MotionTrack: End-to-End Transformer-based Multi-Object Tracking with LiDAR-Camera Fusion

Ce Zhang[*]
Virginia Tech
zce@vt.edu

Chengjie Zhang
Motional
chengjie.zhang@motional.com

Yiluan Guo
Motional
yiluan.guo@motional.com

Lingji Chen
Motional
lingji.chen@motional.com

Michael Happold
Motional
michael.happold@motional.com

Abstract

Multiple Object Tracking (MOT) is crucial to autonomous vehicle perception. End-to-end transformer-based algorithms, which detect and track objects simultaneously, show great potential for the MOT task. However, most existing methods focus on image-based tracking with a single object category. In this paper, we propose an end-to-end transformer-based MOT algorithm (MotionTrack) with multi-modality sensor inputs to track objects with multiple classes. Our objective is to establish a transformer baseline for the MOT in an autonomous driving environment. The proposed algorithm consists of a transformer-based data association (DA) module and a transformer-based query enhancement module to achieve MOT and Multiple Object Detection (MOD) simultaneously. The MotionTrack and its variations achieve better results (AMOTA score at 0.55) on the nuScenes dataset compared with other classical baseline models, such as the AB3DMOT, the CenterTrack, and the probabilistic 3D Kalman filter. In addition, we prove that a modified attention mechanism can be utilized for DA to accomplish the MOT, and aggregate history features to enhance the MOD performance.

1. INTRODUCTION

Perception is a fundamental and key element for autonomous vehicles. Common perception tasks fall into three categories [8]: Multiple Object Detection (MOD), Multiple Object Tracking (MOT), and Multiple Object Prediction (MOP). A reliable MOT algorithm shall comprehend the MOD outcomes and establish a connection for the MOP.

Machine learning-based (ML-based) tracking algorithms recently become popular to improve MOT performance by enhancing the temporal and spatial features through learning [9, 19, 22]. Current ML-based tracking algorithms have two paradigms: tracking by detection, and simultaneous tracking and detection [2]. The former considers detection and tracking as separate and sequential tasks, while the latter jointly processes detection and tracking at the same time. Both paradigms utilize neural networks for motion prediction (MP) or data association (DA). Simultaneous tracking and detection offer a significant advantage through mutual feature sharing. Specifically, temporal and spatial features from tracking can improve detection performance, whereas appearance and position features from detection can enhance DA in tracking. Because of these benefits, we choose the simultaneous tracking and detection paradigm for MotionTrack.

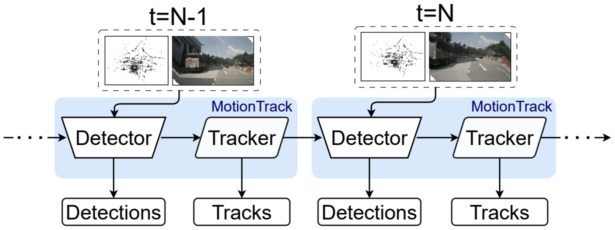


Figure 1. MotionTrack Model Demonstration

The transformer architecture and the attention mechanism, originally applied in the field of natural language processing [34], perform well for vision tasks such as object detection, image quality assessment, and pose estimation [15, 30, 31, 41, 43–45, 48]. Recent studies indicate that the transformer model can be utilized for tracking tasks by estimating object motion and transferring appearance and motion features [6, 21, 26, 32, 42]. In view of the nature of the transformer's self-attention and cross-attention mechanism, the dot product process between the query, key, and

[*] Work conducted during an internship at Motional

value matrices can be likened to a DA process. Thus, we hypothesize that the transformer architecture has the potential to be applied beyond MP and feature transferring for MOT task, which can be adapted for DA.

Additionally, current tracking-related transformer algorithms apply to the case with a single sensor modality input (usually images), in a stationary position, with a high sampling frequency (usually 30 Hz), and for a single object category (the human class) [7]. But for autonomous driving, tracking algorithms can operate with multi-modality sensor inputs (e.g., images and LiDAR-based point clouds), on a moving ego vehicle, with relatively low sampling frequency (10 Hz) and for multiple object categories (classes of pedestrian, car, truck, etc.). To the authors' best knowledge, no existing transformer tracking algorithms handle such intricate situations effectively.

Based on the above-mentioned assumptions and issues, we raise three questions: (1) Can a transformer-based DA algorithm be applied for simultaneous MOD and MOT under an autonomous driving environment? If so, can a DA algorithm suffice without explicit MP and state estimation processes? (2) How to handle the multiple sensor inputs for DA? (3) Is it possible to enhance the detection performance through history-endowed tracking features? To answer these questions, we propose a novel end-to-end transformer-based algorithm (MotionTrack) for simultaneous MOD and MOT with LiDAR and image inputs (Figure 1). MotionTrack algorithm utilizes a modified transformer to achieve DA and another transformer to update potential object features from the tracking information. The proposed algorithm is tested and evaluated through the nuScenes dataset, which achieves 2-3x higher AMOTA results than the other baseline algorithms, and is on par with popular tracking solutions, such as the probabilistic 3D Kalman filter. The contributions of this paper include:

- Designing a transformer-based module for DA with multi-modality sensor inputs to achieve tracking without MP and state estimation.
- Developing a query enhancement module (QEM) to improve detection performance by combining the history tracking features.
- Establishing a baseline for an end-to-end transformer MOD and MOT algorithm in the autonomous driving environment.

To the best of our knowledge, this is the first end-to-end transformer algorithm for simultaneous MOD and MOT with multi-modality sensor inputs in an autonomous driving environment. We emphasize that the objective of this paper is to investigate the feasibility and establish a baseline rather than to achieve state-of-the-art (SOTA) results for tracking tasks with the nuScenes dataset; which would require further improvements on top of our baseline.

2. Related Work

Although objects move in a three-dimensional physical space, MOT can be performed to track objects in 3D or in 2D such as in an image [25]. Inputs to MOT can be 2D such as images, or 3D such as Lidar point clouds, or both. MOT can employ traditional methods for MP, filtering, and DA, and it can employ neural networks to achieve desired goals for tracking.

2.1. 2D MOT

2D MOT usually uses images and object states as input [3, 28, 29, 33, 37, 47]. Most 2D algorithms leverage the rich semantic information available in images and dense temporal features to accomplish MOT. However, image targets do not offer explicit position and motion information, strongly affects the 2D MOT performance. One of the most popular traditional 2D MOT methods is the Simply Online and Realtime Tracking (SORT) [4], which employs a Kalman filter for MP and the Hungarian algorithm for DA across frames. The successor to SORT, namely DeepSORT [39], modifies the DA algorithm to further improve tracking performance by utilizing a Mahalanobis distance assigner and a neural network-based appearance feature descriptor to assist DA between frames. Though traditional MOT algorithms are reliable and easy to deploy, it requires massive parameter tuning. Moreover, traditional MOT cannot deal with edge cases such as object occlusion.

ML-based MOT algorithms are developed to solve the aforementioned issues in traditional tracking. CenterTrack [49] designs a convolutional neural network-based (CNN) MP module to estimate the heatmap displacements between 2 frames for DA. It achieves good performance on the MOT datasets while ID-switch and long-term tracking issues are yet to be solved.

Besides CNN, transformer architecture becomes popular recently [21, 32, 40, 42, 50, 52], mainly thanks to its capability of global feature extraction and temporal feature aggregation. The global feature extraction processes all potential object queries simultaneously, while the temporal feature aggregation transfers past object features to the current frame as prior knowledge. Current popular transformer MOT algorithms are TrackFormer [21], MOTR [42], global tracking transformer (GTR) [50], and MeMOT [6]. The TrackFormer and MOTR follow the simultaneous detection and tracking paradigm by concatenating the detected objects' embedding with the proposed newborn query embedding. The GTR follows the tracking-by-detection paradigm, utilizing the self- and cross-attention mechanisms to associate objects among all input frames. The MeMOT comprises a DETR-based detector and three attention mechanisms to aggregate the object features from previous detections, before using an association solver for MOT. All these transformer-based algorithms demonstrate

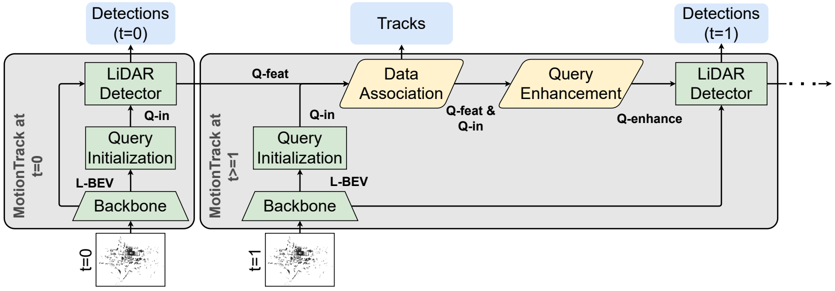


Figure 2. MotionTrack Architecture with LiDAR Input. The green blocks represent the detection modules, the yellow blocks represent the tracking modules, and the blue blocks are the detection and tracking results.

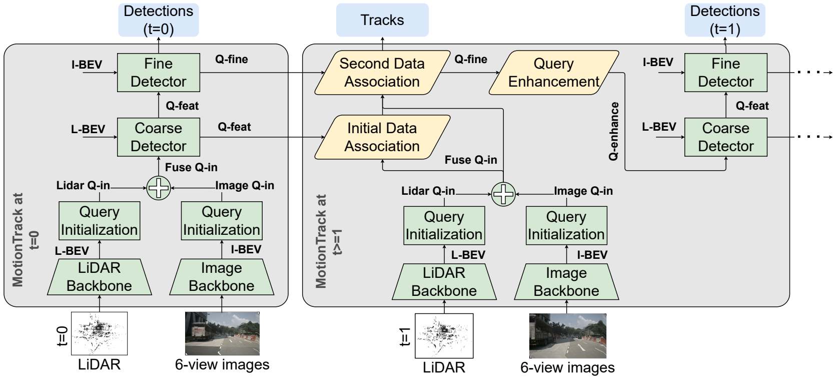


Figure 3. MotionTrack Architecture with LiDAR-Image Input. The overall architecture is similar to the Figure 2, while there is an extra object association module for the fine detector results.

effective 2D MOT. However, these algorithms are designed for image-only tracking applications with a single class.

2.2. 3D MOT

3D MOT algorithms, commonly applied for autonomous driving, take images, LiDAR sensor's point cloud, or LiDAR-image fusion data as inputs.

Image-based 3D MOT algorithms leverage dense appearance features for DA. AB3DMOT [38] employs a 3D Kalman filter and a Hungarian algorithm for tracking. [27] utilizes the Poisson multi-Bernoulli mixture tracking filter to achieve MOT with a single camera input. Besides traditional tracking methods, [23] and [46] introduce transformer-based 3D MOT models. Both use the transformer model in a manner of TrackFormer and MOTR, by concatenating tracked objects to the current frame for MOD and MOT. [23] further aggregates temporal features to enhance tracking and detection. Despite dense appearance features, image-based 3D MOT algorithms' performance cannot compete with LiDAR-based methods due to a lack of explicit position and distance features.

Most LiDAR-based algorithms focus on modeling the tracked objects' motion features to achieve MOT. SimTrack [20], and SimpleTrack [24] project the 3D features to BEV for feature extraction, where SimTrack utilizes neural networks to predict the object motion to achieve tracking, and SimpleTrack is focused on improving the association and motion model performance. [12], inspired from the R-CNN detector [10], develops a track-align module to aggregates the track region-of-interests for MOT.

Currently, researchers are interested in LiDAR-image fusion as the image feature offers intensive appearance features, and the point cloud provides accurate distance and position features. EagerMOT [16] is a traditional 3D tracker that employs a two-stage association algorithm where the first stage is a standard Hungarian data association for 3D bounding boxes, and the second stage is an identical Hungarian assigner with 2D bounding boxes. JMODT [14] is a joint detection and tracking algorithm that uses point clouds and images as inputs with a novel neural network-based object correlation and object association. CAMO-MOT [35] combines motion and appearances features to prevent false detection. In the meantime, they design a tracking cost matrix to prevent tracking occlusion, which achieves the current SOTA algorithm at nuScenes dataset. Even though numerous literatures, to the best of the authors' knowledge, there are no multi-modality, end-to-end transformer algorithms that exist for 3D MOT autonomous driving.

Table 1. MotionTrack terminology and acronym names.

Terminology	Explanation	Acronym
Coarse (LiDAR) Detector	A transformer decoder layer to extract features from the input queries for object detection. It is named as LiDAR detector in the LiDAR-only setup and coarse detector in the LiDAR-image setup.	C-Det
Fine Detector	A transformer decoder layer to extract features from the input queries for object detection with LiDAR-image setup	F-Det
Query Input	Queries generated based on heatmap results and use as the input to the coarse or LiDAR detector at the initial frame or the input to the QEM at the following frames	Q-in
Query Features	The output from the coarse or LiDAR detector, also used as the input to the fine detector and the DA module	Q-feat
Query Features Fine	The output from the fine detector, also use as the input to the extra DA module	Q-fine
Enhanced Query	The enhanced queries generated from the QEM, used as the input to coarse or LiDAR detector	Q-enhance
LiDAR BEV	The LiDAR BEV features	L-BEV
Image BEV	The image BEV features	I-BEV
query cross-attention	A cross attention layer to update the query features from previous frame	Q-cross
head cross-attention	A cross attention layer to update the heading angle features to query features	H-cross

3. PROPOSED METHODOLOGY

MotionTrack is a simultaneous detection and tracking algorithm. For MOD, we employ the TransFusion model. For MOT, we design a transformer-based DA module. Furthermore, we develop a QEM to inherit the history temporal information for better detection performance. The MotionTrack algorithm comprises two setups with different sensor configurations: LiDAR-only input (Figure 2) and LiDAR-image fusion inputs (Figure 3). Both setups contain an object detection module, a DA module, and a QEM. Due to the intricacy of the MOT problem, we define several terms for the MotionTrack (Table 1)

3.1. MotionTrack Detector Module

MotionTrack's detector is TransFusion [1], which flexibly supports LiDAR-only and LiDAR-image setups.

The LiDAR-only setup supports two implementations: PointPillar [18] as a cost-efficient model; or VoxelNet [51] for high performance. The extracted features are transformed into BEV features (L-BEV) for object query initialization. Then, we generate heatmaps based on the L-BEV to determine the initial location of the Q-in. After the heatmap generation, an one-layer transformer decoder takes the L-BEV and Q-in as the inputs to extract Q-feat. Finally, the Q-feat is fed into prediction head layers for object detection.

Compared to the LiDAR-only setup, the LiDAR-image one shares the same LiDAR backbone, outputting L-BEV and LiDAR's Q-in. In the image branch, ResNet-50 [13] backbone first extracts features from 6-view RGB images, and projects them to BEV (I-BEV). Then, we generate the image heatmap, before fusing the image's and LiDAR's Q-in together. The detection in the LiDAR-image setup has two transformer decoder layers: one is C-Det, which consumes L-BEV and Q-in; the other one is F-Det, which uses I-BEV and the C-Det's output (Q-feat) as the inputs. Finally, the F-Det's output is fed into the prediction head layers. Details about the detector module can be found in [1].

3.2. Transformer-based Association Module

Common MOT algorithms perform motion prediction (MP) and data association (DA) in sequence. The core of MOT is DA, as it enables the connection of objects between frames, while the MP serves to support the DA process. The outcome of a DA algorithm is whether current frame objects are tracked or new-born objects, and whether the previous frame's objects are disappeared, namely dead objects [11]. Here we investigate a DA design through the transformer architecture without explicit MP.

Transformer DA Inspiration: The self- and the cross-attention mechanism are the core of transformers. For both mechanisms, the attention function is $softmax(Q*K^T)*V$ where Q, K, and V are known as the query, key, and value matrices learned from the inputs. The Q, K, and V matrices are learned from the same inputs for the self-attention (input-A), while the Q are learned from a different input with the K and V matrices for the cross-attention [34]. The essence of the attention function is to update the value matrix based on a cost matrix ($softmax(Q * K^T)$) obtained from query and value matrices.

The cost matrix inspires us that such an attention mechanism can be used to calculate the affinity between the tracks and the observations. However, the experiment results indicate that directly applying the attention mechanism is not ideal because the softmax activation function causes the gradient vanishing issue during training. Therefore, instead of the softmax activation function, we directly compute the cost matrix with a dot product between the observation and tracks features, which shows excellent association performance. Furthermore, we find that the attention mechanism performs well on updating object features from the previous frame to the current frame because the $softmax(\frac{Q*K^T}{C})$ can easily learn to filter redundant features and preserve necessary features from the previous frame. Based on these findings, MotionTrack's association module employs transformer architecture to update the previous frame's objects features and uses a dot product computation for DA.

The MotionTrack comprises two DA module configurations for LiDAR-only and LiDAR-image inputs. Each DA module outputs independent tracklets estimations. The differences are that the LiDAR-image inputs comprise an extra DA module.

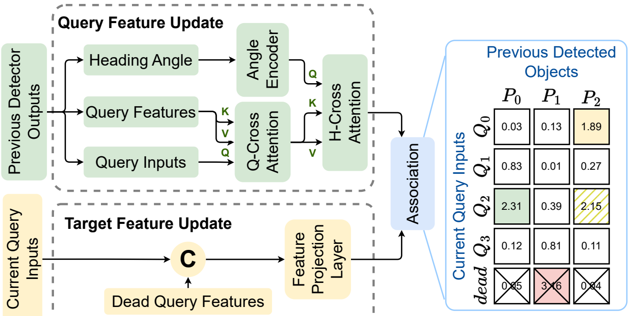
DA Module: The DA module contains three steps: query feature update, target feature update, and query-target feature association (Figure 4).

The query feature update process is aimed at establishing and enhancing detected object features from previous frames. It takes the previous frame's detected Q-feat, Q-in, and the objects' heading angles as the input, passing through two cross-attention layers (H-cross & Q-cross) to update the detected objects' features. The objective of the Q-cross is to update the appearance feature for detected objects. Appearance features are important for DA since the association is determined by the similarity between the appearance features. The H-cross is designed to inherit the objects' motion features. Since most single-frame detection algorithms cannot estimate object movement accurately, such as velocity acceleration, etc., heading angle is one of the most important motion features to introduce for DA.

As for the target update module, it aims to refine the current frame's object candidate features for DA. The target update module simply takes the current frame's Q-in as the input and passes through a two-layer multilayer-perceptron (MLP) to prepare for DA because feature updating is not necessary for current frame's features. Since the previously detected objects might disappear at the current frame, an empty vector (filled with zeros) is concatenated to the Q-in, which we call ”dead query features,” to represent the disappeared object queries.

Finally, the previous frame's updated query features are associated with the current frame's refined target features through a dot product process. The output from the dot product operation is an N by $M + 1$ matrix where N is the number of detected objects from the previous frame, and M is the number of object queries in the current frame. A higher value in the matrix (association score) indicates a higher possibility of an association. During the training phase, we treat the association process as a classification task and compute the loss between the object association module estimated results and the ground truth results with the cross-entropy loss function. During the evaluation phase, we apply the same method as the training phase but employ a greedy-based search method after the object association module to prevent duplicate association.

Extra DA Module: The extra DA module is designed for the LiDAR-image fusion detector. The architecture design is the same as the LiDAR-only DA module. The difference is that the query feature update model uses the previous frame's Q-fine, Q-feat, and detected objects' heading angles as the input. Moreover, the extra DA requires a further decision-making step during the evaluation phase. When both DA modules estimate the same tracklet results, the final association is determined by the extra association module. When there is a conflict between the two association modules, the final association results are determined by the highest association score.



	P_0	P_1	P_2
Q_0	0.03	0.13	1.89
Q_1	0.83	0.01	0.27
Q_2	2.31	0.39	2.15
Q_3	0.12	0.81	0.11
$dead$	0.05	3.16	0.04

Figure 4. Object Association Module. The cross marked rectangle represents dead objects. When a duplicate association occurs, the association outcomes are determined by greedy matching (e.g. Q_2 versus P_0 and P_2).

3.3. Query Enhancement Module

The objective of the QEM is to imbue the current frame's Q-in with the previous frame's Q-feat or Q-f2 to improve the detection performance, and the overall architecture is shown in Figure 5. In the QEM, the previous frame's Q-feat or Q-fine, and the current frame's Q-in are passed into a cross-attention layer to aggregate the previous frame's features to the current frames' corresponding queries. In this way, the history temporal information is aggregated to the current frame. Furthermore, such a cross-attention mechanism can prevent unnecessary or even misleading information contaminate the current frame's features through model learning. J. Koh, et al. have conducted a similar operation for temporal information aggregation. [17]

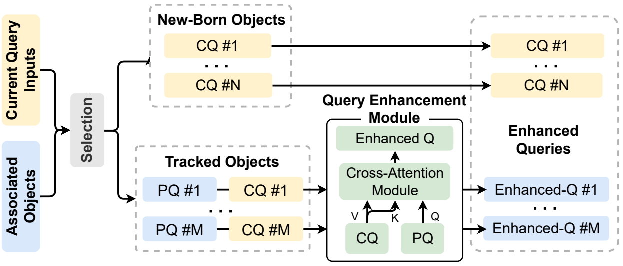


Figure 5. Query Enhancement Module. CQ represents the current frame's queries, PQ represents the previous frame's queries, and Enhanced-Q represents the enhanced queries. For the new-born objects, the query inputs are directly used as the query input to the transformer decoder.

3.4. Training And Evaluation Model Differences

The differences between the training phase and evaluation phase are the number of input frames, object detection decision-making, and object association processing. For the training phase, the number of input frames is set to 2, while for the evaluation phase, the number of input frames is 1 at a time. As for the object detection decision-making, we follow the same procedure with [1]. For the object association process, the association results are processed in a manner of classification task. During training, the association module's outputs are compared with the ground truth associations results and the loss are computed with a cross-entropy equation. In the LiDAR-image setup, both association modules' output tracklets are compared with the ground truth associations results for loss computation. In the evaluation phase, the association results are further processed with a greedy-based search algorithm to prevent duplicate associations. The overall training procedures can be summarized as

1. Train the TransFusion model by following the instructions from [1].
2. Train the MotionTrack model with a DA module based on the TransFusion model's checkpoint. In this step, all parameters are frozen except the DA module.
3. Train the MotionTrack model with both the DA and the QEM. In this process, all parameters are learned except the backbones.

4. RESULTS & DISCUSSIONS

The MotionTrack model and its variants are evaluated by the nuScenes dataset. The results comprise the detection and the tracking results.

4.1. Imeplementation Details

The input data of the MotionTrack is a 360-degree point cloud and 6 RGB images that capture the ego vehicle's surrounding views. As for the training and validation set split, we use the official split of the training, validation, and evaluation dataset. All training experiments are conducted with eight A100 80GB GPUs. As for the training parameters, we use the AdamW optimizer with one cycle learning rate policy, with a max learning rate 0.001, weight decay 0.01, and momentum 0.85 to 0.95. Since the number of tracked objects varies across all training samples, we set the training batch size to be one per GPU.

4.2. MotionTrack Detection Results

The MotionTrack detection results of the nuScenes dataset are presented in Table 2. As for the detection results, the VoxelNet backbone performs better than the PointPillar since the former voxelized the original point cloud data with smaller segments. Furthermore, the VoxelNet backbone achieves better results on small objects such as pedestrians, motorcycles, and bicycles. The MotionTrack-Voxel's AP performances are 13.03%, 44.87%, and 112.45% higher than the MotionTrack-Pillar's on pedestrian, motorcycle, and bicycle. Even though the MotionTrack-Voxel outperforms the MotionTrack-Pillar in all object categories, both models' performance for the car class is similar because the objects size is relatively large and the number of training samples is large. Moreover, the MotionTrack-Pillar is more efficient where the model size is 36.4% smaller than the MotionTrack-Voxel.

Table 2. Detection results on nuScenes validation dataset.

Model	Classes	AP ↑	ATE ↓	AAE ↓
MotionTrack-Pillar	Bicycle	0.27	0.25	0.02
	Bus	0.64	0.40	0.31
	Car	0.83	0.21	0.19
	Motorcycle	0.50	0.25	0.16
	Pedestrian	0.78	0.15	0.09
	Trailer	0.36	0.57	0.21
	Truck	0.50	0.39	0.22
MotionTrack-Voxel	Bicycle	**0.58**	0.16	0.01
	Bus	**0.73**	0.35	0.26
	Car	**0.87**	0.17	0.20
	Motorcycle	**0.72**	0.20	0.23
	Pedestrian	**0.88**	0.13	0.08
	Trailer	**0.44**	0.52	0.17
	Truck	**0.60**	0.33	0.24

4.3. MotionTrack Tracking Results

We compared the MotionTrack with other baseline tracking algorithms, such as the AB3DMOT, CenterTrack, and Probabilistic 3D Kalman filter in the nuScenes test dataset (Table 3). The selected algorithms contain both ML-based and traditional tracking methods. The main evaluation metric for MOT is AMOTA, which is integrals over the MOTA metric using n-point interpolation. The AMOTA equation is available in [5]

According to the AMOTA results, the proposed MotionTrack baseline is 3.7x higher than the CenterTrack, 2.3x higher than the AB3DMOT, and on par with the probabilistic 3D Kalman filter model. This result proves that a simple transformer-based association algorithm can achieve MOT under an autonomous driving environment. Furthermore, the improvement compared with the traditional Kalman filter-based method and the tracking-by-detection paradigm indicates that the simultaneous tracking and detection paradigm with transformer architecture has huge potential. According to the comparison results, we found that the high object ID switching is the reason that affects the overall tracking performance. The reason cause such an issue is that we don't introduce a carefully designed tracking management algorithm during the inference time.

Table 4 tabulates the MotionTrack's tracking results among all categories. According to Table 4, the AMOTA results are proportional to the number of object samples. The reason is that the transformer DA module requires numerous samples to learn the objects' features. Furthermore, Table 4's results indicate that larger objects (car, bus, and truck) exhibit better performance than small objects, especially for the MotionTrack with LiDAR-only input. Even though the multi-sweep LiDAR input method is applied, the point cloud is still sparse for small objects. Therefore, the association accuracy is decreased due to poor object features. Such poor tracking performance is solved by the MotionTrack with image-LiDAR input because of the dense appearance features obtained from images. According to Table 4, the pedestrian and motorcycle AMOTA results for MotionTrack with image-LiDAR input are 5.6% and 11.2% higher than the MotionTrack with LiDAR-only.

4.4. Ablation Studies

We conduct two ablation studies to validate the importance of the transformer-based DA module and the QEM.

Transformer-based Data Association The objective of the transformer module is to refine and update the object features based on previous frame objects' appearance, position, and heading angle features. Furthermore, a simple dot product computation cannot accurately associate objects between consecutive frames. In the DA module ablation study, we conduct two experiments: (1) A MotionTrack model contains the transformer process so that both the previous frame's object features and the current frame's queries are processed through the transformer module before dot product association (2) A MotionTrack model without transformer-based DA so that the previous frame's objects' features and the current frame's queries are directly associated through dot product. According to Table 5, the transformer-based DA process is necessary since the AMOTA is almost 3x higher than the one without the transformer process. Due to the hardware limitations and the complex ego vehicle's dynamic behavior, autonomous vehicle's data quality is inferior to other tracking-related data such as the MOT16. For instance, the sampling frequency of the nuScenes dataset is only 2 Hz, and the target objects' local movement is dependent on the ego vehicle's motion.

QEM The QEM helps with the detection performance while the tracking performance does not significantly improve. The objective of the QEM is to employ temporal features for object detection. According to Table 6, the proposed QEM improves the detection performance by 6% and 3% for the MotionTrack with LiDAR-only input and the image-LiDAR input. Even though better detection results, the tracking performance is not improved accordingly, which against conventional intuition. According to our analysis, when detection performance improved, the ID switch is increased correspondingly. Since the MotionTrack only associates objects between consecutive frames without further track management (e.g. disappear objects buffer), increasing the number of detected bounding boxes increases the possibility of wrong objects associations across frames. Therefore, even though the proposed query enhancement is helpful with object detection but the tracking performance does not improve accordingly. We believe that QEM can help with the overall tracking performance with a well-designed track management algorithm and longer input frames during the training phase.

4.5. Discussions and Potential Improvements

Although MotionTrack is more performant than other baselines, we are aware that it cannot compete with SOTA algorithms such as the ImmortalTrack [36], CAMO-MOT [35], and ByteTrack [47]. Nevertheless, this paper's objective is to provide a good starting point for multi-modality end-to-end transformer-based MOT research. Below, we discuss MotionTrack's four potential improvements.

First, longer input frames for training can improve the robustness of the association against occlusion. MotionTrack set the input frame to 2, which do not contain relatively long temporal information to simulate the actual tracking process. This issue is also reflected by the poor object ID switch results. With a longer number of input frames, the association algorithm can learn associations from more complex cases, such as occlusion.

Second, a better track management module integrated with model inference can help perform object reidentification (ReID). Currently, MotionTrack only considers DA between consecutive frames. If a previous frame's object is failed to associate with any current frame's object, that previous frame's object is directly considered a dead object. Existing algorithms provide a "disappear object" buffer during the inference phase so that the non-associated objects can be associated again with future frames to prevent tracking loss due to occlusion. MotionTrack's MOT performance can be improved after introducing such similar design.

New tracking-oriented data augmentation methods is the third improvement we propose for potential improvement. MotionTrack's current data augmentation methods are mainly designed for MOD, such as random flips, rotations, and scales. We realize that there are several augmentation techniques, such as randomly dropping tracked objects and adding false positive objects, but these techniques only marginally improve the MotionTrack's tracking performance. To further improve the robustness of the DA and QEM, more effective data augmentation algorithms are required.

The last potential improvement is to properly process the MotionTrack with a larger batch size used in model training, in order to speed things up without dramatically increasing the memory footprint. This issue also occurred with

Table 3. MotionTrack compared with other baselines on nuScenes test dataset. Bold and underlined text represent the top and second results.

Model Name	Modality	AMOTA ↑	AMOTP ↓	MOTA ↑	MOTAR ↑	MOTAP ↓	FAF ↓	MT ↑	ML ↓	IDS ↓
PointPillar+AB3DMOT	L	0.03	1.70	0.05	0.24	0.82	220.9	480	5332	7548
Prob-3DKalman	L	0.55	**0.80**	0.46	0.77	0.35	54.5	**4294**	2184	**950**
CenterTrack	L+C	0.11	0.99	0.09	0.27	0.35	206.6	1308	3739	7608
AB3DMOT	L+C	0.15	1.50	0.28	0.55	0.15	55.8	1006	4428	9027
MotionTrack$_{\text{Pillar-L}}$	L	0.42	1.01	0.385	0.74	0.34	42.8	3850	2758	10139
MotionTrack$_{\text{Voxel-L}}$	L	0.51	0.99	0.48	**0.83**	0.30	28.4	3723	1567	9705
MotionTrack$_{\text{Pillar-LC}}$	L+C	0.45	0.90	0.48	0.59	0.31	32.7	3014	1815	9943
MotionTrack$_{\text{Voxel-LC}}$	L+C	**0.55**	0.871	**0.49**	0.77	**0.26**	**22.4**	4211	**1321**	8716

Table 4. MotionTrack tracking results among all categories on nuScenes validation dataset.

					MotionTrack-Voxel-LiDAR					
Classes	**AMOTA ↑**	**AMOTP ↓**	**Recall ↑**	**MOTAR ↑**	**GT**	**MOTA ↑**	**MOTP ↓**	**MT ↑**	**ML ↓**	**FAF ↓**	**IDS ↓**
Bicycle	0.40	0.22	0.48	0.90	1993	0.43	0.21	28	68	6.5	11
Bus	0.73	0.61	0.79	0.90	2112	0.68	0.40	61	10	9.6	74
Car	0.85	0.19	0.74	0.93	58317	0.61	0.21	1856	637	45.2	4631
Motorcycle	0.60	0.47	0.67	0.92	1977	0.56	0.27	58	21	7	126
Pedestrian	0.80	0.22	0.75	0.87	25423	0.60	0.23	923	319	50.9	15678
Trailer	0.37	0.57	0.56	0.72	2425	0.36	0.57	53	50	33.7	143
Truck	**0.57**	0.34	0.63	0.75	9650	0.46	0.34	210	150	39.1	204
					MotionTrack-Voxel-ImageLiDAR					
Classes	**AMOTA ↑**	**AMOTP ↓**	**Recall ↑**	**MOTAR ↑**	**GT**	**MOTA ↑**	**MOTP ↓**	**MT ↑**	**ML ↓**	**FAF ↓**	**IDS ↓**
Bicycle	**0.41**	0.20	0.50	0.95	1993	0.51	0.21	28	65	6.3	11
Bus	**0.78**	0.55	0.85	1.06	2112	0.76	0.44	71	10	8.0	62
Car	**0.86**	0.15	0.83	0.99	58317	0.66	0.19	2150	589	45.1	4403
Motorcycle	**0.66**	0.45	0.72	1.07	1977	0.64	0.17	90	19	6.6	111
Pedestrian	**0.84**	0.24	0.80	1.04	25423	0.65	0.22	1073	304	45.5	15081
Trailer	**0.38**	0.54	0.59	0.86	2425	0.42	0.52	54	49	32.4	139
Truck	0.56	0.31	0.71	0.87	9650	0.54	0.28	219	148	31.9	193

Table 5. DA module ablation study on nuScenes validation dataset.

Model	Module	AMOTA	MOTA
MotionTrack$_{\text{Voxel-L}}$	w/ Transformer	0.62	0.53
	w/o Transformer	0.22	0.20
MotionTrack$_{\text{Voxel-LC}}$	w/ Transformer	0.69	0.61
	w/o Transformer	0.23	0.20

Table 6. QEM ablation study for car class on nuScenes validation dataset.

Model	Module	mAP	AMOTA
MotionTrack$_{\text{Voxel-L}}$	w/ Query Enhance	0.87	0.85
	w/o Query Enhance	0.81	0.85
MotionTrack$_{\text{Voxel-LC}}$	w/ Query Enhance	0.88	0.93
	w/o Query Enhance	0.86	0.93

other end-to-end tracking algorithms, such as the MOTR and the MeMOT algorithm. The reason is that the number of detected objects are varied among different frames, which causes the dimension of the previous frame's object features to be inconsistent. Currently, a common solution is to concatenate zero vectors to represent empty objects so that the object features' dimensions are the same across training samples. However, this method causes the memory footprint and model sizes increase, especially for transformer architecture, which can potentially exceed certain GPUs memory cap. Therefore, a design that can process the MotionTrack with higher batch sizes without wasting the memory footprint can speed up the training.

In summary, the proposed MotionTrack is a baseline for multi-modality end-to-end transformer-based MOT. Our results indicate a huge potential for transformer-based MOT.

5. CONCLUSIONS

This paper proposes a novel simultaneous detection and tracking baseline algorithm, MotionTrack, with multi-modality sensors inputs under autonomous driving environment. MotionTrack proves that the transformer-based algorithm is suitable for MOT under the autonomous driving environment. Furthermore, MotionTrack validates that the self- and the cross-attention mechanism is capable of objects' association with multiple classes. Finally, we propose a transformer-based query update algorithm, QEM, to refine the potential object queries from history frames to improve the overall detection performance.

MotionTrack's tracking results outperform other baseline algorithms on the nuScenes dataset. The current drawbacks and potential improvements to MotionTrack are elaborated in the results section. We believe the MotionTrack can be used as a new baseline algorithm for future deep learning-based end-to-end tracking algorithms in the autonomous driving environment.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. TransFusion: Robust LiDAR-camera fusion for 3d object detection with transformers. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1080–1089, 2022. 4, 6

[2] Mk Bashar, Samia Islam, Kashifa Kawaakib Hussain, Md. Bakhtiar Hasan, A. B. M. Ashikur Rahman, and Md. Hasanul Kabir. Multiple object tracking in recent times: A literature review. *ArXiv*, abs/2209.04796, 2022. 1

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 941–951, 2019. 2

[4] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 2

[5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2019. 6

[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, 2022. 1, 2

[7] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 2

[8] Azim Eskandarian, Chaoxian Wu, and Chuanyang Sun. Research advances and challenges of autonomous and connected ground vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):683–711, 2021. 1

[9] Jiaojiao Fang and Guizhong Liu. Visual object tracking based on mutual learning between cohort multiscale feature-fusion networks with weighted loss. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(3):1055–1065, 2021. 1

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016. 3

[11] Karl Granström and Marcus Baum. Extended object tracking: Introduction, overview and applications. *ArXiv*, abs/1604.00970, 2016. 4

[12] JunYoung Gwak, Silvio Savarese, and Jeannette Bohg. Minkowski tracker: A sparse spatio-temporal R-CNN for joint object detection and tracking. *ArXiv*, abs/2208.10056, 2022. 3

[13] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 4

[14] Kemiao Huang and Qi Hao. Joint multi-object detection and tracking with camera-LiDAR fusion for autonomous driving. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6983–6989, 2021. 3

[15] Zhiyu Huang, Xiaoyu Mo, and Chen Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2605–2611, 2022. 1

[16] Aleksandr Kim, Aljosa Osep, and Laura Leal-Taixé. EagerMOT: 3d multi-object tracking via sensor fusion. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11315–11321, 2021. 3

[17] Junho Koh, Jaekyum Kim, Jin Hyeok Yoo, Yecheol Kim, Dongsuk Kum, and Junghwan Choi. Joint 3d object detection and tracking using spatio-temporal representation of camera image and lidar point clouds. *ArXiv*, abs/2112.07116, 2021. 5

[18] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, 2018. 4

[19] Alan Lukezic, Tomas Vojir, Luka ˇCehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6309–6318, 2017. 1

[20] Chenxu Luo, Xiaodong Yang, and Alan Loddon Yuille. Exploring simple 3d multi-object tracking for autonomous driving. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10468–10477, 2021. 3

[21] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixé, and Christoph Feichtenhofer. TrackFormer: Multi-object tracking with transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8834–8844, 2022. 1, 2

[22] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4293–4302, 2016. 1

[23] Ziqi Pang, Jie Li, Pavel Tokmakov, Di Chen, Sergey Zagoruyko, and Yu-Xiong Wang. Standing between past and future: Spatio-temporal modeling for multi-camera 3d multi-object tracking. *ArXiv*, abs/2302.03802, 2023. 3

[24] Ziqi Pang, Zhichao Li, and Naiyan Wang. SimpleTrack: Understanding and rethinking 3d multi-object tracking. *ArXiv*, abs/2111.09621, 2021. 3

[25] Lionel Rakai, Huansheng Song, ShiJie Sun, Wentao Zhang, and Yanni Yang. Data association in multiple object tracking: A survey of recent techniques. *Expert Systems with Applications*, 192:116300, 2022. 2

[26] Felicia Ruppel, Florian Faion, Claudius Gläser, and Klaus C. J. Dietmayer. Transformers for multi-object tracking on

point clouds. *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 852–859, 2022. 1

[27] Samuel Scheidegger, Joachim Benjaminsson, Emil Rosenberg, Amrit Krishnan, and Karl Granström. Mono-camera 3d multi-object tracking using deep learning detections and PMBM filtering. *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 433–440, 2018. 3

[28] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. Deep network flow for multi-object tracking. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2730–2739, 2017. 2

[29] Sarthak Sharma, Junaid Ahmed Ansari, Krishna Murthy Jatavallabhula, and K. Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3508–3515, 2018. 2

[30] Apoorv Singh. Transformer-based sensor fusion for autonomous driving: A survey. *ArXiv*, abs/2302.11481, 2023. 1

[31] Apoorv Singh and Varun Bankiti. Surround-view vision-based 3d detection for autonomous driving: A survey. *ArXiv*, abs/2302.06650, 2023. 1

[32] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. TransTrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 1, 2

[33] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10840–10849, 2021. 2

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 4

[35] Li Wang, Xinyu Newman Zhang, Wenyuan Qin, Xiaoyu Li, Lei Yang, Zhiwei Li, Lei Zhu, Hong Wang, Jun Li, and Hua Liu. Camo-mot: Combined appearance-motion optimization for 3d multi-object tracking with camera-LiDAR fusion. *ArXiv*, abs/2209.02540, 2022. 3, 7

[36] Qitai Wang, Yuntao Chen, Ziqi Pang, Naiyan Wang, and Zhaoxiang Zhang. Immortal tracker: Tracklet never dies. *ArXiv*, abs/2111.13672, 2021. 7

[37] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *ArXiv*, abs/1909.12605, 2019. 2

[38] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics. *ArXiv*, abs/2008.08063, 2020. 3

[39] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017. 2

[40] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. TransCenter: Transformers with dense representations for multiple-object tracking. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021. 2

[41] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*, 2022. 1

[42] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-end multiple-object tracking with transformer. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, page 659–675, Berlin, Heidelberg, 2022. Springer-Verlag. 1, 2

[43] Ce Zhang and Azim Eskandarian. A quality index metric and method for online self-assessment of autonomous vehicles sensory perception. *arXiv preprint arXiv:2203.02588*, 2022. 1

[44] Ce Zhang, Azim Eskandarian, and Xuelai Du. Attention-based neural network for driving environment complexity perception. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2781–2787. IEEE, 2021. 1

[45] Renrui Zhang, Han Qiu, Tai Wang, Xuanzhuo Xu, Ziyu Guo, Yu Qiao, Peng Gao, and Hongsheng Li. MonoDETR: Depth-aware transformer for monocular 3d object detection. *arXiv preprint arXiv:2203.13310*, 2022. 1

[46] Tianyuan Zhang, Xuanyao Chen, Yue Wang, Yilun Wang, and Hang Zhao. Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4536–4545, 2022. 3

[47] Yifu Zhang, Pei Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, 2021. 2, 7

[48] Yangheng Zhao, Jun Wang, Xiaolong Li, Yue Hu, Ce Zhang, Yanfeng Wang, and Siheng Chen. Number-adaptive prototype learning for 3d point cloud semantic segmentation. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *Computer Vision – ECCV 2022 Workshops*, pages 695–703, Cham, 2023. Springer Nature Switzerland. 1

[49] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, pages 474–490. Springer, 2020. 2

[50] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Phillip Krähenbühl. Global tracking transformers. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8761–8770, 2022. 2

[51] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3d object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2017. 4

[52] Tianyu Zhu, Markus Hiller, Mahsa Ehsanpour, Rongkai Ma, Tom Drummond, and Hamid Rezatofighi. Looking beyond two frames: End-to-end multi-object tracking using spatial and temporal transformers. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021. 2