TGFormer: Transformer with Track Query Group for Multi-Object Tracking

Rui Zeng, Yuanzhou Huang, Songwei Pei*

School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China
{zengrui@, huangyuanzhou@, peisongwei@}bupt.edu.cn

Abstract

Multi-object tracking faces a major challenge in handling the variations of tracked targets within complex scenes. In existing transformer-based tracking methods, typically each tracked target is only associated with one track query. However, trajectories in crowded scenes often experience varying levels of occlusion, making the association brittle for using a single track query to identify the tracked target. Therefore, we argue that relying on a single track query to track a target in complex scenes is inadequate. In this paper, we introduce TGFormer, with the core idea of designing a Track Query Group for each tracked target. Each group encompasses track queries that handle the same tracked target across different levels of occlusion scenes. To achieve long-term robust association, we propose a novel updater that integrates temporal memories and occlusion-aware features to update the Track Query Group, ensuring the tracked target can be consistently captured in complex scenes. Additionally, we introduce a Position Predictor that allows TGFormer to forecast motion trends, helping the model accurately locate moving tracklets. Experimental results show that our method achieves competitive performance on the MOT Challenge and DanceTrack datasets.

Introduction

Multi-Object Tracking (MOT) is a crucial task in computer vision, widely applied in real-world scenarios such as autonomous driving, surveillance, and military applications. Traditional solutions (Wang et al. 2020) follow the Tracking-by-Detection (TbD) paradigm, decomposing the MOT task into two stages: object detection and data association. Although these methods achieve high detection performance, they encounter difficulties with target association under severe occlusion and target variation scenarios (Sun et al. 2022), mainly due to the lack of global (end-to-end) optimization. Recently, the transformer-based trackers following the Tracking-by-Attention (TbA) paradigm have gained popularity. By jointly adopting detect queries and track queries in the transformer decoder to interact with image features, these methods achieve fully end-to-end tracking. Specifically, the detect queries are responsible for recognizing newborn objects, and the track queries serve to prioritize their assigned targets.

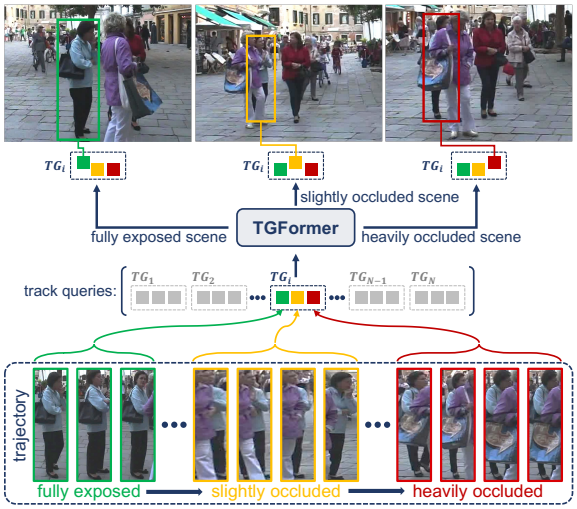

Figure 1: **Illustration of the Track Query Group**. TGFormer identifies each tracked target using a Track Query Group (TG), with each group containing up to three track queries. These track queries are responsible for associating the same tracked target under different levels of occlusion: fully exposed, slightly occluded, and heavily occluded. Unlike most previous transformer-based trackers that use a single track query to associate the tracked target, our model uses a track query group to further leverage multiple appearance features from the trajectory based on their occlusion levels, thereby enhancing tracking performance in challenge scenarios.

Despite the simplicity of these transformer-based trackers, each tracked result relies on a single track query, which can make the queries brittle to losing their targets in crowded scenes. This issue arises because targets in complex scenes may experience varying levels of occlusion and frequent changes in appearance, causing a single track query to struggle with maintaining accurate matching. Although some methods have incorporated long-term temporal information to establish a more robust feature representation for the target, a single track query still has difficulty adapting to target variations in complex scenes, primarily due to the lack of a categorical representation for the target across different lev-

*Corresponding author.

els of occlusion.

Based on the above analysis, this paper proposes a transformer-based Multi-Object Tracking method, which identifies the same target at different occlusion levels with a **T**rack Query **G**roup, called **TGFormer**. As shown in Fig. 1, TGFormer generates a group of track queries for each trajectory, where the group stores the content features of the tracked target across different occlusion levels. Specifically, we categorize occlusion levels into three types: fully exposed, slightly occluded, and heavily occluded. For a newly detected object, TGFormer initializes a Track Query Group and incorporates the output content and position embeddings as track queries within the group. When the occlusion level of the tracked target changes and the new occlusion level is not present in the track query group, we add the newly output embeddings as new track queries to the group. Compared to tracking each target with a single track query, using a track query group for the association is more robust because TGFormer can adapt to the occlusion level of the target by selecting the most suitable prediction as the tracking result. To ensure long-term tracking, all track queries in the group are consistently updated at each frame using the proposed Track Query Group Updater. First, we fuse the queries in the group with the output embedding obtained from the current frame respectively by Short-Term Memory Aggregator. Since the target exhibits unique features at different occlusion levels, we apply an Occlusion-Aware Memory-Attention mechanism to differentiate the features of the target across different occlusion levels. Furthermore, followed by the ideas of MeMOTR (Gao and Wang 2023) and MeMOT (Cai et al. 2022), we maintain a long-term memory for the target, which is connected to the output of the Memory-Attention via residual connections, helping the model learn more distinctive representations.

Additionally, we consider positional information in queries affects object detection accuracy, especially when objects move rapidly, as previous frame information may not reflect the current position. Traditional tracking methods use Kalman filters (Welch and Bishop 1995) for position prediction, but this has been overlooked in transformer-based methods. Leveraging DAB-DETR's (Liu et al. 2022) separation of content and positional features, we capture historical positional information and use a lightweight LSTM network (Hochreiter and Schmidhuber 1997) to predict the target's position in the next frame. This predicted position, combined with content features, is input into the decoder, aiding in more accurate localization of moving targets. We evaluate our method on the MOT Challenge and DanceTrack datasets. Experimental results show that our method improves tracking metrics over the baseline, achieving highly competitive performance among transformer-based methods.

In summary, the contributions of this paper are listed below:

- We introduce a transformer-based multi-object tracking method, TGFormer. Instead of tracking each target with a single track query, TGFormer employs a Track Query Group consisting of multiple track queries that adapt to various occlusion levels, thereby achieving robust tracking in the challenge scene.
- To ensure accurate long-term tracking, we propose Track Query Group Updater and Position Predictor. The updater integrates temporal memories and occlusion-aware features for the track queries in each group, while the Position Predictor forecasts the motion trend to improve localization performance.
- TGFormer achieves highly competitive performance on the MOT Challenge and DanceTrack datasets. Extensive ablation experiments further demonstrate the effectiveness of our method.

Related Work

Tracking-by-Detection Trackers Most current multi-object tracking algorithms follow the Tracking-by-Detection paradigm, using detectors like YOLOX (Ge et al. 2021) to locate objects and extract re-ID features, with frame-to-frame association via IoU (Yang et al. 2023) or re-ID (Ristani and Tomasi 2018). SORT (Bewley et al. 2016) is a classic algorithm in this category, using the FrRCNN detector (Ren et al. 2015) for object localization and combining a Kalman Filter (Welch and Bishop 1995) with the Hungarian matching for data association. Enhancements such as Deep SORT, BoT-SORT, and Deep OC-SORT (Wojke, Bewley, and Paulus 2017; Aharon, Orfaig, and Bobrovsky 2022; Maggiolino et al. 2023) integrate appearance features and camera motion compensation. While these two-stage trackers achieve strong results with high-quality detectors and efficient associations, they rely heavily on detection quality, lack end-to-end training, and require post-processing like NMS (Hosang, Benenson, and Schiele 2017), limiting global optimization (Meinhardt et al. 2022).

Transformer-based Trackers With the rise of DETR-like (Carion et al. 2020; Liu et al. 2022) models in object detection, several advancements have been made in end-to-end multi-object tracking using query-key mechanisms. Trackformer (Meinhardt et al. 2022) and MOTR (Zeng et al. 2022) combine detection and data association by using detect queries and track queries in the decoder. TransTrack (Sun et al. 2021) avoids suppression effects by using separate decoders for the two types of queries and matches targets via IoU. MOTRv2 (Zhang, Wang, and Zhang 2023) enhances detection with an additional detector, while MOTRv3 (Yu et al. 2023) improves label assignment with release-fetch supervision and pseudo-label distillation. MO3TR (Zhu et al. 2023) employs a temporal attention module to update features in the Spatial Transformer. MeMOT (Cai et al. 2022) uses a large spatiotemporal memory for identity embeddings, preserving valuable long-term information. Co-MOT (Yan et al. 2023) introduces a competitive-cooperative label assignment to address training imbalances. MeMOTR (Gao and Wang 2023) enhances target association by incorporating long-term memory to stabilize and distinguish track embeddings. These transformer-based trackers typically use a single track query to associate each target, overlooking challenges from varying occlusion levels. While some models introduce query group to accelerate training convergence, we aim to implement the Track Query Group in online tracking to better adapt to target variations in complex scenarios.

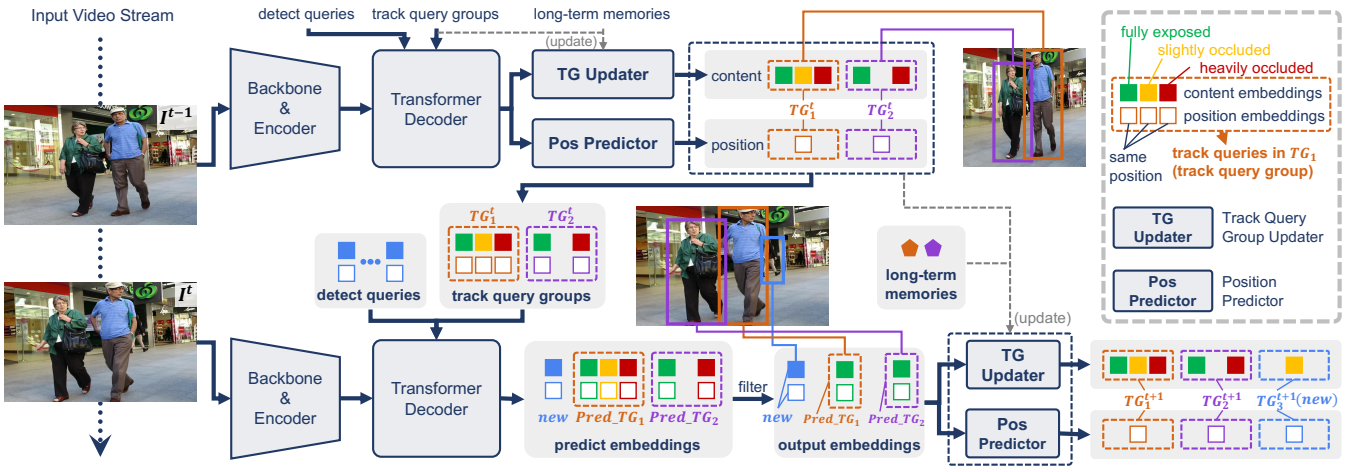

Figure 2: **Visualization of TGFormer**. Unlike tracking each target using a single track query in Transformer, TGFormer identifies each tracked target with a Track Query Group (TG). In each group, the track queries predict the same target. Specifically, these queries share the same positional information but have distinct content embeddings that adapt to different occlusion levels. To achieve long-term tracking, TGFormer introduces Track Query Group Updater (TG Updater) and Position Predictor to update the queries in the Track Query Group. The updater integrates the temporal and occlusion-aware content features, while the Position Predictor predicts the position in the next frame using the previously historical positional information.

The Method

Overview

In this paper, we propose TGFormer, a multi-object tracking model that assigns a Track Query Group to each tracked object. Unlike most existing methods that use a single track query to associate a target, our approach generates a Track Query Group for each object, storing multiple queries across different occlusion levels during the tracking, ensuring that the target can be accurately identified in complex scenarios. Additionally, we maintain a long-term memory for each target to record its stable general features.

As shown in Fig. 2, for the frame I^t of the video input stream, we first input it into the backbone and transformer encoder to extract the 2D features as one of the inputs to the decoder. We use a fixed number of learnable embeddings named detect queries, to detect newborn objects in this frame and each tracked target corresponds to a Track Query Group TG_i^t. We use the Position Predictor to predict target positional information from historical trajectories, which is combined with content embeddings to form TG_i^t for input into the decoder. The decoder generates embeddings for all queries in the group, which are processed by the prediction network to obtain confidence scores and bounding boxes. The model selects the most suitable prediction based on these results and uses the corresponding decoder output as the output embedding. In the post-processing stage, the output embedding, long-term memory, and TG_i^t are fed into the Track Query Group Updater to update the queries in the group. More details are provided in the following sections.

Track Query Group

The traditional method that each tracked target is associated only with one track query is effective when the occlusion level of the target is stable. However, in complex scenarios involving different occlusion levels, it often fails as a single track query cannot handle this challenge, leading to numerous ID switches. Our core idea is to design different queries to targets with different occlusion levels in complex scenes, where each query is responsible for identifying the target at different occlusion levels.

We classify occlusion levels into three distinct categories: fully exposed, slightly occluded and heavily occluded. Since the occlusion level of the target changes in complex scenes, the detection confidence of queries in the group varies. Typically, confidence is high when the target is fully exposed, decreases when slightly occluded, and is low when heavily occluded. So we set three types of queries within the group, with confidence thresholds τ_{high}, τ_{mid}, and τ_{low}. During the tracking, based on the confidence scores predicted by the query, the embeddings are classified into low score embedding, mid score embedding, and high score embedding.

For each newly detected object, TGFormer initializes a Track Query Group, which serves as a dedicated set of queries for tracking the target. This group is initialized by incorporating both the output content embedding and the position embedding, effectively transforming them into track queries that will be utilized for subsequent tracking tasks within the group. The queries in the Group are input into the decoder to get their embeddings, and they are processed through MLP to predict the corresponding confidence scores and bounding boxes. During training, we compute the loss between the predictions of all queries in the Track Query Group and the ground truth. Since high, mid, and low score embeddings are determined by their confidence scores, equally weighting the classification and bounding box loss is unfair. Therefore, we assign a higher weight to the bounding box loss, prioritizing queries that predict more accurate

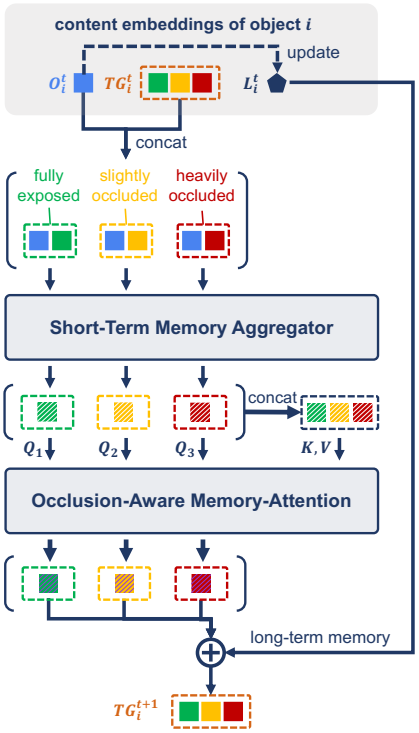

Figure 3: **Illustration of Track Query Group Updater**. For the i-th tracked target at frame t, the updater takes three inputs: output embedding O_i^t, content embeddings from Track Query Group TG_i^t, and long-term memory L_i^t. First, the newly tracked O_i^t is concatenated with each content embedding in TG_i^t for Short-Term Memory Aggregation. Since each content embedding contains distinct semantic information related to occlusion levels, a Memory-Attention mechanism is then used to extract and differentiate occlusion-aware content features. Finally, the Track Query Group TG_i^{t+1} are aggregated by applying a residual connection between the attention results and L_i^t.

bounding boxes. During inference, we prioritize selecting the result with the highest confidence score. Correspondingly, the selected embedding is used as the output embedding in the subsequent update process. When the detection confidence of the output embedding changes into the threshold range of other embeddings, it indicates that the occlusion level of the target has changed. In this case, we will either add it to the group if the new occlusion level is not already represented, or we will update the previously existing embedding in the group.

Track Query Group Updater

In previous transformer-based trackers, the detect query from the previous frame is passed to the next frame as a track query if an object is detected, iteratively updating the content features of the query. However, it doesn't work for our method because multiple queries exist within a group. Therefore, we design a specialized updater for the Track Query Group. Followed by MeMOTR and MeMOT (Gao and Wang 2023; Cai et al. 2022), we use a long-term memory embedding L_i^t to represent the stable general feature of the target. When a newborn object is detected, its output embedding O_i^t is initialized as L_i^t and we apply an effective running average with exponentially decaying weights to update it like MeMOTR (Gao and Wang 2023).

In the frame I^t, the updater receives the output embedding O^t, the Track Query Group TG^t and long-term memory L^t as input. As shown in Fig. 3, each type of query in TG^t is concatenated with O^t and input into the Short-Term Memory Aggregator which consists of an MLP:

$$\begin{cases} S_{high}^t = \text{MLP}([O^t, TG_{high}^t]) \\ S_{mid}^t = \text{MLP}([O^t, TG_{mid}^t]) \\ S_{low}^t = \text{MLP}([O^t, TG_{low}^t]) \end{cases}. \tag{1}$$

This approach aims to enhance target feature representation by concatenating multi-frame features and to extract and fuse target features across various scenarios through the integration of high, mid, and low score embeddings.

After passing through the Aggregator, we obtain three sets of embeddings S_{high}^t, S_{mid}^t, and S_{low}^t respectively. These embeddings serve the same target but contain unique features for different occlusion levels. To differentiate these unique features, we design a multi-head attention module called Occlusion-Aware Memory-Attention. The three embeddings are concatenated $S_{cat}^t = concat([S_{high}^t, S_{mid}^t, S_{low}^t])$ to construct the key and value, while embeddings from different occlusion levels serve as queries to facilitate feature interaction. The features obtained from the attention layer represent unique features of the target at different occlusion levels, and relying solely on these features for target detection may not be sufficient, as demonstrated in our experiments. Therefore, we use residual connections with the stored long-term memory L^t like:

$$\begin{cases} TG_{high}^{t+1} = \text{Attention}(S_{high}^t, S_{cat}^t, S_{cat}^t) + L^t \\ TG_{mid}^{t+1} = \text{Attention}(S_{mid}^t, S_{cat}^t, S_{cat}^t) + L^t \\ TG_{low}^{t+1} = \text{Attention}(S_{low}^t, S_{cat}^t, S_{cat}^t) + L^t \end{cases}. \tag{2}$$

Finally we get the updated TG_i^{t+1}, which is used as the input for the next frame:

$$TG^{t+1} = [TG_{high}^{t+1}, TG_{mid}^{t+1}, TG_{low}^{t+1}]. \tag{3}$$

This design significantly improves the model's ability to associate targets and reduces ID switches, as demonstrated in our ablation experiments.

Position Predictor

After the introduction of the DETR-like model, researchers note that queries uniformly combine content and positional features of the target. DAB-DETR (Liu et al. 2022) separates content features and positional information from the queries, providing explicit positional priors to the decoder, which facilitates faster target localization.

In MOT, however, targets often move, making it inaccurate to use the position information from the previous frame as a prior for the next frame, especially for fast-moving targets. We use an LSTM network to design a simple and efficient Position Predictor that predicts the target's next frame positional information, which is combined with the content embedding to form the track query, aiding more accurate localization. During tracking, we store predicted target positions. When the storage length exceeds five frames, the Position Predictor can predict target motion trends. The positional sequence is input into the LSTM, which outputs h^t and c^t. Here, h^t predicts the next step, and c^t retains hidden information from previous sequences. h^t is used in the next frame to help detect targets, while c^t and the positional information of objects in the frame f_t are fed into the LSTM to continue predicting. This structure eliminates the need to store all target positions continuously.

Experiments

Datasets and Metrics

Datasets We conduct experiments on the MOT Challenge (Milan et al. 2016; Dendorfer et al. 2020) and DanceTrack datasets (Sun et al. 2022). The MOT17 benchmark (Milan et al. 2016) includes 7 training sequences and 7 testing sequences, while the MOT20 benchmark (Dendorfer et al. 2020) has 4 training and 4 testing sequences with more complex and crowded scenes. Compared to the MOT Challenge, DanceTrack features individuals with similar appearances and more complex movements.

Metrics We use CLEAR MOT Metrics (Bernardin and Stiefelhagen 2008) as evaluation standards. CLEAR MOT Metrics include Multiple-Object Tracking Accuracy(MOTA), IDF1 Score(IDF1), Identity Switches(IDSW), Mostly Tracked Trajectories(MT), and Mostly Lost Trajectories(ML). We also list the HOTA(Luiten et al. 2021), AssA, and DetA in our experimental results.

Implementation Details

TGFormer builds on MeMOTR (Gao and Wang 2023) with ResNet50 as the backbone and DAB-Deformable-DETR pretrained on COCO as the detector. Training uses 4 NVIDIA A800 GPUs with a batch size of 1 per GPU, each batch containing a video clip with multiple frames. The AdamW optimizer with a 2.0×10^{-4} learning rate is applied. Targets with scores below $\tau_{update} = 0.5$ or IoU below $\tau_{iou} = 0.5$ are filtered.

Track Query Group In TGFormer, each target corresponds to a Track Query Group. When a detect query detects a newborn object, a Track Query Group is initialized, and the query is classified as high score embedding, mid score embedding, or low score embedding based on its confidence, with the other embeddings initially set to zero vectors. The detection confidence of the output embedding may change during tracking. When the occlusion level of target changes significantly, causing the confidence to fall within the threshold range of the other two embeddings, the output embedding is saved as the corresponding embedding. The confidence thresholds are set as $\tau_{high} = 0.85, \tau_{mid} = 0.7, \tau_{low} = 0.5$. During training, we calculate the loss for each type of query in the track group against the corresponding ground truth: classification loss, L1 loss, and GIoU loss. The query with the smallest loss is selected as the final tracking result. To minimize the impact of classification loss, different weights are assigned to each loss type where $\lambda_{L1} = 5, \lambda_{GIoU} = 5, \lambda_{class} = 1$. During inference, the embedding with the highest confidence is selected as the detection result.

Position Predictor Since we typically input only 2-5 frames during training, collecting just 1-4 frames of historical positional information makes prediction challenging. Therefore, we adopt a two-stage training process. First, we pre-train the Position Predictor by reading sequences of positional information (x, y, w, h) from the dataset, converting them to (cx, cy, w, h) format for the decoder, and applying normalization and sigmoid operations. The model is trained by inputting fixed-length sequences of positional data, predicting the next positions, and calculating L1 and GIoU losses. To enhance prediction capability, we not only predict the next frame's position but also use the predictions to forecast further into the future. Data augmentation methods are also applied to improve robustness.

Training Data In the **MOT17** dataset, the limited video frames often leads to overfitting during training. To address this, we incorporate the CrowdHuman (Shao et al. 2018) Validation set, adding 4K static images, and apply jitter to simulate target motion. For this combined dataset, we train for 130 epochs, reducing the learning rate tenfold at the 120th epoch. The number of clip frames increases from the original 2 frames to 3, 4, 5, and 6 frames at the 50th, 70th, 90th, and 120th epochs, respectively. For the **MOT20** dataset, a two-stage training is employed. In the first stage, we pre-train for 60 epochs on the joint dataset, increasing the number of clip frames at the 15th, 30th, and 45th epochs. In the second stage, we fine-tune on MOT20 with the same settings, increasing the probability of FP and FN occurrences to 0.5 and 0.2 to prevent overfitting. On **DanceTrack** dataset, we train for 18 epochs, increasing the number of clip frames to 3, 4, and 5 at the 6th, 10th, and 14th epochs, respectively.

Comparison with State-of-the-art Methods

MOT Challenge We compare the performance of our model with state-of-the-art transformer-based methods on the MOT17(Milan et al. 2016) and MOT20(Dendorfer et al. 2020) test sets. Due to the additional detectors, CNN-based trackers generally outperform transformer-based trackers. In transformer-based trackers, Tab. 1 shows that TrackFormer(Meinhardt et al. 2022), MOTR(Zeng et al. 2022) and TransTrack(Sun et al. 2021) exhibit imbalanced performance between detection and tracking, with MOTA significantly higher than IDF1. MeMOT(Cai et al. 2022) and MeMOTR(Gao and Wang 2023) leverage temporal information in trajectories, achieving higher IDF1 scores. Building on MeMOTR(Gao and Wang 2023), we design the Track Query Group, further improving HOTA and IDF1, reducing ID-SW, and significantly increasing MOTA. Thus, TGFormer outperforms these transformer-based models in multiple met-

Methods	MOTA↑	IDF1↑	MT↑	ML↓	IDS↓	HOTA↑	DetA↑	AssA↑
CNN-based								
FairMOT (Zhang et al. 2021)	73.7	72.3	43.2	17.3	3303	59.3	60.9	58.0
TraDeS (Wu et al. 2021)	69.1	63.9	36.4	21.5	3555	52.7	55.2	50.8
ByteTrack (Zhang et al. 2022)	80.3	77.3	-	-	2196	63.1	64.5	62.0
GeneralTrack (Qin et al. 2024)	80.6	78.3	-	-	1563	64.0	65.1	63.1
Transformer-based								
TrackFormer (Meinhardt et al. 2022)	74.1	68.0	47.3	10.4	2829	57.3	60.9	54.1
MOTR (Zeng et al. 2022)	73.4	68.6	-	-	2439	57.8	55.7	60.3
TransTrack (Sun et al. 2021)	74.5	63.9	46.8	11.3	3663	54.1	61.6	47.9
TransCenter (Xu et al. 2022)	73.2	62.2	40.8	18.5	3663	54.5	-	49.7
MeMOT (Cai et al. 2022)	72.5	69.0	43.8	18.0	2724	56.9	-	55.2
MeMOTR (Gao and Wang 2023)	72.8	71.5	41.4	19.2	1902	58.8	59.6	58.4
TGFormer(ours)	**74.9**	**72.0**	43.7	18.7	2629	**60.3**	61.4	59.3
Hybrid-based								
MOTRv2 (Zhang, Wang, and Zhang 2023)	78.6	75.0	-	-	-	62.0	63.8	60.6

Table 1: Comparison of private detection results on the MOT17 (Milan et al. 2016) test set. Due to the excellent performance of existing detectors, CNN-based trackers achieve high performance. On the other hand, when compared to similar transformer-based trackers, our method demonstrates more competitive performance. MOTRv2 is categorized as a Hybrid-based approach due to its additional object detector, YOLOX.

Methods	MOTA↑	IDF1↑	HOTA↑
TransTrack (Sun et al. 2021)	64.5	59.2	-
TransCenter (Xu et al. 2022)	67.7	58.7	-
MeMOT (Cai et al. 2022)	63.7	66.1	54.1
TGFormer(ours)	**70.3**	**67.1**	**54.2**

Table 2: Comparison of private detection results on the MOT20 (Dendorfer et al. 2020) test set. Our method achieves outstanding performance on this dataset without extra training data, particularly excelling in the MOTA metric.

rics. MOTRv2(Zhang, Wang, and Zhang 2023), categorized as a Hybrid-based approach, integrates a transformer-based tracker with an additional object detector, YOLOX, to deliver superior tracking performance. However, this design requires an extra detection network, which hinders end-to-end training.

DanceTrack The motion states of targets in DanceTrack(Sun et al. 2022) are more complex, which better assesses the model's tracking capabilities. We test our model on this dataset, as shown in Tab. 3. Our results demonstrate superior detection and tracking performance compared to MeMOTR and other models. This indicates that our method has been effectively validated across multiple datasets and different challenge types.

Ablation Study

In this section, we discuss in detail the effectiveness of several designed modules, such as the Track Query Group Updater and Position Updater, and their contribution to improving the model's tracking capabilities. In terms of experimental setup, to efficiently test the impact of these modules, we split the sequences in the MOT17 training set into two halves, using one half as the training set and the other half as the validation set. We use MOTA, IDF1, and HOTA as the primary metrics to measure performance. For all ablation experiments, we take the average of the results from three trials as the final outcome.

Track Query Group Updater In the Track Query Group Updater, there are three main modules: Short-Term Memory Aggregator, Occlusion-Aware Memory-Attention, and long-term memory. To explore the interactions among these three modules, we conduct ablation experiments. After obtaining the output embedding O^t, the model updates the track query in the group. First, O^t and TG^t are concatenated, and then passed through the Short-Term Memory Aggregator. The primary purpose is to extract and fuse target features in multiple frames.

We first test the baseline model, MeMOTR, on the validation set without introducing the Track Query Group. Then, we introduce the Track Query Group and simply apply the Short-Term Memory Aggregator to update the queries. Tab. 4 shows improvements over the baseline, with IDF1 (+0.1%) and HOTA (+3.3%) both increasing. The most significant improvement is in MOTA (+4.7%), indicating that the introduction of the Track Query Group effectively enhances the model's detection capabilities.

Next, we incorporate Occlusion-Aware Memory-Attention into the Track Query Group Updater to extract and differentiate the unique features of the target under different occlusion levels. After obtaining three differentiated features from the output of Occlusion-Aware Memory-Attention, we also need to compensate for the missing general features of the target. Therefore, we add residual connections from the long-term memory to compensate for the missing general

Methods	MOTA↑	IDF1↑	HOTA↑	DetA↑	AssA↑
CNN-based					
CenterTrack (Zhou, Koltun, and Krähenbühl 2020)	86.8	35.7	41.8	78.1	22.6
GeneralTrack (Qin et al. 2024)	91.8	59.7	59.2	82.0	42.8
Transformer-based					
TransTrack (Sun et al. 2021)	88.4	45.2	45.5	75.9	27.5
MOTR (Zeng et al. 2022)	79.7	51.5	54.2	73.5	40.2
MeMOTR (Gao and Wang 2023)	89.9	71.2	68.5	80.5	**58.4**
TGFormer(ours)	**91.3**	**71.9**	**69.1**	**81.9**	58.3
Hybrid-based					
MOTRv2 (Zhang, Wang, and Zhang 2023)	91.9	71.7	69.9	83.0	59.0

Table 3: Comparison of private detection results on the DanceTrack (Sun et al. 2022) test set. Compared to transformer-based trackers, our method nearly leads in all performance metrics.

Aggregator	Attn + L^t	MOTA↑	IDF1↑	HOTA↑
Baseline		67.4	66.4	54.6
✓		72.1	66.5	57.9
✓	✓	74.3	67.9	58.3

Table 4: Ablation study on modules within the Track Query Group Updater. Baseline means MeMOTR (Gao and Wang 2023) without Track Query Group. Aggregator means Short-Term Memory Aggregator. Attn + L^t means Occlusion-Aware Memory-Attention with long-term memory.

N_{input}	N_{pred}	MOTA↑	IDF1↑	HOTA↑
3	1	74.1	66.1	57.8
5	2	74.3	67.8	58.1
10	3	74.9	68.3	58.4

Table 5: Ablation study on training Position Predictor with different input frame N_{input} and prediction frame N_{pred}.

features in the queries. This integration of general and differentiated features helps in distinguishing between different targets without losing tracking capabilities. Tab. 4 shows that after incorporating long-term memory, MOTA (+2.2%), IDF1 (+1.4%), and HOTA (+0.4%) are further improved, demonstrating the effectiveness of the Track Query Updater in significantly enhancing the model's tracking capabilities.

Position Predictor In traditional multi-object tracking, Kalman filters predict target motion trajectories, providing precise localization priors to detectors. However, this approach has not been adopted in transformer-based models. We introduce a simple LSTM for position prediction, effective in time-series tasks. During training, we explore various input and prediction frame counts to optimize the Position Predictor. As shown in Table 5, the best results are achieved with 10 input frames and 3 prediction frames, improving MOTA (+0.6%), IDF1 (+0.4%), and HOTA (+0.1%).

We also visualize Position Predictor results on MOT17 sequences (Fig. 4). For fast-moving targets or in scenarios with constant camera motion, the position of individuals relative to the previous frame has significantly shifted. The Position Predictor provides the targets' location in the next frame as a prior to help the model track more effectively. Additionally, it performs equally well for slow or stationary targets. This undoubtedly aids in the continuous tracking of targets.

Figure 4: **Visualization of Target Position Prediction**. The true bounding box of the target in the previous frame is shown in gray, while the current frame's true bounding box is in color. The predicted bounding box for the current frame, generated by the Position Predictor, is shown in black.

Conclusion

This paper introduces TGFormer, a model that incorporates Track Query Group into multi-object tracking. We abandon the traditional idea of associating a target with a single query and instead generate a group of queries based on the target's different occlusion levels in the scene. Additionally, we design an updater for the Track Query Group to update queries at different occlusion levels. Finally, we integrate a Position Predictor into the transformer-based model to better utilize the target's temporal positional information. Extensive experiments demonstrate the effectiveness of our approach.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61772061 and by the High-Performance Computing Platform of BUPT.

References

Aharon, N.; Orfaig, R.; and Bobrovsky, B.-Z. 2022. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. arXiv:2206.14651.

Bernardin, K.; and Stiefelhagen, R. 2008. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008: 1–10.

Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; and Upcroft, B. 2016. Simple online and realtime tracking. In *Proceedings of the IEEE/CVF International Conference on Image Processing*, 3464–3468.

Cai, J.; Xu, M.; Li, W.; Xiong, Y.; Xia, W.; Tu, Z.; and Soatto, S. 2022. Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8090–8100.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, 213–229.

Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.; Roth, S.; Schindler, K.; and Leal-Taixé, L. 2020. MOT20: A benchmark for multi object tracking in crowded scenes. arXiv:2003.09003.

Gao, R.; and Wang, L. 2023. Memotr: Long-term memory-augmented transformer for multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9901–9910.

Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. YOLOX: Exceeding YOLO Series in 2021. arXiv:2107.08430.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Hosang, J.; Benenson, R.; and Schiele, B. 2017. Learning non-maximum suppression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4507–4515.

Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhu, J.; and Zhang, L. 2022. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. arXiv:2201.12329.

Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.; Geiger, A.; Leal-Taixé, L.; and Leibe, B. 2021. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129: 548–578.

Maggiolino, G.; Ahmad, A.; Cao, J.; and Kitani, K. 2023. Deep OC-Sort: Multi-Pedestrian Tracking by Adaptive Re-Identification. In *Proceedings of the IEEE International Conference on Image Processing*, 3025–3029.

Meinhardt, T.; Kirillov, A.; Leal-Taixe, L.; and Feichtenhofer, C. 2022. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8844–8854.

Milan, A.; Leal-Taixe, L.; Reid, I.; Roth, S.; and Schindler, K. 2016. MOT16: A Benchmark for Multi-Object Tracking. arXiv:1603.00831.

Qin, Z.; Wang, L.; Zhou, S.; Fu, P.; Hua, G.; and Tang, W. 2024. Towards Generalizable Multi-Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18995–19004.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Ristani, E.; and Tomasi, C. 2018. Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6036–6046.

Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; and Sun, J. 2018. CrowdHuman: A Benchmark for Detecting Human in a Crowd. arXiv:1805.00123.

Sun, P.; Cao, J.; Jiang, Y.; Yuan, Z.; Bai, S.; Kitani, K.; and Luo, P. 2022. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20993–21002.

Sun, P.; Cao, J.; Jiang, Y.; Zhang, R.; Xie, E.; Yuan, Z.; Wang, C.; and Luo, P. 2021. TransTrack: Multiple Object Tracking with Transformer. arXiv:2012.15460.

Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; and Wang, S. 2020. Towards real-time multi-object tracking. In *Proceedings of the European Conference on Computer Vision*, 107–122.

Welch, G.; and Bishop, G. 1995. *An Introduction to the Kalman Filter*. An Introduction to the Kalman Filter.

Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3645–3649.

Wu, J.; Cao, J.; Song, L.; Wang, Y.; Yang, M.; and Yuan, J. 2021. Track to detect and segment: An online multi-object tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12352–12361.

Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; and Alameda-Pineda, X. 2022. TransCenter: Transformers with dense representations for multiple-object tracking. *IEEE transactions on pattern analysis and machine intelligence*, 45(6): 7820–7835.

Yan, F.; Luo, W.; Zhong, Y.; Gan, Y.; and Ma, L. 2023. Bridging the Gap Between End-to-end and Non-End-to-end Multi-Object Tracking. arXiv:2305.12724.

Yang, F.; Odashima, S.; Masui, S.; and Jiang, S. 2023. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4799–4808.

Yu, E.; Wang, T.; Li, Z.; Zhang, Y.; Zhang, X.; and Tao, W. 2023. MOTRv3: Release-Fetch Supervision for End-to-End Multi-Object Tracking. arXiv:2305.14298.

Zeng, F.; Dong, B.; Zhang, Y.; Wang, T.; Zhang, X.; and Wei, Y. 2022. Motr: End-to-end multiple-object tracking with transformer. In *Proceedings of the European Conference on Computer Vision*, 659–675.

Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; and Wang, X. 2022. Bytetrack: Multi-object tracking by associating every detection box. In *Proceedings of the European Conference on Computer Vision*, 1–21.

Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; and Liu, W. 2021. FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking. *International Journal of Computer Vision*, 129(11): 3069–3087.

Zhang, Y.; Wang, T.; and Zhang, X. 2023. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22056–22065.

Zhou, X.; Koltun, V.; and Krähenbühl, P. 2020. Tracking objects as points. In *Proceedings of the European Conference on Computer Vision*, 474–490.

Zhu, T.; Hiller, M.; Ehsanpour, M.; Ma, R.; Drummond, T.; Reid, I.; and Rezatofighi, H. 2023. Looking Beyond Two Frames: End-to-End Multi-Object Tracking Using Spatial and Temporal Transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11): 12783–12797.