	100	81.2	93.8	100	95.8	0	0	0	0	0	0	0.0
SAM3+Hung.	100	100	93.8	55.2	75.1	77.1	83.5	100	100	100	31.2	43.8	56.2	71.9	0	0	0	0	0	0	0.0
SAM3+Greedy	100	100	93.8	55.2	75.1	77.1	83.5	100	100	100	31.2	43.8	56.2	71.9	0	0	0	0	0	0	0.0
YOLO26+SAM2+Hung.	99.7	99.8	95.9	53.7	69.0	75.0	82.2	100	100	100	25.0	43.8	50.0	69.8	0	0	0	0	0	0	0.0
YOLO26+SAM2+Greedy	99.7	99.8	95.9	53.7	69.0	75.0	82.2	100	100	100	25.0	43.8	50.0	69.8	0	0	0	0	0	0	0.0
Self-MVA	−65.0	−75.9	−44.5	−72.4	−66.3	−63.4	−64.6	0	0	0	0	0	0	0.0	100	100	87.5	100	100	93.8	96.9
ASNet	82.1	60.8	58.5	−0.8	6.1	−6.9	33.3	100	60	75	12.5	31.2	18.8	49.6	0	0	12.5	12.5	18.8	25	11.5
SambaMOTR	−0.3	−26.8	16.8	−14.3	−19.8	−18.8	−10.5	25	20	12.5	6.2	6.2	0	11.7	25	80	50	93.8	93.8	100	73.8
MOTIP	44.0	48.0	−1.4	3.9	−0.2	−6.1	14.7	50	60	0	6.2	6.2	6.2	21.4	25	20	25	68.8	75	43.8	42.9
ReST	81.0	73.1	80.4	54.2	32.4	46.8	61.3	100	80	100	62.5	43.8	62.5	74.8	0	0	0	0	0	0	0.0
MCTR	99.4	46.9	86.9	83.3	88.0	97.5	83.7	100	40	87.5	81.2	87.5	100	82.7	0	0	0	0	0	0	0.0

Tab. 13: Per-scene identity stability metrics on the SynFish test set (6 held-out scenes). Same scene abbreviations as Table 12.

	IDSw ↓							**Frag ↓**							**MTBF ↑**						
Method	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**	**Avg**	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**	**Avg**	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**	**Avg**
TrackFish3D (SAM3)	0	0	0	0	0	0	0.0	0	0	2	18	9	10	6.5	359.0	358.6	276.4	158.6	224.7	220.5	266.3
TrackFish3D (YOLO26+SAM2)	0	0	0	0	0	1	0.2	0	0	0	11	5	6	3.7	359.0	359.0	350.6	198.5	271.2	240.5	296.5
SAM3+Hung.	0	0	5	12	17	11	7.5	0	0	6	27	25	19	12.8	359.0	359.0	142.4	66.0	77.8	98.3	183.8
SAM3+Greedy	0	0	5	12	17	11	7.5	0	0	6	27	25	19	12.8	359.0	359.0	142.4	66.0	77.8	98.3	183.8
YOLO26+SAM2+Hung.	0	0	3	21	13	15	8.7	0	0	7	27	18	21	12.2	358.0	358.4	154.1	54.1	88.2	85.9	183.1
YOLO26+SAM2+Greedy	0	0	3	21	13	15	8.7	0	0	7	27	18	21	12.2	358.0	358.4	154.1	54.1	88.2	85.9	183.1
Self-MVA	0	0	6	2	1	1	1.7	0	0	6	1	3	3	2.2	34.0	9.0	16.6	10.6	10.1	14.9	15.9
ASNet	2	5	8	34	37	39	20.8	5	5	17	44	47	56	29.0	116.1	92.9	69.0	32.3	30.4	23.5	60.7
SambaMOTR	0	0	0	0	0	0	0.0	26	3	9	2	7	0	7.8	24.7	74.5	73.8	103.0	34.8	0.0	51.8
MOTIP	0	0	0	0	0	0	0.0	30	19	29	18	18	40	25.7	30.4	55.8	34.9	45.9	44.9	35.3	41.2
ReST	12	51	77	228	334	340	173.7	5	23	26	114	174	129	78.5	49.9	15.8	16.6	9.8	6.2	6.8	17.5
MCTR	0	7	3	0	0	0	1.7	1	14	17	32	10	11	14.2	286.4	48.5	96.9	110.6	209.1	211.8	160.6

H Per-Scene Results: 3D-ZeF Test Set

Tab. 14: Per-scene 3D tracking results on the 3D-ZeF test set (zebra2: 5 fish, zebra4: 5 fish). ReST fails to produce any valid cross-camera associations on zebrafish sequences.

	MOTA% ↑			MT% ↑			ML% ↓			IDSw ↓			Frag ↓			MTBF ↑		
Method	**z2**	**z4**	**Avg**	**z2**	**z4**	**Avg**	**z2**	**z4**	**Avg**	**z2**	**z4**	**Avg**	**z2**	**z4**	**Avg**	**z2**	**z4**	**Avg**
TrackFish3D	82.7	79.4	81.1	100	80	90.0	0	0	0.0	2	6	4.0	24	37	30.5	103.9	83.3	93.6
Hung.+NN	77.1	77.7	77.4	100	80	90.0	0	0	0.0	6	6	6.0	60	86	73.0	43.8	41.2	42.5
Greedy+NN	76.3	77.4	76.9	100	80	90.0	0	0	0.0	8	12	10.0	64	88	76.0	40.3	37.1	38.7
Self-MVA	84.7	16.8	50.8	100	20	60.0	0	0	0.0	2	41	21.5	33	114	73.5	81.2	15.8	48.5
ASNet	13.9	−13.4	0.3	0	0	0.0	0	0	0.0	55	90	72.5	152	198	175.0	11.5	9.7	10.6
SambaMOTR	−31.3	−21.4	−26.4	0	0	0.0	100	100	100.0	0	1	0.5	0	2	1.0	7.0	6.0	6.5
MOTIP	−34.1	−57.6	−45.9	0	0	0.0	100	100	100.0	1	2	1.5	4	9	6.5	3.3	2.1	2.7
ReST	0.0	0.0	0.0	0	0	0.0	100	100	100.0	0	0	0.0	0	0	0.0	0.0	0.0	0.0
MCTR	−1.2	1.3	0.1	0	0	0.0	0	80	40.0	27	15	21.0	68	47	57.5	8.7	8.8	8.8

I Temporal Linking Weight Ablation

During training, the temporal linking weights (α, β) govern which cross-frame group pairs are selected as positive examples for the temporal prediction loss $\mathcal{L}_{\text{temp}}$ (Sec. 3.5). We set α=0.6, β=0.4 so that matched pairs must satisfy both spatial proximity *and* embedding agreement, yielding a high-quality supervision signal for the predictor $\mathcal{M}$: pairs that are merely close in 3D but visually dissimilar are down-weighted in the cost matrix, reducing noisy gradients.

At inference, the same (α, β) parameterises the temporal linking cost (Sec. 3.6). Table 15 ablates this weighting on the SynFish test set (6 held-out scenes) with all other components fixed. Performance is stable across a wide operating range ($\alpha \in [0.4, 1.0]$), confirming that the upstream cross-camera association produces well-separated 3D trajectories where spatial proximity is already a reliable temporal cue. Degradation appears only at the embedding similarity-only extreme (α=0, β=1), where removing the 3D distance signal increases identity switches and fragmentation. We therefore adopt α=0.6, β=0.4 as it (i) matches the training regime, ensuring consistency between the distribution of temporal matches seen during training and those encountered at inference, and (ii) retains the learned embedding similarity as a complementary cue for disambiguating spatially proximate individuals in denser scenarios.

Tab. 15: Ablation of temporal linking weights (α, β) on the SynFish test set (6 scenes, averaged). The cost is robust for $\alpha \geq 0.4$; removing the distance component (α=0) degrades identity stability. The selected setting is **bolded**.

α	β	**MOTA ↑**	**MT% ↑**	**ML% ↓**	**IDSw ↓**	**Frag ↓**	**MTBF ↑**
1.0	0.0	95.8%	95.8%	0.0%	0.0	6.5	266.3
0.8	0.2	95.8%	95.8%	0.0%	0.0	6.5	266.3
0.6	**0.4**	**95.8%**	**95.8%**	**0.0%**	**0.0**	**6.5**	**266.3**
0.4	0.6	95.8%	95.8%	0.0%	0.0	6.5	266.3
0.2	0.8	95.8%	95.8%	0.0%	0.2	6.8	262.5
0.0	1.0	95.8%	95.8%	0.0%	0.3	7.5	257.3

J Robustness to Camera Calibration Noise

TrackFish3D relies on calibrated camera extrinsics for geometric pseudo-labelling, triangulation, and temporal linking. To characterise how sensitive the system is to calibration error, we inject increasing levels of Gaussian noise into the camera extrinsics at *inference time* (the model is trained with the original, unperturbed calibration) and re-evaluate on the SynFish test set.

Perturbation protocol. For each noise level σ, we independently perturb (i) rotation by sampling $\delta\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \sigma_R^2 I_3)$ and composing $R' = \exp([\delta\boldsymbol{\omega}]_\times)\, R$ via the Rodrigues formula, and (ii) translation by adding $\delta\mathbf{t} \sim \mathcal{N}(\mathbf{0}, \sigma_t^2 I_3)$. We set $\sigma_R = \sigma_t = \sigma$ and report results for four noise levels alongside the unperturbed baseline (Table 16). All other hyperparameters remain unchanged.

Tab. 16: Robustness of TrackFish3D (SAM 3) to camera extrinsic perturbation on the SynFish test set (6 scenes, averaged). σ_R is the rotation noise standard deviation (degrees) and σ_t is the translation noise standard deviation (world units). The unperturbed baseline (σ=0) is shown in the first row.

σ_R (°)	σ_t	MOTA↑	MT%↑	ML%↓	IDSw↓	Frag↓	MTBF↑
0	0	95.8%	95.8%	0.0%	0.0	6.5	266.3
0.05	0.05	94.1%	95.8%	0.0%	0.0	5.0	274.0
0.1	0.1	92.1%	93.8%	0.0%	0.0	9.2	246.6
0.2	0.2	81.3%	82.3%	1.0%	0.2	20.3	206.0

Discussion. The system is robust to small calibration errors: at σ=0.05 the MOTA drops by only 1.7 percentage points (pp) and all tracks remain mostly-tracked. At σ=0.1 the degradation is still moderate ($-$3.7 pp MOTA) with no increase in identity switches, though fragment count rises from 6.5 to 9.2 as noisy triangulation occasionally pushes 3D positions outside the matching gate. Performance degrades more steeply at σ=0.2 ($-$14.5 pp MOTA, MT% drops to 82.3%) because the corrupted projection matrices $P{=}K[R'|t']$ cause the reprojection-based group validation to reject an increasing number of valid cross-camera associations, which in turn produces fewer and less accurate 3D detections.

These results are consistent with the geometry of the problem. TrackFish3D solves associations in 2D and lifts them to 3D via Direct Linear Transform (DLT) triangulation; small extrinsic errors propagate quadratically through the $P{=}K[R|t]$ projection and are amplified at greater object distances. The practical implication is that TrackFish3D requires calibration accuracy on the order of $\sigma_R \lesssim 0.1°$ and $\sigma_t \lesssim 0.1$ world units to maintain near-baseline performance.

K Robustness to Detection Noise

In addition to camera calibration errors (Appendix J), real-world deployments also face imperfect 2D detectors. We note that *all* main results already operate on noisy detections: the 2D inputs are produced by off-the-shelf models (SAM 3 or YOLO26+SAM2) finetuned on training data, not from ground-truth bounding boxes. These detectors inherently introduce false positives (spurious boxes), false negatives (missed fish), and centroid localisation error. The ablation below therefore injects *additional* synthetic noise on top of this realistic detection noise to stress-test TrackFish3D's tolerance beyond normal operating conditions.

Perturbation protocol. A single noise parameter σ controls both failure modes simultaneously:

- **False negatives (FN):** each real detection is independently dropped with probability σ.
- **False positives (FP):** for each frame-camera with n real detections, $\lfloor \sigma \cdot n \rfloor$ (plus a probabilistic extra for the fractional part) spurious detections are injected at random image locations with bounding-box sizes sampled from the empirical distribution of real detections and scores drawn from a Beta distribution ($\text{mean} \approx 0.85$) to mimic marginal false positives.

We evaluate at four noise levels ($\sigma \in \{0.05, 0.10, 0.20, 0.40\}$) alongside the unperturbed baseline on the SynFish test set (6 scenes). All experiments use a fixed random seed for reproducibility.

Tab. 17: Robustness of TrackFish3D (SAM 3) to detection noise on the SynFish test set (6 scenes, averaged). σ is the noise level controlling both the false-negative drop rate and the false-positive injection rate. The unperturbed baseline (σ=0) is shown in the first row.

σ	MOTA↑	MT%↑	ML%↓	IDSw↓	Frag↓	MTBF↑
0	95.8%	95.8%	0.0%	0.0	6.5	266.3
0.05	90.3%	95.8%	0.0%	3.8	12.2	175.2
0.10	84.1%	94.8%	0.0%	7.0	15.3	126.0
0.20	72.3%	86.5%	0.0%	11.2	24.8	86.4
0.40	45.4%	40.6%	3.1%	15.3	63.5	37.1

Discussion. TrackFish3D is robust to moderate detection noise. At σ=0.05 (5% FN drop rate, 5% FP injection rate), the MOTA drops by only 5.5 percentage points (pp) and all tracks remain mostly-tracked (MT% = 95.8%), indicating that the model's cross-camera association and temporal

linking readily absorb occasional missed or spurious detections. At σ=0.10 the degradation remains moderate (-11.7 pp MOTA) with MT% still at 94.8% and zero mostly-lost tracks; the identity switch count rises to 7.0 as the increased FP rate occasionally creates phantom groups that briefly capture a track.

At σ=0.20, performance drops more substantially (-23.5 pp MOTA, MT% falls to 86.5%) but no tracks are mostly-lost, demonstrating that the temporal predictor ρ and post-hoc merge successfully bridge the gaps caused by the 20% false-negative rate. The fragment count roughly quadruples compared to baseline (24.8 vs. 6.5), reflecting the increasing frequency of detection gaps that interrupt continuous tracking.

At σ=0.40 the system degrades severely (MOTA drops to 45.4%, MT% to 40.6%), with the degradation following the same density-dependent pattern observed for camera noise (Appendix J). The 8-fish scene retains 51.4% MOTA despite the extreme noise, whereas the three 16-fish scenes fall to 29–36% MOTA, because the high false-positive rate roughly doubles the effective number of detections per frame, amplifying the combinatorial complexity of cross-camera association in already-dense scenes. Meanwhile, the 4- and 5-fish scenes maintain 54% and 71% MOTA respectively, as the low object count limits the number of plausible spurious groupings.

L Runtime Performance

Table 18 reports the training time, inference speed, and peak GPU memory of TrackFish3D. All measurements are taken on a single NVIDIA A100 80GB PCIe GPU with batch size 1 (one frame-pair at a time).

Tab. 18: TrackFish3D runtime and memory profile. Training time is for the full 400-epoch schedule on SynFish (8 training scenes). Inference speed is measured per frame (all cameras), averaged over the SynFish test set.

Metric	**Value**
Parameters	655 K
Training time	17h 36m
Inference (ms / frame)	124.8
Peak GPU memory (GB)	0.02

M Extended Method

This appendix contains only the implementation details explicitly referenced from the main Method section.

M.1 Grouping, Thresholding, and Geometric Validation

Predicted correspondence probabilities are converted into multi-view fish hypotheses through a four-stage grouping pipeline.

Affinity matrix. For every detection pair (i, j) from different cameras, we construct a combined affinity score:

$$A_{ij} = w_p \, P_{ij} + w_f \, \frac{\cos(\boldsymbol{h}_i, \boldsymbol{h}_j) + 1}{2},$$

where P_{ij} is the learned correspondence probability, $\cos(\cdot, \cdot)$ denotes cosine similarity between the transformer embeddings, and the weights w_p=0.6, w_f=0.4 balance association confidence and embedding agreement. The cosine term is shifted from $[-1, 1]$ to $[0, 1]$ so that both terms share the same range.

Per-camera-pair Hungarian matching. For each unordered camera pair (c_a, c_b), we extract the directed sub-matrix A_{c_a, c_b} with rows indexed by detections from camera c_a and columns indexed by detections from camera c_b, and solve a maximum-weight bipartite assignment using the Hungarian algorithm [34] (equivalently, minimising $-A_{c_a, c_b}$). Because the pairwise head operates on ordered concatenations $[\boldsymbol{h}_i \| \boldsymbol{h}_j]$, the predicted matrix is generally asymmetric; the implementation therefore uses a fixed row/column camera ordering rather than symmetrising P_{ij} and P_{ji}. Matched pairs whose affinity falls below a threshold τ_{assoc} are discarded; two-camera groups require a stricter threshold to reduce false triangulations.

Transitive merging. Accepted matches are treated as edges in a graph over all N detections. Connected components are computed via union-find, yielding multi-view groups that may span all C cameras. Within each component, if multiple detections originate from the same camera, only the one with the highest detection confidence is retained. Groups with fewer than two unique cameras are discarded.

Triangulation and geometric validation. Each group is triangulated [24] into a candidate 3D point by solving the linear system arising from the member centroids and camera projection matrices. The candidate is validated by reprojecting the 3D point into every member camera and checking that the reprojection falls inside the corresponding bounding box. Groups that fail this containment check are rejected, ensuring that accepted hypotheses are geometrically consistent. A group embedding $\bar{\boldsymbol{h}}_g = \ell_2\text{-norm}\big(\frac{1}{|g|} \sum_{i \in g} \boldsymbol{h}_i\big)$ is computed as the ℓ_2-normalised mean of its member transformer embeddings. The group confidence used downstream is the implementation heuristic

$$\text{conf}(g) = \min(1.0,\, 0.2 + 0.2\, n_{\text{cams}}) \times \begin{cases} 0.75, & n_{\text{cams}} = 2 \\ 1, & n_{\text{cams}} \geq 3, \end{cases}$$

where n_{cams} is the number of unique supporting cameras. Thus, 2-view groups receive confidence 0.45 and 3-view groups receive confidence 0.8, making the threshold θ_{conf}=0.3 in Table 7 meaningful rather than vacuous.

M.2 Positive Mining for the Contrastive Loss

Within each frame, detections belonging to the same multi-view group are treated as positives, and all other detections as negatives.

Positive mining from grouping. For each group with confidence above a minimum threshold, every member detection i is paired with all other members of the same group. Because groups are formed from cross-camera detections of the same fish, this yields positive pairs without any identity labels: the self-supervised grouping step provides the supervision signal.

Frame-level contrastive implementation. Given the transformer output embeddings $\{\boldsymbol{h}_i\}_{i=1}^{N}$, we first ℓ_2-normalise to obtain unit vectors $\boldsymbol{z}_i = \boldsymbol{h}_i/\|\boldsymbol{h}_i\|$. We then compute a scaled similarity matrix $S_{ij} = \boldsymbol{z}_i^\top \boldsymbol{z}_j/\tau$, where τ is a temperature parameter. For each anchor detection i with positive set $\mathcal{P}_i$ (same-group members), the loss is:

$$\mathcal{L}_{\text{ctr}}^{(i)} = -\log \frac{\displaystyle\sum_{j\in\mathcal{P}_i} \exp(S_{ij})}{\displaystyle\sum_{\substack{k=1\\k\neq i}}^{N} \exp(S_{ik})},$$

and the frame-level contrastive loss averages over all anchors: $\mathcal{L}_{\text{ctr}} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} \mathcal{L}_{\text{ctr}}^{(i)}$, where $\mathcal{A}$ is the set of detections that have at least one positive. The denominator sums over all other detections regardless of group membership, pushing different-identity embeddings apart.

M.3 Fallback Matching and Temporal Predictor Details

To encourage identity stability over time, TrackFish3D links triangulated group hypotheses between consecutive frames and trains a temporal predictor.

Hungarian temporal matching. Given groups from frames t and $t{+}1$ (each with a 3D position $\mathbf{X}$ and a group embedding $\bar{\boldsymbol{h}}$), we construct a cost matrix:

$$\text{cost}(i,j) = \alpha\,\frac{\|\mathbf{X}_j^{(t+1)} - \mathbf{X}_i^{(t)}\|}{d_{\max}} + \beta\,\big(1 - \cos(\bar{\boldsymbol{h}}_i^{(t)}, \bar{\boldsymbol{h}}_j^{(t+1)})\big),$$

where α=0.6 and β=0.4 balance 3D proximity and embedding similarity. The assignment is solved by the Hungarian algorithm [34], and matched pairs are gated: a match is accepted only if the 3D distance is below a threshold $d_{\max}$ and the cosine similarity exceeds a threshold $s_{\min}$. Each accepted match receives a weight $w_{ij} = \min(\text{conf}_i, \text{conf}_j)$ reflecting the geometric reliability of both groups.

Fallback multi-frame matching. Groups in frame $t{+}1$ that remain unmatched may correspond to fish that were temporarily missed in frame t. A sliding window of past group embeddings (up to 10 frames) is maintained. For each unmatched group in $t{+}1$, we search older frames for a compatible partner, relaxing the distance gate by a factor proportional to the frame gap g and reducing the similarity gate accordingly. This fallback mechanism provides training pairs even under transient occlusion, ensuring the predictor receives supervision for longer-horizon identity linking.

Predictor architecture and loss implementation. We train a temporal predictor ρ, a two-layer MLP, on matched groups from adjacent frames to preserve identity consistency through short occlusions. For each matched pair (i,j), the predictor takes the observed group embedding $\bar{\boldsymbol{h}}_i^{(t)}$ and predicts $\hat{\bar{\boldsymbol{h}}}_j^{(t+1)} = \rho(\bar{\boldsymbol{h}}_i^{(t)})$. The loss is a weight-normalised mean squared error over all immediate and fallback matches:

$$\mathcal{L}_{\text{temp}} = \frac{\displaystyle\sum_{(i,j)} w_{ij}\,\|\hat{\bar{\boldsymbol{h}}}_j - \bar{\boldsymbol{h}}_j^{(t+1)}\|^2}{\displaystyle\sum_{(i,j)} w_{ij}},$$

where each weight w_{ij} is the minimum confidence of the two matched groups, ensuring that geometrically unreliable matches contribute less to the gradient. During training the predictor always receives a real observed embedding as input; chained rollout of predicted embeddings is used only at inference for occlusion bridging.

M.4 Tracker and Fragment Stitching Details

At inference, TrackFish3D outputs full 3D trajectories using a training-faithful tracker: all identity decisions are driven by the same quantities learned during training, namely cross-view correspondence probabilities P_{ij}, identity-discriminative embeddings, and the temporal predictor.

Per-frame multi-view reconstruction. For each frame, we encode detections, predict P_{ij} with the association transformer, form multi-view groups via Hungarian assignment on the learned affinity, and triangulate each group to obtain a 3D hypothesis with a group embedding.

Online temporal linking with predictor-based occlusion bridging. We maintain active tracks storing the last 3D position and embedding. Tracks are matched to current hypotheses with Hungarian assignment using the same distance–similarity cost and gap-dependent gate relaxation used during training-time fallback matching. When a track is missing for g frames, we roll its group embedding forward with the trained predictor, $\bar{\boldsymbol{h}}^{(t+g)} \approx \rho^g(\bar{\boldsymbol{h}}^{(t)})$, and use this predicted embedding for matching.

Track-fragment stitching using learned representations. After the online pass, we stitch non-overlapping track fragments by comparing their boundary embeddings in the same learned space, additionally testing $\rho^g(\bar{\boldsymbol{h}}^A_{\text{last}})$ against $\bar{\boldsymbol{h}}^B_{\text{first}}$ across the gap g. Fragments are merged only when embedding similarity and 3D boundary proximity exceed fixed thresholds. This step introduces no new learned parameters and no extra supervision.

Post-processing. For presentation, we optionally interpolate short gaps between two already-associated observations of the same track and remove very short-lived tracks as ghosts; these operations do not create new identity correspondences and are applied to baselines for fairness.

Tab. 19: Summary of baseline families, capability profiles, and the fair 3D tracking protocol. **Temp** = learned temporal association; **Spat** = learned spatial (cross-view) association; **No Sup** = no identity supervision required; **No Vis** = no visual appearance features used. All methods receive the same inputs (multi-view detections and calibrated cameras) and use the same post-processing for fairness.

Family	Methods	Temp	Spat	No Sup	No Vis	How 3D tracks are produced
Det. + geometric MVA	SAM3 / YOLO26+SAM2 + {Hung., Greedy}	–	–	✓	✓	Geometric MVA → triangulate → NN linking
Det. + geometric MVA + Kalman	SAM3 + 3D-SORT	–	–	✓	✓	Geometric MVA → triangulate → Kalman tracking
Det. + learned MVA	Self-MVA	–	✓	✓	–	Learned MVA → triangulate → NN linking
	ASNet	–	✓	–	–	Learned MVA → triangulate → NN linking
2D MOT → 3D	SambaMOTR, MOTIP	✓	–	–	–	2D tracking per view → MVA at t=0 → triangulate
Multi-camera MOT	ReST, MCTR	✓	✓	–	–	Cross-view 2D tracking → triangulate
Ours	TrackFish3D	✓	✓	✓	✓	Learned grouping + triangulation; temporal linking

N Extended Experiment

This appendix contains only the supplementary material explicitly referenced from the main Results section.

N.1 Datasets and difficulty characterisation

We evaluate TrackFish3D on the synthetic **SynFish** benchmark and the real-world zebrafish benchmark 3D-ZeF [47]. All datasets provide calibrated cameras; SynFish additionally provides ground-truth 3D trajectories for quantitative evaluation.

SynFish dataset. The SynFish corpus contains 14 scenes with visually near-indistinguishable individuals, spanning 4–16 fish, rendered at 1920×1080 with C=3 synchronised cameras. The scenes use 7 distinct camera-rig configurations; the training set spans 6 rigs and the test set uses 4, one of which is unseen during training (Appendix F). We split these into 8 training scenes used for self-supervised learning and 6 held-out test scenes used only for evaluation.

Real-world zebrafish (3D-ZeF). 3D-ZeF uses C=2 calibrated cameras at 2704×1520 resolution with 2 or 5 fish. We use zebra1 and zebra3 for training and zebra2 and zebra4 for testing; baselines that require supervision are finetuned on the training split, while TrackFish3D remains self-supervised on the corresponding split.

Occlusion difficulty score. To quantify difficulty beyond fish count, we report an occlusion score (OccScore), defined as the average, over frames and views, of the maximum pairwise 2D bounding-box IoU in that frame-view. We also report %Overlap, the percentage of frame-view instances containing any overlap (IoU$>$0.01). Full per-scene difficulty statistics are provided in Appendix Table 8, and camera calibration parameters are listed in Appendix F.

N.2 Baselines and fair 3D tracking pipelines

No prior method solves dense multi-camera 3D tracking of visually homogeneous fish end-to-end without identity supervision, so we evaluate 13 complete 3D tracking pipelines assembled from strong components.

Detection + geometric MVA + 3D nearest-neighbour linking. We pair SAM 3 [9] and YOLO26+SAM2 [32, 51] with geometric multi-view association using Hungarian [34] or greedy matching. Costs are derived from epipolar geometry and reprojection consistency; matched detections are triangulated into 3D points and linked over time by nearest-neighbour 3D distance.

Detection + geometric MVA + 3D Kalman tracking. The 3D-SORT baseline uses the same SAM3 detections and Hungarian cross-view associations, then replaces nearest-neighbour temporal linking with the constant-velocity Kalman filtering and gated assignment described in Appendix Q.

Detection + learned MVA + 3D nearest-neighbour linking. We additionally evaluate *Self-MVA* [12] and *ASNet* [7]. For each learned MVA method, we report the stronger of the SAM3 and YOLO26+SAM2 detector variants, while keeping the same nearest-neighbour 3D linking stage.

Single-view 2D MOT → 3D. We evaluate *SambaMOTR* [54] and *MOTIP* [22], then lift their per-view 2D tracks to 3D by performing Hungarian multi-view association at the first frame and keeping that cross-view ID mapping fixed during triangulation.

Multi-camera MOT → triangulate. We evaluate *ReST* [13] and *MCTR* [43], whose outputs already provide cross-view-consistent 2D tracks that can be triangulated directly into 3D trajectories.

Finetuning / retraining protocol. All external components are finetuned or retrained on the SynFish and 3D-ZeF training sets using their default configurations, including any required re-identification backbone or feature extractor. TrackFish3D is trained self-supervisedly on the corresponding training split using SAM3 or YOLO26+SAM2 detections.

Shared post-processing and fairness. All pipelines use the same post-processing: short-gap interpolation between already-associated observations of the same track and removal of very short-lived ghost tracks. These steps use no privileged information and create no new identity correspondences.

N.3 Implementation details

TrackFish3D is trained for 400 epochs with a 100-epoch warmup using only calibrated cameras and 2D observations, without identity labels or ground-truth 3D trajectories. The model comprises a geometric encoder, a 4-layer association transformer with RoPE, and a temporal predictor, totalling about **655K** trainable parameters (Appendix B). We use AdamW with learning rate 10^{-4}, weight decay 10^{-4}, gradient clipping at 5, and loss weights λ_{asso}=1.0, λ_{ctr}=0.5, and λ_{temp}=0.25.

Key operational thresholds are as follows: grouping uses an association probability threshold θ_{assoc}=0.6, an embedding cosine-similarity threshold θ_{emb}=0.55, and a hybrid affinity $0.6\,P + 0.4\,\cos$; temporal linking uses a Hungarian cost $\alpha\, d_{3\text{D}} + \beta\,(1-\cos)$ with α=0.6, β=0.4, gated by a distance threshold θ_{dist}=0.5 and a similarity threshold θ_{sim}=0.6. Complete hyperparameters, including fallback matching and the full training schedule, are listed in Appendix Table 7. Inference uses the training-faithful tracker from Sec. 3.8, with dynamic track birth/death and predictor-based embedding rollforward; further inference details are given in Appendix D. All reported results use detections from SAM 3 or YOLO26+SAM2 rather than ground-truth boxes, and a controlled ablation with additional synthetic detection noise is provided in Appendix Table 17.

O Proof-of-principle downstream analysis on Abnormal Behaviour Detection

Detecting abnormal swimming behaviour is a critical task in aquaculture and marine ecology: sick or stressed fish often exhibit reduced speed, social isolation, and loss of schooling coordination, and early identification can prevent disease spread through the population [14, 35]. However, real-world behavioural annotation is scarce and labour-intensive, motivating automated approaches that operate on 3D trajectories rather than raw video. We show that the 3D trajectories produced by TrackFish3D are sufficiently accurate to support fully unsupervised anomaly detection without behavioural labels.

Experimental setup. We generate a SynFish scene with 8 fish, one of which is programmed as abnormal. The sick fish is animated with reduced swimming speed, erratic heading changes, and diminished schooling tendency, mimicking symptoms of common fish diseases [37]. TrackFish3D is trained self-supervisedly on this scene using the same geometric objectives and training protocol as the main paper, with no identity labels and no ground-truth 3D trajectories. We then run inference to produce 3D trajectories and apply a sliding-window unsupervised anomaly detection pipeline.

Trajectory feature extraction. We adopt a frame-level evaluation using a sliding window of $W{=}3$ frames with stride 1, yielding 359 overlapping windows. Within each window, we extract a 9-dimensional feature vector for every tracked individual: mean speed, speed standard deviation, maximum speed, total path length, straightness (net displacement / path length), mean absolute acceleration, mean distance to group centroid, standard deviation of centroid distance, and mean nearest-neighbour distance. These features capture the key behavioural signatures of sick fish: reduced locomotion, spatial isolation, and failure to coordinate with the school [46].

Anomaly detection methods. At each window position, we apply three unsupervised anomaly detectors to the standardised feature vectors: (i) Isolation Forest [36] with $n_{\text{estimators}}{=}200$, (ii) Local Outlier Factor (LOF) [5] with $k{=}5$, and (iii) a domain-informed z-score on the two most discriminative features, mean speed and mean centroid distance, flagging individuals with combined z-score greater than 1.5. We also report a consensus prediction via majority vote across the three methods. Short-lived ghost tracks with duration below 30% of total frames are filtered before analysis.

Results. Table 20 reports per-method detection accuracy averaged across all sliding windows, and Fig. 8 visualises the trajectory-derived features. The z-score detector achieves the highest window-level F1 of 93.9%, correctly identifying the sick fish in 92.2% of all windows. Isolation Forest and LOF perform comparably at 78.8% and 80.2% F1, respectively. Consensus voting across the three methods yields 80.8% F1, showing that even with only 3 frames of observation, the anomalous individual can be detected reliably in most temporal windows.

Comparison with 2D-only baselines. To verify that 3D trajectory reconstruction genuinely contributes to anomaly detection, we repeat the same sliding-window pipeline using 2D bounding-box trajectories from each individual camera view. All 2D detections are produced by the same finetuned SAM 3 model used by TrackFish3D; the only difference is that the 2D baselines operate on per-view detections independently, whereas TrackFish3D fuses detections across all three views into 3D trajectories. Because the 2D detections carry no identity labels, we construct tracks via nearest-neighbour linking of bbox centres across consecutive frames and identify the sick track by projecting the known 3D ground-truth position into each camera view.

Every camera view yields substantially lower scores than the 3D baseline: the best single-view consensus F1 is only 52.1% (Cam 0) versus 80.8% for 3D trajectories, and the best single-view z-score F1 is 85.2% (Cam 0) versus 93.9% for 3D. Performance varies strongly across viewpoints: Cam 1 achieves 0.0% z-score F1 because the sick fish's projected 2D motion in that view does not differ sufficiently from healthy individuals, and Cam 2 drops to 25.9% consensus F1. These results confirm that 2D bounding-box features from any single camera view are unreliable for anomaly detection, whereas calibrated 3D reconstruction recovers view-invariant behavioural features that enable consistent detection across windows.

Discussion. The sick fish exhibits roughly half the mean speed of healthy individuals and markedly higher centroid distance, producing a clear kinematic signature even within very short observation windows. The z-score detector, which directly targets the two most discriminative features, is

Tab. 20: Frame-level abnormal fish detection: 2D bounding-box trajectories from each individual camera view versus 3D trajectories from TrackFish3D. A sliding window of $W=3$ frames is moved across the sequence (359 windows). We report exact-match rate, precision, recall, and F1 averaged over all windows. 2D features are extracted from bbox-centre tracks linked by nearest-neighbour association. Consensus = majority vote of Isolation Forest, LOF, and z-score.

Input	Method	Exact%	Prec.	Rec.	F1
2D Cam 0	Iso. Forest	48.2%	48.2%	48.2%	48.2%
	LOF	45.7%	45.7%	45.7%	45.7%
	Z-score	84.7%	85.1%	85.5%	85.2%
	Consensus	52.1%	52.1%	52.1%	52.1%
2D Cam 1	Iso. Forest	51.8%	51.8%	51.8%	51.8%
	LOF	44.0%	44.0%	44.0%	44.0%
	Z-score	0.0%	0.0%	0.0%	0.0%
	Consensus	42.9%	42.9%	42.9%	42.9%
2D Cam 2	Iso. Forest	36.5%	36.5%	36.5%	36.5%
	LOF	25.9%	25.9%	25.9%	25.9%
	Z-score	12.3%	12.3%	12.3%	12.3%
	Consensus	25.9%	25.9%	25.9%	25.9%
3D (Ours)	Iso. Forest	78.8%	78.8%	78.8%	78.8%
	LOF	80.2%	80.2%	80.2%	80.2%
	Z-score	92.2%	93.5%	94.7%	93.9%
	Consensus	80.8%	80.8%	80.8%	80.8%

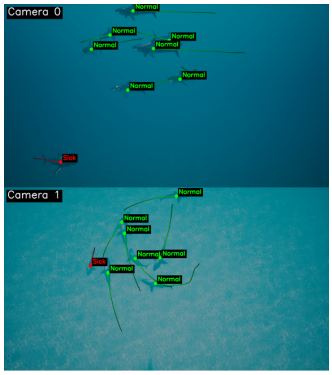


Fig. 8: Anomaly detection on the 8-fish SynFish scene (1 sick fish). Trajectory-derived behavioural features—mean speed and mean centroid distance—cleanly separate the anomalous individual (red) from healthy fish (green), using only 3D trajectories predicted by TrackFish3D with no behavioural labels.

the most robust method, while Isolation Forest and LOF are occasionally confused by natural behavioural variation among healthy individuals. Importantly, these results are obtained with a window of only 3 frames, showing that TrackFish3D trajectories preserve fine-grained behavioural information sufficient for near-real-time anomaly detection. While the current evaluation uses simulated abnormalities, the same pipeline is directly applicable to real aquaculture data where calibrated multi-camera rigs are increasingly common.

Tab. 21: Cross-species evaluation on the held-out Pigeon10 sequence `seq48` from 3D-POP. TrackFish3D is trained on four real-world pigeon sequences and evaluated on the fifth. Results show that the same geometry-driven formulation transfers beyond fish to another visually similar animal collective.

Method	MOTA ↑	Recall ↑	Prec. ↑	IDF1 ↑	IDSW ↓	Frag ↓	MT ↑	PT ↓	ML ↓
Greedy	86.0%	92.6%	95.2%	63.3%	60	36	8	2	0
Hungarian	91.5%	93.9%	97.6%	92.0%	3	6	**10**	**0**	0
ASNet [7]	85.6%	91.6%	95.3%	63.0%	46	40	9	1	0
Self-MVA [12]	68.7%	78.4%	90.7%	47.8%	50	61	6	4	0
ReST [13]	88.3%	91.5%	**98.5%**	58.0%	56	35	8	2	0
MCTR [43]	28.3%	58.2%	67.7%	35.7%	66	87	1	7	2
SambaMOTR [54]	74.0%	79.1%	94.0%	85.8%	5	36	6	3	1
MOTIP [22]	90.4%	94.1%	96.3%	**95.1%**	**1**	6	9	1	0
TrackFish3D	**91.9%**	**94.5%**	97.4%	91.9%	3	**4**	9	1	0

P Cross-species generalisation to real-world pigeons

This appendix provides the full setup and complete results for the pigeon transfer experiment summarised in Sec. 4.5.

Experimental setup. To test whether the same geometry-driven training recipe extends beyond fish, we evaluate TrackFish3D on the real-world **Pigeon10** subset from 3D-POP [39], a four-camera benchmark for markerless 2D–3D tracking of freely moving birds. We use four sequences for training and hold out one sequence (`seq48`) for evaluation. The model is optimised on the training split with the same self-supervised objectives used throughout this paper, then tested on the held-out sequence without any bird identity labels.

Baselines. For comparison, we evaluate the same association baselines used elsewhere in the paper: geometric matching with greedy and Hungarian assignment, the supervised association model ASNet, the self-supervised appearance-based method Self-MVA, the single-view MOT baselines SambaMOTR and MOTIP, and the multi-camera trackers ReST and MCTR.

Discussion. These results suggest that the core assumption behind TrackFish3D is not fish-specific. When individuals are visually similar and multi-view geometry is available, the same self-supervised formulation can transfer to other animal species—here, freely moving pigeons—without requiring any appearance-based identity supervision. In particular, TrackFish3D achieves the best MOTA (91.9%) and recall (94.5%), while reducing fragmentation from 6 to 4 relative to the strongest competing baselines. MOTIP attains the strongest IDF1 (95.1%) and the lowest IDSW count (1), but TrackFish3D remains the top overall method on detection-and-tracking accuracy.

Q 3D-SORT Kinematic Baseline

To compare the learned temporal predictor against standard kinematic filtering, we implement a 3D adaptation of SORT [4]. Per-frame cross-view association is identical to the SAM3 + Hungarian baseline: detections are matched using calibrated triangulation and pairwise reprojection cost. Each resulting 3D track is then represented by a constant-velocity Kalman filter. The filter predicts the next 3D position, and unmatched incoming groups are assigned to predicted positions by gated nearest-neighbour matching. Unmatched tracks are maintained for up to 30 frames on SynFish and 3 frames on 3D-ZeF.

Tab. 22: Comparison with the 3D-SORT kinematic baseline.

Dataset	Method	MOTA ↑	MT% ↑	ML% ↓	IDSw ↓	Frag ↓	MTBF ↑
SynFish	SAM3 + Hungarian	83.5%	71.9%	0.0%	7.5	12.8	183.8
SynFish	3D-SORT	87.7%	77.1%	0.0%	4.0	9.0	209.4
SynFish	TrackFish3D (SAM3)	**95.8%**	**95.8%**	**0.0%**	**0.0**	**6.5**	**266.3**
3D-ZeF	Hungarian	77.4%	90.0%	0.0%	6.0	73.0	42.5
3D-ZeF	3D-SORT	75.6%	90.0%	0.0%	13.5	65.5	43.4
3D-ZeF	TrackFish3D	**81.1%**	**90.0%**	**0.0%**	**4.0**	**30.5**	**93.6**

The Kalman filter improves the geometric baseline on SynFish by bridging short gaps under approximately smooth motion, but remains 8.1 percentage points below TrackFish3D (SAM3). On 3D-ZeF, constant-velocity prediction reduces fragmentation but increases identity switches and lowers MOTA. When visually identical fish cross or turn unpredictably, a predicted position may become closer to a different fish, causing an identity swap. In contrast, TrackFish3D predicts in a contextual embedding space that represents each fish relative to the full scene rather than relying only on individual linear kinematics.

R Density and Computational Scaling

The original SynFish benchmark contains at most 16 fish. We additionally render one 30-fish test scene and evaluate the unchanged SAM3-based model and direct geometric baseline:

Tab. 23: Single-scene 30-fish SynFish stress test.

Method	MOTA ↑	MT% ↑	IDSw ↓	Frag ↓	MTBF ↑
TrackFish3D (SAM3)	**70.2%**	**63.3%**	**8**	**22**	**143.7**
SAM3 + Hungarian	29.9%	6.7%	40	54	36.0

Let N be the total number of detections from all cameras in a frame and d the embedding dimension. Global self-attention scales as $O(N^2 d)$ and stores N^2 pairwise attention entries. Assuming three detections per fish across the three-camera SynFish setup, the approximate scaling is:

Tab. 24: Approximate attention scaling with fish count for the three-camera SynFish configuration.

Fish	Detections per frame (N)	Attention entries (N^2)
16	48	2,304
30	90	8,100
32	96	9,216
64	192	36,864
100	300	90,000

Increasing from 16 to 30 fish therefore increases the attention matrix by approximately $3.5\times$, but 8,100 entries remain modest. Computation may become relevant for substantially larger schools, where sparse or hierarchical attention would be appropriate. At the evaluated densities, the more immediate limitation is geometric: higher density increases mutual occlusion, missed detections, and the number of plausible cross-view correspondences. If too few fish remain visible in at least two cameras, reliable triangulation becomes impossible regardless of the association model. The 30-fish result in Table 23 is consequently a synthetic single-scene stress test, not a universal capacity threshold or evidence of real high-density generalisation.

S Pseudo-Label Noise Audit

We directly audit geometry-derived pseudo-positive pairs against SynFish ground-truth identities. Ground truth is used only for this diagnostic evaluation and is never provided during training. A

pseudo-positive is counted as correct when both detections belong to the same ground-truth fish identity.

Tab. 25: Pairwise quality of geometry-derived pseudo-positive associations as density increases.

Fish	Precision ↑	Recall ↑
4	70.8%	100.0%
16	25.3%	100.0%
30	17.0%	100.0%

Precision falls substantially as geometrically plausible incorrect pairs become more common, while recall remains complete in the evaluated scenes. Thus, we do not claim that pseudo-labels are clean or intrinsically stable under density. Instead, they provide noisy, high-coverage supervision aggregated across detections, views, and frames. The learned objectives exploit this repeated signal, while direct geometric matching treats each ambiguous association as a final decision.

T Training-Time Calibration Noise

Appendix J perturbs calibration only at inference while training on clean geometry. To separately evaluate calibration uncertainty during pseudo-label generation and geometric feature construction, we conduct a controlled 3D-ZeF experiment in which the training camera extrinsics receive Gaussian rotation and translation perturbations with $\sigma_R{=}\sigma_t{=}0.1$. We regenerate all geometry-derived supervision under the perturbed calibration, train TrackFish3D on these data, and evaluate using the original clean calibration and ground truth.

Tab. 26: 3D-ZeF performance after training with perturbed calibration and evaluating with clean calibration.

Training calibration	MOTA ↑	MT% ↑	ML% ↓	IDSw ↓	Frag ↓	MTBF ↑
Clean	81.1%	90.0%	0.0%	4.0	30.5	93.6
Noisy, $\sigma{=}0.1$	66.8%	60.0%	0.0%	13.5	35.5	59.0

Training-time calibration noise causes a clear degradation relative to clean training, so this experiment does not establish calibration invariance. Nevertheless, performance does not collapse and no trajectories become mostly lost. The result supports the narrower conclusion that useful tracking performance remains under this perturbation while confirming calibration quality as a practical limitation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) state that TrackFish3D is a self-supervised framework for dense multi-camera 3D tracking of visually homogeneous fish, achieving 96.6% MOTA on SynFish and 81.1% on 3D-ZeF. These claims are supported by the experimental results in Section 4, notably Tab. 2 and Tab. 3, with method-family context summarised in Tab. 1.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The paper discusses limitations and future directions in the conclusion, including the reliance on calibrated cameras, sensitivity to calibration errors and missed detections, and the need for evaluation on larger and more diverse real-world datasets. These limitations are further substantiated by the robustness analyses in Appendix J and Appendix K.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not contain theoretical results or proofs. It is an empirical systems paper.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The appendix provides the full training algorithm (Algorithm 1), complete network architecture with parameter counts, all hyperparameters (Table in Appendix), the inference pipeline, camera calibration parameters, and dataset statistics. These details are sufficient to re-implement the system.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and the SynFish dataset will be released upon acceptance. The 3D-ZeF dataset is publicly available [47]. An anonymous code repository is provided in the supplementary material.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 and the appendix specify train/test splits, while Appendix C and Appendix D specify the AdamW optimiser, learning rate, weight decay, gradient clipping, loss weights, grouping thresholds, temporal linking parameters, and checkpoint selection criteria.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The tracking pipeline is deterministic given fixed 2D detections (no random initialisation), so there is no run-to-run variance to report. Per-scene results are provided in the appendix to show performance variation across scenes of different difficulty.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The appendix reports training time (17h 36m), inference speed (124.8 ms/frame), peak GPU memory (0.02 GB), and the hardware used (NVIDIA A100 80GB).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research involves tracking fish in synthetic and controlled laboratory environments. No human subjects, personally identifiable information, or ethically sensitive data are involved.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses positive impacts in automated health monitoring for aquaculture and behavioural ecology. The technique is domain-specific (fish tracking in tanks) and does not raise surveillance or privacy concerns for humans.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released model is a small (655K parameter) geometric tracking network for fish in aquaria. It poses no foreseeable risk for misuse.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The 3D-ZeF dataset [47] is properly cited. All baseline methods are cited with their original publications.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The SynFish dataset is documented in Section 4.1 and the appendix, including scene counts, fish counts, frame counts, camera configurations, and difficulty statistics.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects are involved. The zebrafish recordings used (3D-ZeF) are from a previously published and publicly released benchmark.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of the core methodology.