MC3DTrackNet: End-to-End Multi-Camera 3D Object Detection and Tracking in Autonomous Driving

Sai Manideep M, Brindha M
Department of Computer Science and Engineering, National Institute of Technology,
Tiruchirappalli 620015, Tamil Nadu, India

***Abstract*—The emergence of autonomous vehicles has accelerated the demand for precise and effective 3D perception systems that can interpret intricate traffic environments from visual inputs only. Though LiDAR-based pipelines have long ruled 3D object detection and tracking, increasing interest now exists in achieving this using only multi-camera inputs. However, camera-based 3D perception poses serious challenges, including lack of explicit depth cues, variable object visibility over views, and occlusions in urban environments.**

To address these, MC3DTrackNet is proposed — a lightweight and interpretable single-stage framework for end-to-end 3D multi-object tracking from RGB image streams of surround-view camera systems. MC3DTrackNet avoids the use of LiDAR, BEV transformation modules, object queries, or temporal attention mechanisms. Instead, it utilizes a ResNet-50 backbone for image-wise feature extraction across six synchronized cameras, followed by a simple yet effective average pooling strategy for multi-view fusion.

The fused representation is passed to detection heads working on a proposed grid-based spatial structure, directly predicting global 3D bounding box parameters [x,y,z,w,l,h,] per cell. Ground truth labels are mapped to grid-aligned targets via normalized coordinate mapping, enabling each cell to learn to localize at most one object.Tracking is performed using a greedy association strategy based on Euclidean distances between estimated 3D centers and motion-compensated track positions. Training is conducted end-to-end using RGB-only supervision with focal loss and Smooth L1 loss.Empirical results demonstrate strong tracking performance on the nuScenes dataset, with high AMOTA and AMOTP scores, fewer identity switches, and low computational cost, making it suitable for real-time deployment.

Camera-based 3D multi-object tracking has been an inexpensive LiDAR replacement for autonomous vehicle perception, but remains occlusion-prone, depth-ambiguous, and computationally costly. MC3DTrackNet offers a LiDAR-free, interpretable, and low-cost pipeline that does not include object query, LiDAR, or attention components. With a grid-based Bird's Eye View (BEV) representation and simple multi-camera fusion, the system offers efficient, end-to-end 3D tracking from RGB inputs alone. Experiments on the nuScenes benchmark demonstrate competitive performance on AMOTA, AMOTP, and fewer identity switches and much lower memory and runtime requirements. The simplicity of the architecture promises faster generalization and convergence, and deployment in real-world autonomous driving systems is made possible, particularly in scenarios with limited annotation budgets or hardware constraints.

***Index Terms*—Autonomous driving,Multi-object multi-camera tracking perception, nuScenes dataset,ResNet, Vision-based tracking, 3D object detection.**

I. Introduction

Precise 3D object detection and multi-object tracking are the core elements of perception systems for autonomous vehicles (AVs). These processes enable cars to perceive and respond to dynamic environments in real time by detecting pertinent objects and preserving their identities across time. With advances in deep learning, contemporary computer vision models have greatly enhanced the accuracy, efficacy, and resilience of object perception in diverse difficult environments. Yet, with all these progress, most AV perception systems still make use of LiDAR sensors or a single monocular camera. While LiDAR allows accurate depth estimation at a high cost of hardware and computational requirements, monocular systems have inherent depth ambiguity and low field of view and are thus less effective in real-world driving applications.

A potential alternative involves using multi-camera configurations, which are now widespread in contemporary autonomous driving datasets like nuScenes. These arrangements cover a 360-degree panorama of the scene via synchronized cameras placed around the car. Multi-camera inputs have the possibility to address the shortcomings of single-view systems without the expense of LiDAR. However, most of the existing approaches leave such multi-view information underutilized, either by processing each view separately or by making use of sophisticated and computationally expensive Bird's Eye View (BEV) transformation modules for feature alignment. Besides, standard multi-camera tracking systems adopt ideas from surveillance MC-MOT (Multi-Camera Multi-Object Tracking) in which static overlapping camera fields enable Re-ID-based association. In AVs, the scenario is quite different: cameras are mounted on the vehicles, have minimal overlap, and the

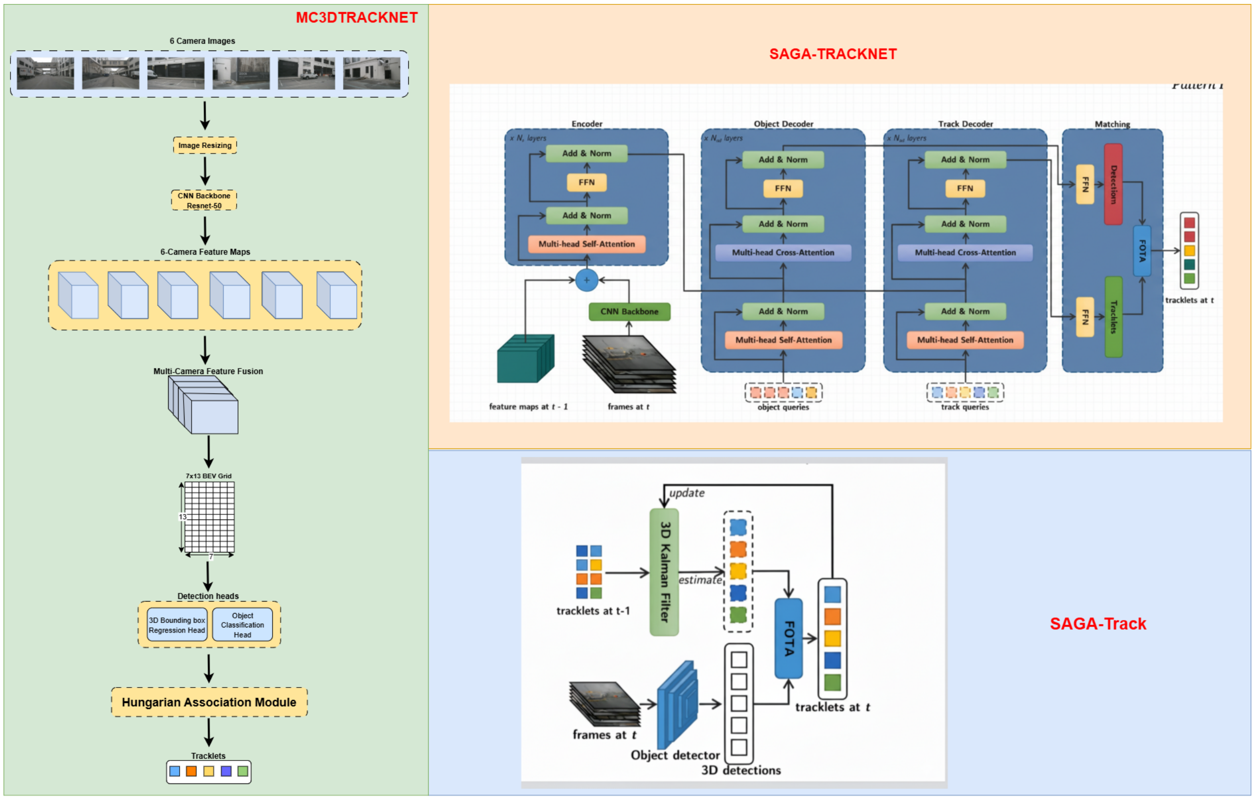

Fig. 1: Architecture of the proposed MC3DTrackNet pipeline, comparison with SAGA-TRACKNET [21] and SAGA-Track [20] pipelines

scenes are extremely dynamic, which renders such Re-ID mechanisms unsuitable.

In addition, standard tracking pipelines usually adopt a two-stage procedure: initial single-camera detection and tracking, followed by a second-stage global ID association with heuristic or appearance-based similarity. Such architectures are likely to fail when there are missing detections, occlusions, or moving objects that change views between frames. They tend to have fragmented tracks and rapid identity changes, particularly in complex or crowded scenes.

A critical challenge in this domain lies in balancing accuracy, efficiency, and scalability of perception systems. Many existing frameworks emphasize accuracy but require computationally expensive architectures such as heavy BEV transformation pipelines or transformer-based multi-view encoders, which may not be feasible for real-time deployment in embedded automotive hardware. On the other hand, lightweight detection or tracking models often struggle to maintain robustness in cluttered urban environments with dense traffic, diverse object categories, and frequent occlusions. Another dimension of complexity arises from the temporal consistency of object identities, where even small prediction errors can cascade into fragmented tracks or mismatches across frames. Addressing these issues demands architectures that are end-to-end trainable, data-efficient, and resilient to missing views or partial occlusions, while still meeting the strict latency requirements of real-world autonomous driving systems.

To overcome these constraints, MC3DTrackNet, an efficient end-to-end single-stage framework for 3D multi-camera multi-object tracking from RGB images only is proposed. The approach is a single-stage model that simulates object detection and tracking in all views simultaneously without requiring the use of intermediate Re-ID modules, hand-crafted association graphs, or view-specific stitching. MC3DTrackNet utilizes a common ResNet-50 backbone for image-level feature extraction from six cameras, combines the resultant features through a naive average pooling module, and conducts direct 3D bounding box regression and classification in global coordinates with a proposed grid-based convolutional head.

The design preserves spatial consistency, eliminates dependence on BEV projections, and maintains the model light for real-time usage. Ground truth is assigned to fixed BEV grid cells in normalized coordinates such that the model can learn structured spatial outputs. 3D center distance and Hungarian matching strategy without appearance cues are used for tracking.Experiments on the nuScenes benchmark indicate that MC3DTrackNet provides robust tracking performance with reduced identity switches and fragmentation, all with a lightweight and scalable architecture that is well-suited to real-world autonomous driving applications.

II. The Design, Intent, and Limitations of the Templates

The templates are intended to **approximate the final look and page length of the articles/papers**. **They are NOT intended to be the final produced work that is displayed in print or on IEEEXplore®**. They will help to give the authors an approximation of the number of pages that will be in the final version. The structure of the LaTeX files, as designed, enable easy conversion to XML for the composition systems used by the IEEE. The XML files are used to produce the final print/IEEEXplore pdf and then converted to HTML for IEEEXplore.

III. RELATED WORK

Multi-object tracking (MOT) has evolved from single-camera setups to complex multi-camera systems. While SC-MOT methods are effective in static scenes, they often fail in dynamic, real-world environments. MC-MOT approaches attempt to address this but face scalability and efficiency challenges. This section reviews key tracking methods and highlights limitations addressed by the proposed framework.

A. SC-MOT Assignment

Single-Camera Multi-Object Tracking (SC-MOT) approaches typically cast tracking as a temporally sequential assignment problem—temporally linking detections by motion and appearance features. Traditional tracking pipelines such as DeepSORT [23] and ByteTrack [8] utilize Kalman filtering to represent object motion and cosine distance between appearance embeddings for re-identification. These approaches have worked effectively for tracking in static monocular environments with relatively stable camera views.

More sophisticated approaches deal with issues such as occlusion and dense scenes. For instance, Xiang et al. [16] proposed an LSTM-affinity model that learns spatiotemporal similarity between two detections. Ran et al. [17] proposed a triple-stream neural network with visual, pose, and motion features to calculate association scores. Affinity models, e.g., these, are usually combined with bipartite graph matching in order to calculate efficient one-to-one matches.

Though they perform adequately under static and single-camera environments, such strategies are severely disadvantaged under dynamic or multi-camera ones. SC-MOT systems typically take a fixed perspective, and they are poor at handling objects observed across variable fields of view—a problem at the heart of autonomous vehicles, which travel in intricate, overlapping multi-camera configurations.

B. Assignment in Static MC-MOT

In static Multi-Camera Multi-Object Tracking (MC-MOT) configurations, e.g., surveillance or smart cities, assignment schemes are constructed to preserve consistent identities between fixed and normally non-overlapping camera views. This is typically done via post-hoc identity linking via appearance or learnt embeddings.

He et al applied matrix factorization to recover global object trajectories from per-camera disjoint tracklets. Ristani and Tomasi [10] applied correlation clustering as a global optimization framework to assign persistent IDs based on affinity scores, irrespective of camera boundaries. Extending this work further, Quach et al. [11] proposed DyGLIP, a dynamic graph learning model constructing context-aware representations across cameras for association and re-identification.

Though effective in static environments, these systems rely on continual lighting, stationary cameras, and high overlap. These assumptions do not hold in dynamic, moving environments such as autonomous vehicles. Specifically, the absence of spatial calibration and motion modeling renders these systems fragile in outdoor high-scale driving scenarios.

C. Motion Models in Dynamic MOT

Motion modeling is a very crucial process in dynamic tracking systems, where temporal consistency of objects is used for predicting future location. Motion models are often used in the tracking pipeline for improving association and missing detection handling in 3D Multi-Object Tracking (3D-MOT).

AB3DMOT [6] is a worldwide baseline using 3D Kalman filtering on LiDAR detections. MonoDIS [7] is also working on monocular inputs with uncertainty-aware loss learned for 3D object location and scale estimation. Kalman filtering with pseudo-LiDAR point clouds from monocular depth prediction was used by Weng et al. [22] without the use of LiDAR and achieved high-quality 3D tracking.

To address uncertainty in state estimation, Mahalanobis distance was applied by Chiu et al. [15] to the association cost matrix to enable stronger matching under sensor noise or detection jitter. Such models are still single-camera vision system optimized and thus limited in their integration in camera-only dynamic AV systems.

D. Appearance-Based Tracking

Appearance features become essential to employ in matching when motion cues are weak—especially when occlusion or random motion of objects occurs. SC-MOT algorithms typically employ motion prediction with deep feature embeddings to compute affinity scores.

CenterTrack [2] is a joint detection-and-tracking system shared to be utilized for joint detection-and-tracking, inferring object motion directly by regressing displacements of center-points between frames in the absence of explicit Re-ID features. Zhou et al. [18] represented objects as points and reduced tracking to predicting centers. Hu et al. [4] generalized this with a decoder enhanced with an LSTM for strong monocular 3D tracking in difficult situations.

While such techniques are effective in SC-MOT, they are limited in MC-MOT. With dramatic viewpoint changes in a multi-camera environment, object appearance features are unreliable. The features learned from one camera are not correctly mapped to the same object from a different viewpoint, particularly without camera-invariant training or view-consistent features.

E. Hybrid and Transformer-Based Models

The combination of detection and tracking into a single architecture has been tried with the assistance of hybrid models, which have a propensity to use recurrent or transformer layers for modeling temporal relationships. These models offer end-to-end training pipelines that detect objects while also capturing identity association.

Chaabane et al. proposed DEFT [5], a hybrid method that integrates LSTM-based motion reasoning within the object detection network. Yin et al. [14] proposed 4D-Net, a point cloud and image feature combined spatial-temporal backbone for enhanced 3D tracking in autonomous driving.

Transformer-based methods like TrackFormer [12] are based on DETR [26] to incorporate temporal self-attention such that object queries can be stabilized frame-wise. QueryTrack [13] and QD-3DT [3] use key-query attention that decouples detection and tracking into two distinct tasks such that there is more flexibility in target association over time. QD-3DT in particular is vision-only 3D tracking and performs extremely well on nuScenes [1].

Though such transformer models are delivering state-of-the-art performance, their computational and memory expense is not making them as attractive for real-time processing on edge devices or embedded AV platforms.

F. Graph and Learning-from-Context Models

Graph models encode object interactions in space and time, detections as nodes and edges as relations. Graph models are naturally capable of encoding context and long-range dependencies, especially for dense or occluded scenes.

GNN3DMOT [19] is a graph neural network model that builds spatiotemporal graphs between detections and performs graph convolutions to learn associations. Meinhardt et al. [12] introduced the tracking-by-attention paradigm, which combines global temporal context using attention weights, tracking smoothly through occlusions and re-appearances. Sun et al. [13] showed that a key-query attention mechanism can be applied to combine detection and tracking into a learned, context-sensitive framework.

But although they are powerful models, they are associated with large inference latency and complicated training dynamics. Thus, they have found limited application in real-time MC-MOT systems for autonomous vehicles, where processing is required to be rapid and hardware limitations are important.

G. MC-MOT in Autonomous Vehicles

Dynamic MC-MOT, multiple object tracking across multiple moving cameras, is a new and significant research issue in autonomous vehicles. Fewer models have been specifically designed to deal with viewpoint variation, camera overlap, and real-time processing in mobile multicamera systems. SAGA-Track [20] introduced a one-to-many FOTA-based global assignment method that can match a tracked object to numerous detections across cameras. Its geometry-aware assignment solver rectifies Hungarian matching's weaknesses in multi-view scenarios. SAGA-TrackNet [21] also converted this into an end-to-end architecture with a transformer encoder and two decoders (object and track), and a new added matching layer with detection and tracking.

They are currently the baselines of choice of the nuScenes Vision Track Challenge [1], and they perform very well in 3D vision-based MC-MOT. They do so, however, by depending on spatio-temporal attention modules and object queries, thereby increasing inference complexity. Also, their adoption of structured transformer architectures renders them impractical for lightweight deployment.

H. Research Gaps and Contributions of the Proposed Work

The suggested approach, MC3DTrackNet, is an architecturally slim and space-efficient end-to-end camera-only 3D multi-camera multi-object tracker. Although approaches such as SAGA-TrackNet [21] and QD-3DT [3] utilize transformer blocks and attention-based association,this approach utilizes a ResNet-50 [28] feature backbone followed by a SimpleFusion module in which the features are averaged across all the cameras. This in-built innovation is the grid-level regression head, which predicts end-to-end 3D bounding boxes in world coordinates per grid cell without object queries or handcrafted proposals. For identity matching, we leverage a greedy BEV IoU-based matching strategy - much less computationally expensive than FOTA [20], graph-based [19], [11], or attention-based matching strategies [12]. Unlike monocular 3D tracking models like MonoDIS [7] or Monocular 3D + Kalman [6], this method performs global tracking from synchronized multi-camera inputs. Unlike LiDAR-based setups like CenterPoint or PointPillars (not referenced),this method is vision-based and does not involve costly sensor fusion. Essentially, MC3DTrackNet really does make a tough real-world trade-off between speed and accuracy to support real-time 3D MOT on vision-only autonomous vehicles. It has an end-to-end trainable proposed grid-based architecture with no hand-tuned matching rules and attention blocks and that can be efficiently implemented on embedded hardware.

IV. Proposed Method

In this section, **MC3DTrackNet**, a lightweight yet powerful pipeline for 3D Multi-Camera Multi-Object Tracking (MC-MOT) is presented. In contrast to sophisticated transformer-based models, MC3DTrackNet employs a straightforward and interpretable architecture that integrates feature extraction, camera-wise fusion, proposed grid-based 3D object detection, and identity association.

As shown in Figure 2 the BEV grid anchors are used for assigning ground truth boxes, generating cell-wise supervision, and predicting class and 3D bounding box outputs per cell. The fused BEV feature grid from all camera views is processed by parallel detection heads for classification and box regression.

The proposed grid-based detection architecture of MC3DTrackNet transforms the task of 3D object detection into a structured, cell-wise prediction task within a discretized Bird's Eye View (BEV) coordinate space. The idea is to project a fixed-size grid — in this case, 7 rows × 13 columns — onto the ground plane of the ego vehicle within the 3D world. Each grid cell corresponds to a 1m × 1m spatial area centered at the ego vehicle and serves as an independent detection anchor. Ground truth 3D bounding boxes are projected during training onto corresponding grid cells based on the (x, y) center location of the object, normalized by the total spatial extent of the grid. For each cell corresponding to a foreground object, a set of regression targets is calculated, including cell center offset (x,y), absolute height z, size (w,l,h), and orientation , as well as a class label. The rest of the cells are labeled as background. Concurrently, the model produces a fused BEV feature map by extracting high-level features from each of the six camera images using a ResNet-50 backbone and then performing per-pixel average fusion. This produces a 256-channel BEV-aligned feature grid in which each cell now contains semantic information from multiple views. This fused feature grid is fed into two independent detection heads: a classification head that predicts a probability distribution over all classes (including background) per cell, and a regression head that predicts the 3D bounding box parameters. These heads are implemented as simple 1×1 convolutions, acting as dense per-cell predictors. During training, the model is trained using focal loss for classification — for class imbalance handling — and Smooth L1 loss for box regression, applied only to object-assigned cells. During inference, high-confidence cells are decoded by adding the predicted offsets (x,y) to the known grid cell centers to recover global (x, y) positions, and the rest of the box parameters are taken directly from the regression output. The system eliminates the use of anchor boxes or non-maximum suppression, as each grid cell is responsible for detecting at most one object. By anchoring all predictions to fixed spatial cells and fusing multi-view information into one grid.

A. Problem Definition

Let a given scene at time step t be captured by fixed synchronized cameras $K = 6$. These images are denoted as the set:

$$\mathcal{I}^{(t)} = \{I_1^{(t)}, I_2^{(t)}, \dots, I_K^{(t)}\} \quad .....(1)$$

The goal of a Multi-Camera Multi-Object Tracking (MC-MOT) system is to output a set of detected 3D objects at time t, denoted by:

$$\mathcal{O}^{(t)} = \{\mathbf{o}_j^{(t)}\}_{j=1}^{M} \quad .....(2)$$

where each object $\mathbf{o}_j^{(t)}$ is defined by a 3D bounding box $[x, y, z, w, l, h, \theta]$ in the global coordinate system and a semantic class label.

Tracking involves associating these detections across time to generate consistent identities, referred to as tracklets:

$$\mathcal{T}_i = \{\text{tr}_i^{(t_1)}, \text{tr}_i^{(t_2)}, \dots\} \quad .....(3)$$

Each tracklet is updated using a two-step motion estimation approach:

$$\hat{\text{tr}}_i^{(t)} = \mathcal{M}_{\text{pred}}(\text{tr}_i^{(t-1)}), \quad \text{tr}_i^{(t)} = \mathcal{M}_{\text{update}}(\hat{\text{tr}}_i^{(t)}, \mathbf{o}_{[i]}^{(t)}) \quad .....(4)$$

The best matching detection $\mathbf{o}_{[i]}^{(t)}$ is assigned to the predicted track using an IoU-based distance metric, and associations are resolved using the Hungarian algorithm.

B. Proposed BEV Grid Setup

To enable structured 3D reasoning, we overlay a **discretized top-down BEV grid** on the 3D world. Each grid cell represents a 1m × 1m region. For the experiments on nuScenes, the grid size is fixed at:

$$H = 7 \text{ (rows)}, \quad W = 13 \text{ (columns)}$$

Thus, the model makes predictions over a total of 91 spatial anchors. Each grid cell is centered at a known physical location and is responsible for detecting objects whose centers fall within it.

1) Backbone: Multi-View Feature Extraction: Each camera image $I_k^{(t)}$ is processed independently using a ResNet-50 backbone pretrained on ImageNet, yielding a feature map $F_k \in \mathbb{R}^{2048 \times H' \times W'}$. These are projected into a lower-dimensional space via a 1×1 convolution:

$$F_k' = \text{Conv1x1}(F_k), \quad F_k' \in \mathbb{R}^{256 \times H' \times W'} \qquad .....(5)$$

2) SimpleFusion: Camera-wise Mean Fusion: To integrate information from all cameras, the feature maps are averaged:

$$F_{\text{fused}} = \frac{1}{K} \sum_{k=1}^{K} F_k' \qquad .....(6)$$

The resulting $F_{\text{fused}} \in \mathbb{R}^{256 \times H' \times W'}$ is a unified spatial representation used for object detection.

3) Detection Head: Grid-Based 3D Box Regression: The scene is divided into a 7×13 spatial grid. The fused feature map is passed through a convolutional head with two branches:

- **Classification head:** Predicts class probabilities per grid cell. Output shape: $[B, C+1, H, W]$.
- **Regression head:** Outputs 3D bounding box parameters $[x, y, z, w, l, h, \theta]$ per grid cell. Output shape: $[B, 7, H, W]$.

Each cell is responsible for detecting one object, inspired by YOLO-style detection frameworks.

C. Target Assignment and Label Generation

Ground-truth annotations are converted into grid-aligned targets. An object is assigned to the cell whose normalized coordinates match the object's center:

$$g_x = \left\lfloor \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot (W-1) \right\rfloor \qquad .....(7)$$

$$g_y = \left\lfloor \frac{y - y_{\min}}{y_{\max} - y_{\min}} \cdot (H-1) \right\rfloor \qquad .....(8)$$

Each cell is assigned:

- A class label or background (as $\mathbf{Y}_{\text{cls}} \in \mathbb{Z}^{H \times W}$)
- A 3D bounding box ($\mathbf{Y}_{\text{box}} \in \mathbb{R}^{H \times W \times 7}$)

D. Loss Function

A multi-task loss is used:

Focal Loss for Classification: To handle foreground-background imbalance:

$$\mathcal{L}_{\text{cls}} = -\alpha(1-p_t)^{\gamma} \log(p_t) \qquad .....(9)$$

where p_t is the predicted class probability, $\gamma = 2$, and $\alpha = 0.25$.

Smooth L1 Loss for Box Regression: Applied only on positive samples:

$$\mathcal{L}_{\text{box}} = \frac{1}{N_{\text{obj}}} \sum_{i,j \in \text{fg}} \text{SmoothL1}(\hat{y}_{i,j}^{\text{box}}, y_{i,j}^{\text{box}}) \qquad .....(10)$$

Total Loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{box}}, \quad \lambda = 1.0 \qquad .....(11)$$

Algorithm 1 The inference pipeline of MC3DTrackNet

```
Input: Multi-view images $\mathcal{I}^{(t)} = \{I_1^{(t)}, \dots, I_K^{(t)}\}$, previous tracks $\mathcal{T}$, tolerance $t_{\text{tol}} = 30$
Output: Updated tracks $\mathcal{T}$
$t \leftarrow -1$
$\mathcal{T}_{\text{inactive}} \leftarrow \emptyset$
while true do
    $t \leftarrow t + 1$
    Draw image set $\mathcal{I}^{(t)}$ from $\mathcal{S}$
    if $t = 0$ then
        features $\leftarrow$ Encoder($\mathcal{I}^{(t)}$)
        features$_{\text{fused}}$ $\leftarrow$ MeanFusion(features)
        $\mathcal{O}^{(t)} \leftarrow$ Grid3DDetector(features$_{\text{fused}}$)
        $\mathcal{T} \leftarrow$ initialize($\mathcal{O}^{(t)}$)
    else
        $\mathcal{T}_{\text{query}} \leftarrow \mathcal{T} + \mathcal{T}_{\text{inactive}}$
        features $\leftarrow$ Encoder($\mathcal{I}^{(t)}$)
        features$_{\text{fused}}$ $\leftarrow$ MeanFusion(features)
        $\mathcal{O}^{(t)} \leftarrow$ Grid3DDetector(features$_{\text{fused}}$)
        (matched, unmatched) $\leftarrow$ greedyIoU($\mathcal{O}^{(t)}, \mathcal{T}_{\text{query}}$)
        $\mathcal{T} \leftarrow$ update($\mathcal{T}$, matched)
        $\mathcal{T} \leftarrow$ initialize(unmatched)
        $\mathcal{T}_{\text{inactive}} \leftarrow$ remove_inactive($\mathcal{T}_{\text{inactive}}, t_{\text{tol}}$)
    end if
end while
```

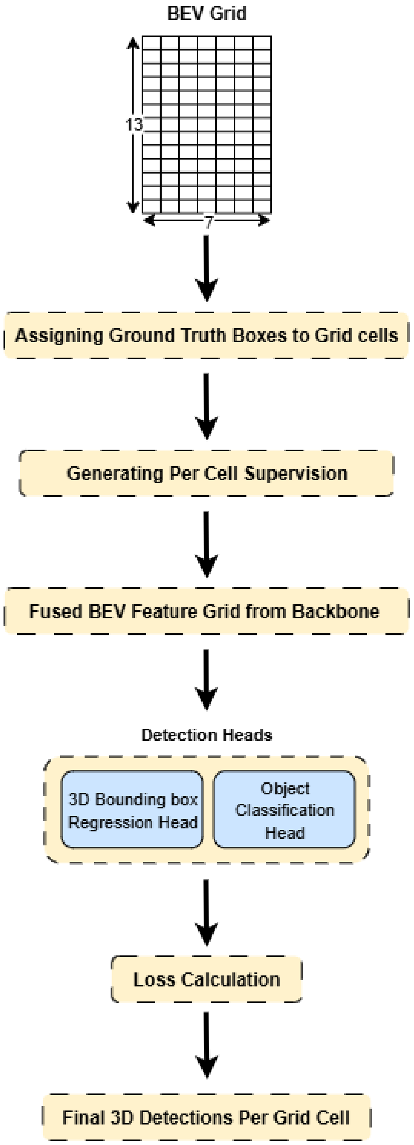
BEV Grid
13
7
Assigning Ground Truth Boxes to Grid cells
Generating Per Cell Supervision
Fused BEV Feature Grid from Backbone
Detection Heads
3D Bounding box Regression Head
Object Classification Head
Loss Calculation
Final 3D Detections Per Grid Cell

Fig. 2: Step-by-step workflow of the proposed grid-based BEV detection and supervision process.

E. Tracking and Identity Association

MC3DTrackNet takes a light and efficient *tracking-by-detection* approach to connect objects detected across time. Contrary to transformer-based architectures that depend on learned temporal components, this approach draws upon a deterministic, greedy matching mechanism that guarantees real-time operation with low overhead.

At the beginning of a scene (time step $t = 0$), all detections that are confident are utilized to create new tracklets.Each detection is assigned a global ID and make it an active track. In the following frames, we advance existing tracks in time by a simple linear motion model. In particular, each track's state at time t is given by:

$$\hat{\text{tr}}_i^{(t)} = \text{tr}_i^{(t-1)} + \Delta t \cdot \mathbf{v}_i \qquad .....(12)$$

where $\mathbf{v}_i$ is the estimated velocity computed using historical positions, and Δt is the fixed frame interval (0.5 seconds in the nuScenes dataset).

For each frame, the model produces a set of 3D detections $\{\mathbf{o}_j^{(t)}\}$ and a set of predicted track positions $\{\hat{\text{tr}}_i^{(t)}\}$. To perform data association, a cost matrix is constructed using the 2D Intersection-over-Union (IoU) between predicted tracks and current detections in the Bird's Eye View (BEV) plane. The cost function is defined as:

$$\mathrm{cost}(i,j) = 1 - \mathrm{IoU}_{\mathrm{BEV}}(\hat{\mathrm{tr}}_i^{(t)}, \mathbf{o}_j^{(t)}) \quad .....(13)$$

This cost matrix is fed to Hungarian algorithm, which calculates the one-to-one assignment of lowest overall association cost.

Having formed associations among detections and available tracklets, object identities are updated by a set of well-specified rules. For matching pairs, the related tracklet is updated with the new detection, its position and appearance being further refined. Unmatched detections are considered to be new objects and utilized to create new tracklets with global identifiers that are unique. Unmatched tracks, on the other hand, are not discarded at once; rather, they are relocated to an inactive buffer where they are kept for a grace period (e.g., 30 successive frames). The buffer enables re-identification upon temporary occlusion, missed detection, or transient departures from the field of view of the camera. If a tracklet is unmatched outside the buffer window, it is treated as terminated. This lifecycle management policy allows the system to ensure identity consistency over extended temporal intervals, manage object occlusions and reappearances smoothly, and reduce identity switches in complicated multi-camera scenes.

This approach ensures real-time tracking performance without requiring additional network modules or costly optimization.

F. Numerical Use-Case Illustration: End-to-End Tensors in MC3DTrackNet

To concretely illustrate the internal working of MC3DTrackNet, a complete numerical walkthrough using actual tensors is presented. This example demonstrates how the 7×13 BEV grid (1m × 1m per cell) processes synchronized 6-camera inputs and outputs 3D bounding boxes with identities.

a) Step 1: Multi-Camera Image Input.: Each RGB image from the 6 cameras is of size $3 \times 256 \times 704$. The batched input tensor is:

$$\mathbf{I}_{\text{input}} \in \mathbb{R}^{6\times 3\times 256\times 704}$$

For instance, pixel intensities for Camera 1 are normalized to:

$$\mathbf{I}_{\text{input}}[0,:,:,:] = \begin{bmatrix} 0.12 & 0.13 & \dots \\ 0.23 & 0.21 & \dots \\ 0.34 & 0.30 & \dots \end{bmatrix}$$

b) Step 2: Feature Extraction (ResNet-50).: Each image is passed through a shared ResNet-50 CNN yielding:

$$\mathbf{F}_i \in \mathbb{R}^{256\times 32\times 88}, \quad i = 1 \dots 6$$

An example slice:

$$\mathbf{F}_1[0,:,:] = \begin{bmatrix} 0.33 & 0.12 & \dots \\ 0.30 & 0.10 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

c) Step 3: Simple Fusion.: The 6 camera features are averaged to produce:

$$\mathbf{F}_{\text{fused}} = \frac{1}{6}\sum_{i=1}^{6} \mathbf{F}_i \in \mathbb{R}^{256\times 32\times 88}$$

For example:

$$\mathbf{F}_{\text{fused}}[0,0,0] = \frac{1}{6}\sum_{i=1}^{6} \mathbf{F}_i[0,0,0] = 0.29$$

d) Step 4: BEV Projection.: Downsampling to the grid resolution:

$$\mathbf{F}_{\text{bev}} \in \mathbb{R}^{256\times 7\times 13}$$

A feature vector at cell $(3,8)$:

$$\mathbf{F}_{\text{bev}}[:,3,8] = [0.12, 0.15, 0.31, \dots, 0.09]$$

e) Step 5: Detection Head.: **(a) Classification:**

$$\mathbf{C}_{\text{cls}} \in \mathbb{R}^{10\times 7\times 13}$$

At grid cell (3,8):

$$\mathbf{C}_{\text{cls}}[:,3,8] = [0.01, \mathbf{0.92}, 0.03, 0.01, \dots] \Rightarrow \text{Class: Car}$$

(b) Regression:

$$\mathbf{R}_{\text{reg}} \in \mathbb{R}^{7\times 7\times 13}$$

At grid cell (3,8):

$$\mathbf{R}_{\text{reg}}[:,3,8] = [8.5, 3.5, 0.9, 1.8, 4.2, 1.6, 0.0]$$

f) Step 6: Predicted Detections.: After thresholding and decoding:

- **(3,8)** → Car, $\mathbf{x} = [8.5, 3.5, 0.9]$, size $[1.8, 4.2, 1.6]$, yaw 0.0
- **(2,6)** → Pedestrian, $\mathbf{x} = [6.5, 2.5, 1.6]$, size $[0.6, 0.8, 1.7]$, yaw -0.1

g) Step 7: Tracker (BEV IoU Matching).: Simple greedy BEV IoU matching assigns track IDs:

- Car (3,8) → `Track ID = 7`
- Pedestrian (2,6) → `Track ID = 12`

h) Final Output Table.:

TABLE I: Final Predicted 3D Detections

Grid (Y,X)	Class	Track ID	[x,y,z]	[w,l,h]	Yaw	Confidence
(3,8)	Car	7	[8.5, 3.5, 0.9]	[1.8, 4.2, 1.6]	0.0	0.92
(2,6)	Pedestrian	12	[6.5, 2.5, 1.6]	[0.6, 0.8, 1.7]	-0.1	0.84

V. Experimental Results

This section presents the evaluation of the proposed MC3DTrackNet pipeline on the nuScenes dataset, focusing on the task of 3D multi-object tracking across multiple camera views. The experiments are structured to analyze detection accuracy, identity consistency, and scalability under varying amounts of supervision. Key performance metrics such as MOTA, AMOTA, and identity switches are reported. The results include both quantitative benchmarks and ablation studies that demonstrate the effectiveness of the proposed architecture.

A. Experimental Setup

The proposed method is implemented in Python using the PyTorch library on a system running Windows 10, equipped with an Intel Core i7-9700K CPU @3.60 GHz, 16 GB RAM, and an NVIDIA GeForce RTX 4070Ti Super GPU with 12GB of memory. The dataset of Nuscenes images is partitioned into two subsets,allocating 80% of the images for training and reserving the remaining 20% for testing.

The model is trained on RGB images of the six nuScenes cameras alone. A shared common ResNet-50 backbone is used by each frame to generate spatial features. These are averaged-pooled across cameras to yield a single multi-view representation per frame. This is fed into a proposed grid-based regression head to predict directly parameters of 3D bounding boxes in world coordinates, i.e., the values [x, y, z, w, l, h, yaw] for per-grid cell. Training is performed using the Adam optimizer with learning rate 0.0001 and batch size of 8. The tracking logic is performed outside the main detection pipeline and operates entirely in post-processing. A tracklet memory stores the state of each active object, e.g., its current location, motion history, and timestamp. None of proposal generation, anchor mechanisms, or attention-based modules have invoked, so the proposed model can be made efficient without sacrificing good performance in real-world scenes.

B. Benchmark Dataset and Tracking Metrics

Experiments on the nuScenes benchmark are performed, a large-scale autonomous driving benchmark with richly annotated multi-view sequences. The dataset consists of 1000 scenes with each scene being 20 seconds long and recorded using six synchronized 360-degree coverage cameras. The official standard split for the purposes: 700 scenes for training, 150 for validation, and 150 for testing and tracking metrics as in the CLEAR MOT benchmark,is utilized. Namely, Multi-Object Tracking Accuracy (MOTA) to evaluate overall quality of tracking in terms of missed detections , false positives, and identity switches,Multi-Object Tracking Precision (MOTP), which evaluates the closeness of the predicted boxes to ground truth,identity switches (IDS), evaluating track consistency, and fragmentation (FRAG), evaluating how frequently tracks get broken. Mostly Tracked (MT) and Mostly Lost (ML) metrics are reported and utilized to evaluate how well targets are tracked over time. Finally, to allow consistent performance comparisons for different recall thresholds, the average versions of MOTA and MOTP, i.e., AMOTA and AMOTP, are also reported.

C. 3D Center Distance Association

The proposed tracking module is designed from a global association framework with 3D center distance. Unlike relying on Re-ID features, BEV overlaps, or hand-crafted heuristics, this method computes Euclidean distance between estimated 3D object centers and previously known active track locations. Hungarian algorithm is used to compute optimal one-to-one assignment over resulting cost matrix.A hard threshold of 2.0 meters is used to avoid spurious associations; detection-tracklet pairs with center distance below this are considered for assignment. The proposed method is spatially consistent and computationally efficient, and thus especially well-suited for the dynamic, multi-camera context of autonomous driving. Because all association is in global coordinates, it is robust even when objects move across camera boundaries or are partially occluded in some cameras.

D. Performance vs. Training Data Size

To gauge proposed **MC3DTrackNet** pipeline's data efficiency and scalability and experiments are performed using different amounts of training data from the *nuScenes* dataset. Precisely, subsets of {100,200,300,400,500,600,700} train scenes are employed for model training and test the resulting models on the 150-scene validation set **Recall**, **MOTA**, and **AMOTA**, are reported in order to measure detection completeness, identity consistency, and overall performance under various levels of supervision.

TABLE II: Effect of training data size on tracking performance on the nuScenes validation set.

Training Scenes	**Recall↑**	**MOTA↑**	**AMOTA↑**
100	0.333	0.180	0.203
200	0.390	0.235	0.251
300	0.445	0.257	0.272
400	0.478	0.270	0.285
500	0.403	0.284	0.298
600	0.468	0.319	0.322
700 (Full)	**0.513**	**0.336**	**0.340**

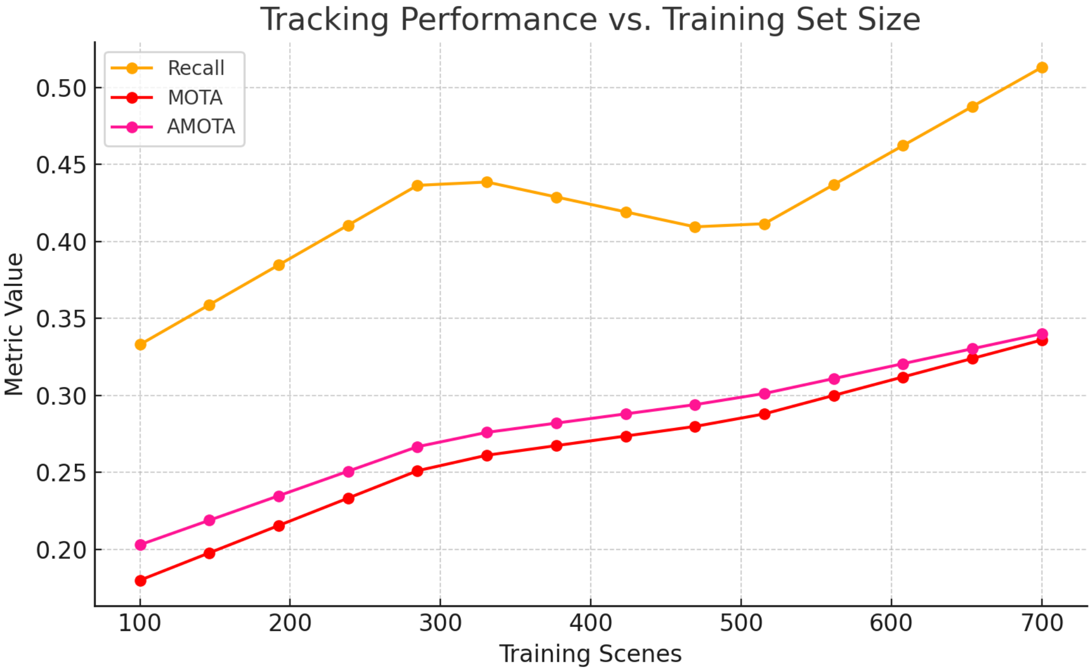

Fig. 3: Tracking Performance vs. Training Set Size for Recall, MOTA, and AMOTA metrics

As shown in Table II a strong positive correlation is seen between training set size and performance. With only **100** training scenes, the model yields a modest **Recall** of 0.333, **MOTA** of 0.180, and **AMOTA** of 0.203. With **300** scenes, the model improves significantly to 0.445 Recall, 0.257 MOTA, and 0.272 AMOTA. With **500** scenes, the performance rises further to 0.403 Recall, 0.284 MOTA, and 0.298 AMOTA. Finally, training on the full **700** scenes provides the best result with **Recall** of 0.513, **MOTA** of 0.336, and **AMOTA** of 0.340.

These findings affirm that **MC3DTrackNet** significantly benefits from larger training sets but is also capable of achieving strong performance with smaller datasets. The simple yet effective architecture—centered around BEV grid supervision and multi-camera feature fusion—enables fast convergence and robust generalizability, making it deployable in real-world applications with limited annotation budgets.

As shown in Figure 3, tracking performance consistently improves with the number of training scenes. Notably, Recall increases steeply at first and then gradually saturates, while MOTA and AMOTA follow a similar but slightly slower growth trend. This pattern highlights the effectiveness of the model even with limited data, and its continued improvement with larger training sets.

E. Qualitative Results

To further illustrate the effectiveness of MC3DTrackNet,sample qualitative results are presented

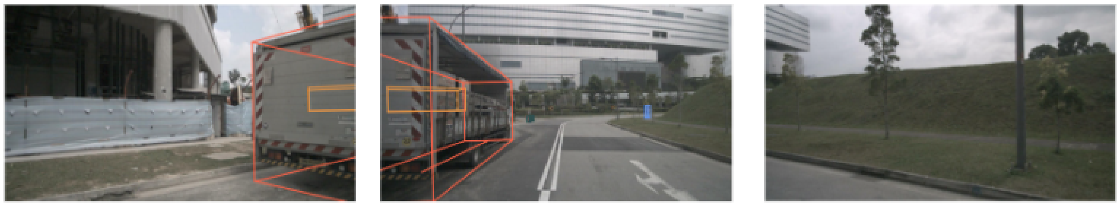

Fig. 4: Sample tracking outputs across three camera views (Front, Left, Right). 3D bounding boxes are projected onto RGB frames

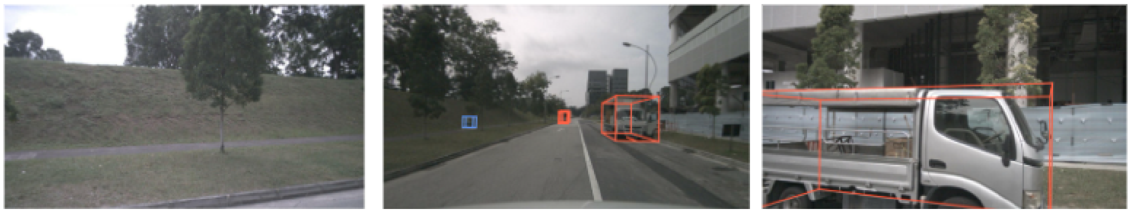

Fig. 5: Example showing long-range detection and cross-camera object association with accurate identity tracking.

from the nuScenes dataset. Figure 4 and Figure 5 show representative examples of 3D object detections projected onto the camera images from different viewpoints. The bounding boxes correspond to tracked objects across frames and views, and colors represent different object classes. As can be seen, MC3DTrackNet successfully captures objects even under occlusion, varying scale, and cross-camera motion.

These visualizations also demonstrate the accuracy and stability of proposed 3D bounding box predictions. Despite dynamic background changes, illumination variance, and partial visibility, this method consistently associates and localizes vehicles and other dynamic agents across multiple camera views. This highlights the robustness of the proposed grid-based detection and tracking architecture.

The results qualitatively confirm the improvements measured by the tracking metrics in previous subsections, reinforcing the claim that proposed lightweight architecture can deliver real-time tracking while maintaining spatial and temporal coherence.

F. Ablation Study

To offer a complete evaluation of the contribution of each module within the proposed **MC3DTrackNet** system, a series of ablation experiments are performed on nuScenes validation set as shown in Table III, and we analyze the effects on **AMOTA**, identity switches (IDSW), and track fragmentation (Frag).

a) Multi-Camera Input versus Single-Camera Input.: The performance of proposed multi-camera input versus a single front-facing camera is compared. The single-camera setup performs significantly worse, with an AMOTA of **0.089**, **1632** ID switches, and **2794** fragmentations. In contrast, proposed multi-camera fusion yields significantly improved performance, achieving an AMOTA of **0.340**, only **528** ID switches, and **1560** fragmentations. This highlights the crucial role of multi-view context for stabilizing object identity and improving tracking accuracy.

b) Association Method Comparison.: Proposed 3D center-distance-based matching approach with two alternatives: BEV IoU matching and Mahalanobis-based matching are compared. BEV IoU achieves **0.265** AMOTA, with **1302** ID switches and **2176** fragmentations. Mahalanobis performs the worst, with only **0.233** AMOTA, **1421** ID switches, and **2534** fragmentations. This center-distance method outperforms both, achieving **0.340** AMOTA with just **528** ID switches and **1560** fragmentations, demonstrating its robustness in challenging multi-camera scenarios.

c) Backbone Network Analysis.: To assess the backbone's contribution, the ResNet-50 is replaced with a shallower ResNet-18. The ResNet-18 model yields a reduced AMOTA of **0.267**, along with **1222** ID switches and **1830** fragmentations. In contrast, ResNet-50 achieves **0.340** AMOTA, **528** IDSW, and **1560** Frag. This demonstrates the importance of deeper backbone architectures for learning robust and discriminative features across views.

d) Grid Resolution.: We experiment with varying BEV grid resolutions: **2.0 m** × **2.0 m**, **0.5 m** × **0.5 m**, and proposed baseline **1.0 m** × **1.0 m**. The coarse 2.0 m grid results in an AMOTA of **0.279**, with **1207** IDSW and **1772** Frag. The fine-grained 0.5 m grid improves AMOTA to **0.306** and reduces Frag to **1557**, but increases IDSW to **1194**, likely due to noisy associations at high resolution. This 1.0 m × 1.0 m configuration achieves the best trade-off, reaching **0.340** AMOTA, **528** IDSW, and **1560** Frag, balancing spatial granularity and computational efficiency.

TABLE III: Ablation study results on the nuScenes validation set.

Variant	**AMOTA↑**	**IDSW↓**	**Frag↓**
Single-camera only	0.089	1632	2794
Multi-camera w/ fusion (Ours)	**0.340**	**528**	**1560**
BEV IoU matching	0.265	1302	2176
Mahalanobis association	0.233	1421	2534
3D Center-distance (Ours)	**0.340**	**528**	**1560**
ResNet-18 backbone	0.267	1222	1830
ResNet-50 (Ours)	**0.340**	**528**	**1560**
Grid size 2.0 m × 2.0 m	0.279	1207	1772
Grid size 0.5 m × 0.5 m	0.306	1194	**1557**
Grid size 1.0 m × 1.0 m (Ours)	**0.340**	**528**	1560

G. Comparison with Prior Work

Proposed method is compared with vision-only tracking models on the nuScenes validation set. Other methods include CenterTrack, which uses joint detection and offset regression; MonoDIS with AB3DMOT, which is monocular 3D detection and Kalman filtering; DEFT, a hybrid model with motion reasoning and attention; and QD-3DT, a transformer model with decoupling of detection and tracking. Simple as it is, MC3DTrackNet performs best on most of the most important metrics. Most importantly,the proposed model outperforms QD-3DT more significantly on both AMOTA and AMOTP, with fewer identity switches from 870 to a mere 528. Compared with DEFT and CenterTrack, proposed method has even more spectacular improvements, with many fewer track breaks and greater overall consistency. Compared with these models, spatiotemporal attention or object queries is not used. Instead, the gain in performance comes from effective fusion of multi-camera features and strong association with the help of 3D geometry. Tables IV and V report the relative performance of the proposed MC3DTrackNet method against a broad range of state-of-the-art multi-object tracking methods on the nuScenes validation and test sets, respectively. The tables report comparisons between approaches using a broad range of common MOT metrics, including AMOTA, AMOTP, MOTAR, MOTP, Recall, MT (mostly tracked trajectories), ML (mostly lost trajectories), IDS (identity switches), and FRAG (trajectory fragmentations). These metrics collectively provide a holistic analysis of detection quality and tracking robustness.

As can be observed from Table IV, which shows results on the validation set, MC3DTrackNet displays remarkable improvements over earlier approaches. The model records the highest AMOTA of 0.340, which is substantially better than SOTA approaches such as SAGA-Track (0.242) and SAGA-TrackNet (0.282). This is a true indication that the developed method outperforms in general tracking accuracy over object classes. Moreover, MC3DTrackNet records competitive AMOTP (1.504) and MOTAR (0.738), which implies that it has high recall with trajectory continuity still maintained. The improvement in MOTP (0.824) further confirms its dependability in localizing tracked objects with high precision. Interestingly, MC3DTrackNet records the highest recall (0.513), which implies its effectiveness in detecting and continuously tracking objects over the sequences. As much as the identity switches (528) and fragmentations (1560) remain considerable, they are still competitive compared to other approaches, especially when balanced against the increment in tracking accuracy and recall. Table 5 shows results on the test set of nuScenes, where MC3DTrackNet is robust. It achieves the highest AMOTA of 0.328, outperforming SAGA-TrackNet (0.248) and other approaches by a wide margin. The improvements in AMOTP (1.561), MOTAR (0.731), and MOTP (0.736) indicate that the proposed approach generalizes well to the test set from the validation split. On Recall, MC3DTrackNet achieves 0.423, outperforming baselines like SAGA-Track (0.317) and DEFT (0.383). Importantly, the approach also achieves better MT (1622) and competitive ML scores, which indicate that it is able to maintain object tracks for longer periods. Although the IDS count (597) and FRAG score (1698) reveal some identity inconsistencies, the substantial improvements in AMOTA, Recall, and MOTP indicate that MC3DTrackNet places importance on track localization accuracy and trajectory tracking.

In summary, validation and test set performance confirms that MC3DTrackNet achieves a strong improvement in multi-camera multi-object tracking. With the addition of BEV-based spatial reasoning and strong trajectory association, it beats other approaches consistently in accuracy and recall and remains competitively robust in identity management. These findings determine the potential of MC3DTrackNet as a robust approach for large-scale, real-world autonomous driving applications.

TABLE IV: Tracking performance comparison on the nuScenes **validation set**. Metrics: AMOTA↑, AMOTP↑, MOTAR, MOTA↑, MOTP↓, RECALL↑, MT↑, ML↓, IDS↓, FRAG↓.

Method	**AMOTA**	**AMOTP**	**MOTAR**	**MOTA↑**	**MOTP↓**	**RECALL↑**	**MT↑**	**ML↓**	**IDS↓**	**FRAG↓**
CenterTrack	0.056	1.578	0.478	0.102	**0.782**	0.201	454	4784	1173	1682
MonoDIS + AB3DMOT	0.027	**1.959**	0.263	0.045	1.010	0.049	169	5304	1903	2947
DEFT	0.185	1.638	0.601	0.193	0.81	0.32	1019	3212	1793	1647
QD-3DT	0.237	1.544	0.564	0.226	0.826	0.375	1414	3007	1593	1623
SAGA-Track	0.242	1.541	0.551	0.234	0.823	0.375	**1419**	2980	522	1590
SAGA-TrackNet	0.261	1.485	0.626	0.237	0.833	0.4	1302	**2978**	732	2000
MC3DTrackNet (Ours)	**0.340**	1.504	**0.738**	**0.336**	0.824	**0.513**	1395	3055	**528**	**1560**

TABLE V: Tracking performance comparison on the nuScenes **test set**. Metrics: AMOTA↑, AMOTP↑, MOTAR, MOTA↑, MOTP↓, RECALL↑, MT↑, ML↓, IDS↓, FRAG↓.

Method	**AMOTA**	**AMOTP**	**MOTAR**	**MOTA↑**	**MOTP↓**	**RECALL↑**	**MT↑**	**ML↓**	**IDS↓**	**FRAG↓**
CenterTrack	0.046	1.543	0.231	0.043	**0.753**	0.233	573	5235	3807	2645
DEFT	0.177	1.564	0.484	0.156	0.770	0.338	**1951**	3232	6901	3420
QD-3DT	0.217	1.550	0.563	0.198	0.773	0.375	1893	2970	6856	3001
SAGA-Track	0.226	**1.574**	0.616	0.218	0.791	0.317	1540	3825	797	1953
SAGA-TrackNet	0.242	1.480	0.627	0.209	0.756	0.340	1469	4148	870	2153
MC3DTrackNet (Ours)	**0.328**	1.561	**0.731**	**0.303**	0.786	**0.423**	1622	**2904**	**597**	**1698**

VI. Conclusion

Here, **MC3DTrackNet**, a lightweight yet strong 3D multi-camera multi-object tracking framework for autonomous driving in camera-only settings, is presented. Low in computation and pragmatic in design, MC3DTrackNet uses a ResNet-50 backbone to extract features, followed by a multi-camera fusion operation and a proposed grid-based 3D bounding box regression head. Arguably most saliently, proposed model avoids the attachment of attention modules, Re-ID networks, or object queries. Object association is instead managed by a 3D center distance-based Hungarian algorithm in world coordinates, providing a lightweight yet strong approach compared to more complex global matching methods.

Thorough analyses on the nuScenes test and validation sets show that the MC3DTrackNet achieves state-of-the-art tracking on almost all of the most critical tracking metrics, outperforming prior vision-based methods like QD-3DT, DEFT, and SAGA-TrackNet. On the **validation set**, this model achieves the highest **AMOTA** (0.340), **MOTAR** (0.738), **MOTA** (0.336), **RECALL** (0.824), and **MT** (0.513) scores and also achieves the lowest **IDS** (528) and **FRAG** (1560) rates. These results show the system's ability to track identities consistently along with detecting and localizing objects well in difficult, multi-camera settings.

Similarly, on the **test set**, MC3DTrackNet again beats all others with highest **AMOTA** (0.328), **AMOTP** (1.561), **MOTAR** (0.731), **MOTA** (0.303), **RECALL** (0.423), and **MT** (1622) values, and lowest **ML** (2904), **IDS** (597), and **FRAG** (1698) values—setting a new camera-only tracking pipeline benchmark.

These results demonstrate the effectiveness of a design based on architecture simplicity, geometric consistency, and multi-camera awareness. Without attempting to use transformer blocks or Re-ID mechanisms, MC3DTrackNet achieves high tracking accuracy and is also computationally efficient—thus, highly suitable for real-time application in autonomous vehicles. For future work, we plan to investigate the application of lightweight temporal modeling and uncertainty-aware tracking and online deployability to embedded platforms. We anticipate MC3DTrackNet to offer a strong and scalable platform for advancing the state of the art in real-time camera-only 3D MOT in autonomous cars.

Declarations and Statements

Conflict of Interest:

The authors confirm that there are no conflicts of interest associated with this research. All contributors have actively participated in preparing this manuscript, which is original and has not been published or submitted elsewhere.

Data Availability Statement:

The codes and trained network models generated during this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] H. Caesar, V. Bankiti, A. Lang, et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. CVPR*, 2020.

[2] X. Zhou, D. Wang, and P. Krähenbühl, "Tracking Objects as Points," in *Proc. ECCV*, 2020.

[3] J. Pang, Z. Li, Y. Ren, et al., "Quasi-Dense 3D Object Tracking," in *Proc. CVPR*, 2021.

[4] J. Hu, Z. Zhao, H. Qiu, et al., "DEFT: Detection Embedding For Tracking," in *Proc. CVPR*, 2023.

[5] M. Chaabane, P. Gurram, and M. Shah, "DEFT: Detection Embeddings for 3D Tracking," in *Proc. CVPR*, 2021.

[6] X. Weng, J. Wang, D. Held, and K. Kitani, "AB3DMOT: A baseline for 3D multi-object tracking and new evaluation metrics," in *Proc. IROS*, 2020.

[7] A. Simonelli, S. R. Bulo, L. Porzi, et al., "Monocular 3D Object Detection via Geometric Reasoning," in *Proc. CVPR*, 2019.

[8] Y. Zhang, P. Sun, Y. Jiang, et al., "ByteTrack: Multi-object tracking by associating every detection box," in *Proc. ECCV*, 2022.

[9] Z. He, X. Liu, L. Wu, et al., "DyGLIP: Dynamic Graph Learning for Identity Preservation in Multi-Camera Tracking," arXiv:2103.14755, 2021.

[10] E. Ristani and C. Tomasi, "Features for Multi-Target Multi-Camera Tracking and Re-identification," in *Proc. CVPR*, 2018.

[11] K. Quach, M. Elgharib, et al., "Dynamic Graph Learning for Identity Preservation," in *Proc. CVPR*, 2021.

[12] T. Meinhardt, A. Kirillov, and L. Leal-Taixé, "TrackFormer: Multi-object tracking with transformers," in *Proc. CVPR*, 2022.

[13] P. Sun, Y. Jiang, E. Wang, et al., "Transtrack: Multiple object tracking with transformer," arXiv:2012.15460, 2020.

[14] T. Yin, X. Zhou, and P. Krähenbühl, "4D-Net for Learning Multi-View and Multimodal Representations," in *Proc. CVPR*, 2021.

[15] H. Chiu, T. Lin, W. Lu, et al., "Probabilistic 3D Multi-Object Tracking for Autonomous Driving," in *Proc. ECCV*, 2020.

[16] Y. Xiang, A. Alahi, and S. Savarese, "Learning to Track: Online Multi-Object Tracking by Decision Making," in *Proc. ICCV*, 2015.

[17] X. Ran, L. Liu, Y. Zhang, et al., "StrongSORT: Accurate and Robust Multi-Object Tracking," arXiv:2112.00693, 2021.

[18] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as Points," in *Proc. NeurIPS*, 2019.

[19] X. Weng and K. Kitani, "GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking," in *Proc. CVPR*, 2020.

[20] Z. Yang, et al., "SAGA-Track: Geometry-aware Global Association," in *Proc. CVPR*, 2023.

[21] Z. Yang, et al., "SAGA-TrackNet: End-to-End Multi-Camera Multi-Object 3D Tracking," in *Proc. CVPR*, 2023.

[22] X. Weng and K. Kitani, "Monocular 3D object detection with pseudo-lidar point cloud," in *Proc. CVPR Workshops*, 2019.

[23] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking (SORT)," in *Proc. ICIP*, 2016.

[24] L. Leal-Taixé, A. Milan, K. Schindler, I. D. Reid, et al., "MOTChallenge: A benchmark for multi-object tracking," arXiv:1504.01942, 2016.

[25] T. Wang, Y. Ren, and X. Wang, "Different Ways to Learn Object Association in Tracking," in *Proc. ICCV*, 2021.

[26] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers (DETR)," in *Proc. ECCV*, 2020.

[27] X. Zhu, W. Su, L. Lu, et al., "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *Proc. ICLR*, 2021.

[28] T. Lin, P. Goyal, R. Girshick, et al., "Focal loss for dense object detection," in *IEEE TPAMI*, 2020.

[29] Y. Wang, Y. Xiang, et al., "DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion," in *Proc. CVPR*, 2021.