MOTRv2: Bootstrapping End-to-End Multi-Object Tracking by Pretrained Object Detectors

Yuang Zhang[1*], Tiancai Wang[2], Xiangyu Zhang[2,3]
[1]Shanghai Jiao Tong University [2]MEGVII Technology [3]Beijing Academy of Artificial Intelligence

Abstract

*In this paper, we propose MOTRv2, a simple yet effective pipeline to bootstrap end-to-end multi-object tracking with a pretrained object detector. Existing end-to-end methods, e.g. MOTR [43] and TrackFormer [20] are inferior to their tracking-by-detection counterparts mainly due to their poor detection performance. We aim to improve MOTR by elegantly incorporating an extra object detector. We first adopt the anchor formulation of queries and then use an extra object detector to generate proposals as anchors, providing detection prior to MOTR. The simple modification greatly eases the conflict between joint learning detection and association tasks in MOTR. MOTRv2 keeps the query propogation feature and scales well on large-scale benchmarks. MOTRv2 ranks the 1st place (**73.4%** HOTA on DanceTrack) in the 1st Multiple People Tracking in Group Dance Challenge. Moreover, MOTRv2 reaches state-of-the-art performance on the BDD100K dataset. We hope this simple and effective pipeline can provide some new insights to the end-to-end MOT community. Code is available at https://github.com/megvii-research/MOTRv2.*

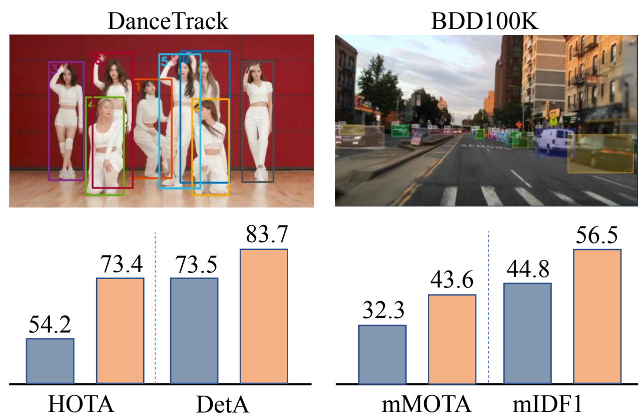


Figure 1. Performance comparison between MOTR (grey bar) and MOTRv2 (orange bar) on the DanceTrack and BDD100K datasets. MOTRv2 improves the performance of MOTR by a large margin under different scenarios.

1. Introduction

Multi-object tracking (MOT) aims to predict the trajectories of all objects in the streaming video. It can be divided into two parts: detection and association. For a long time, the state-of-the-art performance on MOT has been dominated by tracking-by-detection methods [4, 36, 44, 45] with good detection performance to cope with various appearance distributions. These trackers [44] first employ an object detector (e.g., YOLOX [11]) to localize the objects in each frame and associate the tracks by ReID features or IoU matching. The superior performance of those methods partially results from the dataset and metrics biased towards detection performance. However, as revealed by the DanceTrack dataset [27], their association strategy remains to be improved in complex motion.

Recently, MOTR [43], a fully end-to-end framework is introduced for MOT. The association process is performed by updating the tracking queries while the new-born objects are detected by the detect queries. Its association performance on DanceTrack is impressive while the detection results are inferior to those tracking-by-detection methods, especially on the MOT17 dataset. We attribute the inferior detection performance to the conflict between the joint detection and association processes. Since state-of-the-art trackers [6, 9, 44] tend to employ extra object detectors, one natural question is how to incorporate MOTR with an extra object detector for better detection performance. One direct way is to perform IoU matching between the predictions of track queries and extra object detector (similar to TransTrack [28]). In our practice, it only brings marginal improvements in object detection while disobeying the end-to-end feature of MOTR.

Inspired by tracking-by-detection methods that take the detection result as the input, we wonder if it is possible to feed the detection result as the input and reduce the learning of MOTR to the association. Recently, there are some ad-

* The work was done during internship at MEGVII Technology and supported by National Key R&D Program of China (2020AAA0105200) and Beijing Academy of Artificial Intelligence (BAAI).

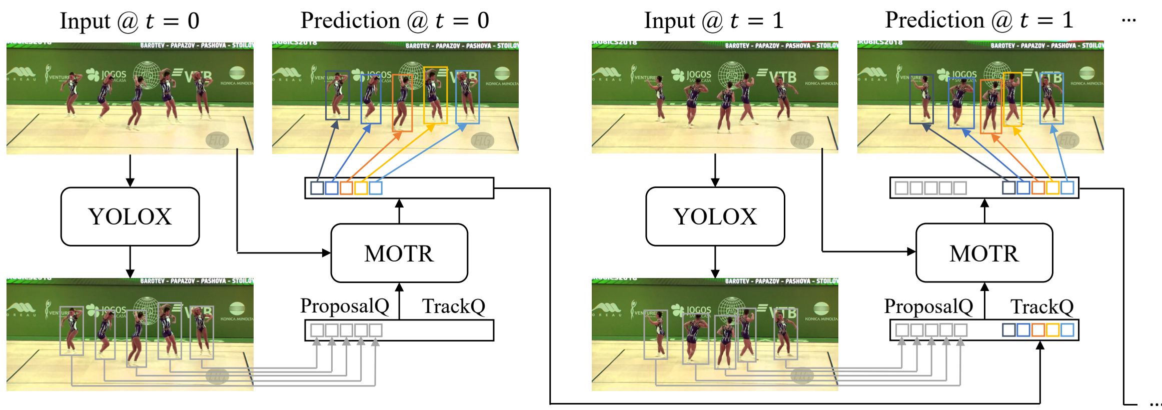


Figure 2. The overall architecture of MOTRv2. The proposals produced by state-of-the-art detector YOLOX [11] are used to generate the proposal queries, which replaces the detect queries in MOTR [43] for detecting new-born objects. The track queries are transferred from previous frame and used to predict the bounding boxes for tracked objects. The concatenation of proposal queries and track queries as well as the image features are input to MOTR to generate the predictions frame-by-frame.

vances [18, 35] for anchor-based modeling in DETR. For example, DAB-DETR initializes object queries with the center points, height, and width of anchor boxes. Similar to them, we modify the initialization of both detect and track queries in MOTR. We replace the learnable positional embedding (PE) of detect query in MOTR with the sine-cosine PE [30] of anchors, producing an anchor-based MOTR tracker. With such anchor-based modeling, proposals generated by an extra object detector can serve as the anchor initialization of MOTR, providing local priors. The transformer decoder is used to predict the relative offsets w.r.t. the anchors, making the optimization of the detection task much easier.

The proposed MOTRv2 brings many advantages compared to the original MOTR. It greatly benefits from the good detection performance introduced by the extra object detector. The detection task is implicitly decoupled from the MOTR framework, easing the conflict between the detection and association tasks in the shared transformer decoder. MOTRv2 learns to track the instances across frames given the detection results from an extra detector.

MOTRv2 achieves large performance improvements on the DanceTrack, BDD100K, and MOT17 datasets compared to the original MOTR (see Fig. 1). On the DanceTrack dataset, MOTRv2 surpasses the tracking-by-detection counterparts by a large margin (**14.8%** HOTA compared to OC-SORT [6]), and the AssA metric is **18.8%** higher than the second-best method. On the large-scale multi-class BDD100K dataset [42], we achieved 43.6% mMOTA, which is 2.4% better than the previous best solution Unicorn [41]. MOTRv2 also achieves state-of-the-art performance on the MOT17 dataset [15, 21]. We hope our simple and elegant design can serve as a strong baseline for future end-to-end multi-object tracking research.

2. Related Works

Tracking by Detection Predominant approaches [6, 44] mainly follow the tracking-by-detection pipeline: an object detector first predicts the object bounding boxes for each frame, and a separate algorithm is then used to associate the instance bounding boxes across adjacent frames. The performance of these methods greatly depends on the quality of object detection.

There are various attempts using the Hungarian algorithm [14] for association: SORT [4] applies a Kalman filter [37] for each tracked instance and uses the intersection-over-union (IoU) matrix among the predicted boxes of Kalman filter and the detected boxes for matching. DeepSORT [38] introduces a separate network to extract the appearance features of the instances and uses the pairwise cosine distances on top of SORT. JDE [36], TrackRCNN [25], FairMOT [45], and Unicorn [41] further explores the joint training of object detection and appearance embedding. ByteTrack [44] leverages a powerful YOLOX-based [11] detector and achieves state-of-the-art performance. It introduces an enhanced SORT algorithm to associate the low score detection boxes as well instead of only associating the high score ones. BoT-SORT [1] further designs a better Kalman filter state, camera-motion compensation, and ReID feature fusion. TransMOT [9] and GTR [48] employ spatial-temporal transformers for instance feature interaction and historical information aggregation when calculating the assignment matrix. OC-SORT [6] relaxes the linear motion assumption and uses a learnable motion model.

While our approach also benefits from a robust detector, we do not compute similarity matrices but use track queries with anchors to jointly model motion and appearance.

Tracking by Query Propagation Another paradigm of MOT extends query-based object detectors [7, 29, 49] to tracking. These methods force each query to recall the same instance across different frames. The interaction between the query and image feature can be performed in parallel or serially in time.

The *parallel* methods take a short video as input and use a set of queries to interact with all frames to predict the trajectories of instances. VisTR [34] and subsequent works [8, 40] extend DETR [7] to detect tracklets in short video clips. Parallel methods need to take the entire video as input, so they are memory-consuming and limited to short video clips of a few dozen frames.

The *serial* methods perform frame-by-frame query interaction with image features and iteratively update the track queries associated with the instances. Tracktor++ [2] utilizes the R-CNN [12] regression head for iterative instance re-localization across frames. TrackFormer [20] and MOTR [43] extend from the Deformable DETR [49]. They predict the object bounding boxes and update the tracking query for detecting the same instances in subsequent frames. MeMOT [5] builds the short-term and long-term instance feature memory banks to generate the track queries. TransTrack [28] propagates track queries once to find the object location in the following frame. P3AFormer [46] adopts flow-guided image feature propagation. Unlike MOTR, TransTrack and P3AFormer still use location-based Hungarian matching in historical tracks and current detections, rather than propagating queries throughout the video.

Our approach inherits the query propagation method for long-term end-to-end tracking, while also utilizing a powerful object detector to provide object location prior. The proposed method greatly outperforms the existing matching and query-based methods in terms of tracking performance in complex motions.

3. Method

Here, we present MOTRv2 based on proposal query generation (Sec. 3.4) and proposal propagation (Sec. 3.5).

3.1. Revisiting MOTR

MOTR [43] is a fully end-to-end multiple-object tracking framework built upon the Deformable DETR [49] architecture. It introduces the track query and object query. The object query is responsible for detecting new-born or missed objects, while each track query is responsible for tracking a unique instance over time. To initialize track queries, MOTR uses the output of the object query associated with newly detected objects. Track queries are updated by their state and current image features over time, which allows them to predict tracks in an online manner.

The tracklet-aware label assignment in MOTR assigns track queries to their previously tracked instances while assigning object queries to the remaining instances by bipartite matching. MOTR introduces a temporal aggregation network to enhance the features of track queries, and a collective average loss to balance the loss across frames.

3.2. Motivation

One major limitation of the end-to-end multiple-object tracking frameworks is their poor detection performance, compared to tracking-by-detection approaches [6, 44] that rely on standalone object detectors. To address this limitation, we propose to incorporate the YOLOX [11] object detector to generate proposals as object anchors, providing detection prior to MOTR. It greatly eases the conflict between joint learning detection and association tasks in MOTR, improving the detection performance.

3.3. Overall Architecture

As shown in Figure 2, the proposed MOTRv2 architecture consists of two main components: a state-of-the-art object detector and a modified anchor-based MOTR tracker.

The object detector component first generates proposals for both training and inference. For each frame, YOLOX generates a set of proposals that include center coordinates, width, height, and confidence values. The modified anchor-based MOTR component is responsible for learning track association based on the generated proposals. Sec. 3.4 describes the replacement of the detect queries in the original MOTR framework with proposal queries. The modified MOTR now takes the concatenation of the track query and proposal query as input. Sec. 3.5 describes the interaction between concatenated queries and frame features to update the bounding boxes of tracked objects.

3.4. Proposal Query Generation

In this section, we explain how the proposal query generation module provides MOTR with high-quality proposals from YOLOX. The input to this module is a set of proposal boxes generated by YOLOX for each frame in the video. Unlike DETR [7] and MOTR, which use a fixed number of learnable queries for object detection, our framework dynamically determines the number of proposal queries based on the selected proposals generated by YOLOX.

Specifically, for frame t, YOLOX generates N_t proposals, each represented by a bounding box with center coordinates (x_t, y_t), height h_t, width w_t, and confidence score s_t. As illustrated in the orange part of Figure 3, we introduce a shared query q_s to generate a set of proposal queries. The shared query, which is a learnable embedding of size $1 \times D$, is first broadcasted to the size of $N_t \times D$. The predicted

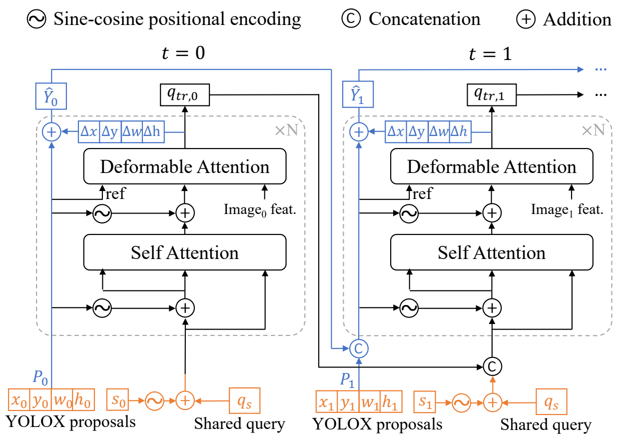


Figure 3. Proposal query generation and proposal propagation for Tracking. The orange color marks proposal query generation while the blue color marks the proposal propagation path; the dashed gray box stands for N transformer decoders. The query interaction module in MOTR is omitted for simplicity.

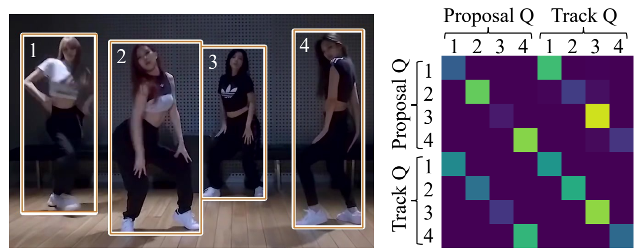
scores s_t of N_t proposal boxes produce the score embeddings of size $N_t \times D$ by sine-cosine positional encoding. The broadcasted queries are then added with the score embeddings to generate the proposal queries. The YOLOX proposal boxes serve as the anchors of the proposal queries. In practice, we also use 10 learnable anchors (similar to DAB-DETR [18]) and concatenate them with YOLOX proposals to recall objects missed by the YOLOX detector.

3.5. Proposal Propagation

In MOTR [43], track query and detect query are concatenated and input to the transformer decoder for simultaneous object detection and track association. The track queries generated from the previous frame represents the tracked objects, which are used to predict the bounding boxes of current frame. The detect queries is a fixed set of learnable embeddings and used to detect the new-born objects. Different from MOTR, our method uses the proposal query for detecting new-born objects and the prediction of track queries are made based on previous frame prediction.

For the first frame ($t = 0$), there are only new-born objects, which are detected by the YOLOX. As mentioned above, the proposal queries are generated given the shared query q_s and predicted scores of YOLOX proposals. After the positional encoding by YOLOX proposals P_0, the proposal queries are further updated by the self-attention and interact with the image features by deformable attention to produce the track queries $q_{tr,0}$ and the relative offsets $(\Delta x, \Delta y, \Delta w, \Delta h)$ w.r.t. to the YOLOX proposals P_0. The prediction $\hat{Y}_0$ is the sum of the proposals P_0 and the predicted offsets.

For other frames ($t > 0$), similar to MOTR, track queries



(a) Proposal and track query prediction (b) Query attention map

Figure 4. Visualization of (a) MOTR track query box prediction (brown boxes) highly overlaps the YOLOX proposals (while boxes in bold) on the 100th frame of sequence "dancetrack0005", and (b) the query self-attention map shows a clear exchange of information between the proposal query and the corresponding tracking query of the same instance.

$q_{tr,t-1}$ generated from the previous frame will be concatenated with the proposal queries $q_{p,t}$ of the current frame. The box predictions $\hat{Y}_{t-1}$ of the previous frame will also be concatenated together with the YOLOX proposals P_t to serve as the anchors for the current frame. The sine-cosine encoding of the anchors is used as the positional embedding for the concatenated queries, which then go to the transformer decoder to produce the prediction and updated track queries. The bounding box prediction consists of confidence scores and relative offsets w.r.t. anchors, and the updated track queries $q_{tr,t}$ are further transferred to the next frame for detecting tracked objects.

Analysis In the above design, the proposals queries are constrained to detect only the new-born or missing objects and the track queries are responsible for relocating the tracked objects. The proposal queries need to aggregate information from track queries to avoid duplicated detection of tracked objects, and track queries can utilize YOLOX proposals to improve object localization. This is accomplished by the self-attention layers in the transformer decoder. To better understand the interaction between proposal queries and track queries, we visualize the query self-attention map in Figure 4. For the same instance, the proposal query and the corresponding track query have high similarity, and there is a clear exchange of information between them, which verifies our assumptions.

4. Experiments

4.1. Datasets and Metrics

Datasets We use the DanceTrack [27], MOT17 [15, 21] and BDD100K [42] datasets to evaluate our approach.

DanceTrack [27] is a large-scale dataset for multi-human tracking in dancing scenes. It features a uniform appearance and diverse motion which is challenging for associating instances across frames. DanceTrack includes 100 videos: 40 for training, 25 for validation, and 35 for testing. The average length of videos is 52.9s.

MOT17 [15,21] is a widely used dataset containing 7 sequences for training and 7 for testing. It mainly contains relatively crowded street scenes with simple and linear movement of pedestrians.

BDD100K [42] is a dataset of autonomous driving scenarios. It contains a multi-object tracking subset with 1400 sequences for training and 200 sequences for validation. The sequence length is about 40 seconds and the number of object classes is 8. We use it to test the multi-class multi-object tracking performance.

Metrics We use the higher order metric for multi-object tracking (Higher Order Tracking Accuracy; HOTA) [19] to evaluate our method and analysis the contribution decomposed into detection accuracy (DetA) and association accuracy (AssA). For MOT17 and BDD100K datasets, we list the MOTA [3] and IDF1 [23] metrics.

4.2. Implementation Details

Proposal Generation We use the YOLOX [11] detector with weights provided by ByteTrack [44] and DanceTrack [27] to generate object proposals. The hyperparameters, such as the input image size, are consistent with ByteTrack. To maximize the proposal recall, we keep all YOLOX predicted boxes with confidence scores over 0.05 as proposals. For DanceTrack [27], we use the YOLOX weights from the DanceTrack official GitHub repository[1]. For CrowdHuman [24] and MOT17, we use the public weight for the MOT17 test set from ByteTrack [44]. We do not train YOLOX on these two datasets and only use it to generate proposals for all images before training MOTR. For BDD100K [42], we use its MOT set together with the 100k images set for training. The YOLOX detector is trained on 8 Tesla V100 GPUs for 16 epochs. We follow ByteTrack [44] for other training hyperparameters.

MOTR Our implementation is based on the official repo[2] with a ResNet50 [13] backbone for feature extraction. All MOTR models are trained on 8 GPUs, with a batch size of 1 per GPU. For DanceTrack [27], we follow YOLOX [11] and adopt the HSV augmentation for training MOTR. As opposed to the original implementation of propagating track queries that match ground truth tracks during training, we propagate track queries with a confidence score above 0.5, which naturally creates false positive (FP; high score but no instances, *e.g.* lost tracks) and false negative (FN; instances not detected) track queries to enhance the handling of FPs and FNs during inference. In this way, we do not follow MOTR to manually insert negative or drop positive track queries, *i.e.*, $p_{drop} = 0$ and $p_{insert} = 0$. We train the ablation study and state-of-the-art comparison models for 5 epochs with a fixed clip size of 5. The sampling stride of the frames inside a clip is randomly chosen from 1 to 10. The initial learning rate 2×10^{-4} is dropped by a factor of 10 at the 4^{th} epoch. For MOT17 [15, 21], the number of training epochs is tuned down to 50 and the learning rate drops at the 40^{th} epoch. For BDD100K [42], we train for 2.5 epochs using a clip size of 4 with a random sampling stride from 1 to 4. The learning rate drops at the end of the 2^{nd} epoch. For the extension to multi-class MOT, each YOLOX proposal additionally includes a class label and we use a different learnable embedding (shared query) for each class. Other settings are not changed.

[1]https://github.com/DanceTrack/DanceTrack
[2]https://github.com/megvii-research/MOTR

Table 1. Performance comparison with state-of-the-art methods on the DanceTrack [27] test set. Results for existing methods are from DanceTrack [27]. MOTRv2* denotes MOTRv2 with an extra association, adding validation set for training, and test ensemble.

Methods	HOTA	DetA	AssA	MOTA	IDF1
FairMOT [45]	39.7	66.7	23.8	82.2	40.8
CenterTrack [47]	41.8	78.1	22.6	86.8	35.7
TransTrack [28]	45.5	75.9	27.5	88.4	45.2
TraDes [39]	43.3	74.5	25.4	86.2	41.2
ByteTrack [44]	47.7	71.0	32.1	89.6	53.9
GTR [39]	48.0	72.5	31.9	84.7	50.3
QDTrack [22]	54.2	80.1	36.8	87.7	50.4
MOTR [43]	54.2	73.5	40.2	79.7	51.5
OC-SORT [6]	55.1	80.3	38.3	92.0	54.6
MOTRv2 (ours)	69.9	83.0	59.0	91.9	71.7
MOTRv2* (ours)	**73.4**	**83.7**	**64.4**	**92.1**	**76.0**

Joint Training with CrowdHuman To improve the detection performance, we also utilize a large number of static CrowdHuman images. For the DanceTrack dataset, similar to the joint training of MOT17 and CrowdHuman in MOTR, we generate pseudo-video clips for CrowdHuman and perform joint training with DanceTrack. The length of the pseudo-video clips is also set to 5. We use the 41,796 samples from the training set of the DanceTrack [27] dataset and the 19,370 samples from the training and validation set of the CrowdHuman [24] dataset for joint training. For the MOT17 Dataset, we keep the original settings in MOTR that concatenate the validation set of CrowdHuman and the training set of MOT17.

4.3. State-of-the-art Comparison on DanceTrack

We compare MOTRv2 with the state-of-the-art methods on the DanceTrack [27] test set and the results are shown in Table 1. Without bells and whistles, our method achieves 69.9 HOTA and shows the best performance on all

Table 2. Performance comparison with state-of-the-art methods on the BDD100K [42] MOT validation set. MOTR* means MOTRv2 without using YOLOX proposals.

Methods	mMOTA	mIDF1	MOTA	IDF1
Yu *et al*. [42]	25.9	44.5	56.9	66.8
QDTrack [22]	36.6	50.8	63.5	71.5
TETer [17]	39.1	53.3	/	/
Unicorn [41]	41.2	54.0	**66.6**	71.3
MOTR [43]	32.3	44.8	56.2	65.8
MOTR*	35.5	48.2	59.6	68.9
MOTRv2 (ours)	**43.6**	**56.5**	65.6	**72.7**

higher-order metrics, surpassing other state-of-the-art methods by a large margin. Compared to those matching-based methods, *e.g*. ByteTrack [44] and OC-SORT [6], our approach shows a much better association accuracy (59.0% *vs*. 38.3%) while also achieves decent detection accuracy (83.0% *vs*. 80.3%). MOTRv2 achieves 69.9% higher order tracking accuracy (HOTA), which is **14.8%** better than the previous best method. The large gap in the IDF1 metric between previous methods and MOTRv2 also shows the superiority of our method in complex motions. For better performance, we apply an extra association in post-processing: if only one track exits and another one appears within 20 to 100 frames, we consider them to be tracks of the same instance. With the extra association, adding the validation set for training, and using an ensemble of 4 models, we further achieve 73.4% HOTA on the DanceTrack test set.

4.4. State-of-the-art Comparison on BDD100K

Table 2 shows the results on the BDD100k [42] tracking validation set. MOTRv2 achieved the highest mMOTA and mIDF1 among all methods. For a fair comparison, we equip the MOTR with 100k image set joint training and box propagation, denoted as MOTR*. By utilizing YOLOX proposals, MOTRv2 outperforms MOTR* by 8.1% mMOTA and 8.3% mIDF1, showing that the YOLOX proposals greatly improve the multi-class detection and tracking performance. Compared to other state-of-the-art methods, MOTRv2 outperforms the best tracker Unicorn by 2.4% mMOTA and 1.1% mIDF1. The higher mMOTA and mIDF1 (averaged among all classes) indicate that MOTRv2 handles the multi-class scenarios better. The difference in overall MOTA (-1.0%) and IDF1 (+1.4%) shows that our method is better in terms of association.

4.5. Comparison on the MOTChallenge

We further compare the performance of MOTRv2 with the state-of-the-art methods on the MOT17 [15, 21] and MOT20 [10] datasets. Table 3 shows the comparison on MOT17. Compared to the original MOTR [43], the introduction of YOLOX proposals consistently improves the detection (DetA) and association (AssA) accuracy by 3.5% and 4.9% correspondingly. The proposed approach pushes the performance of query-based trackers in crowded scenarios to the state-of-the-art level. We attribute the remaining performance gap to the fact that the scale of the MOT17 dataset is too small (215 seconds in total), which is insufficient to train a query-based tracker. Table 4 shows our result on the MOT20 [10] dataset. The performance gap between our method and ByteTrack [44] can be reduced with the joint training of MOT17, especially for the AssA metric. It also suggests that the lower performance in the MOT challenge is more likely due to the small size of real videos.

Table 3. Comparison to existing methods on the MOT17 dataset.

Methods	HOTA	AssA	DetA	IDF1	MOTA
CNN-based:					
Tracktor++ [2]	44.8	45.1	44.9	52.3	53.5
CenterTrack [47]	52.2	51.0	53.8	64.7	67.8
TraDeS [39]	52.7	50.8	55.2	63.9	69.1
QuasiDense [22]	53.9	52.7	55.6	66.3	68.7
GSDT [33]	55.5	54.8	56.4	68.7	66.2
FairMOT [45]	59.3	58.0	60.9	72.3	73.7
CorrTracker [31]	60.7	58.9	62.9	73.6	76.5
Unicorn [41]	61.7	/	/	75.5	77.2
GRTU [32]	62.0	62.1	62.1	75.0	74.9
MAATrack [26]	62.0	60.2	64.2	75.9	79.4
ByteTrack [44]	63.1	62.0	64.5	77.3	80.3
OC-SORT [6]	63.2	63.2	/	77.5	78.0
BoT-SORT [1]	64.6	/	/	79.5	80.6
Transformer-based:					
TrackFormer [20]	/	/	/	63.9	65.0
TransTrack [28]	54.1	47.9	61.6	63.9	74.5
MOTR [43]	57.8	55.7	60.3	68.6	73.4
P3AFormer [46]	/	/	/	78.1	81.2
MOTRv2 (ours)	62.0	60.6	63.8	75.0	78.6

Table 4. Comparison to existing methods on the MOT20 test set.

Methods	HOTA	AssA	DetA	IDF1	MOTA
FairMOT [45]	54.6	54.7	54.7	67.3	61.8
ByteTrack [44]	61.3	59.6	63.4	75.2	77.8
OC-SORT [6]	62.4	62.5	/	76.4	75.9
MOTRv2 (ours)	60.3	58.1	62.9	72.2	76.2
+ MOT17 joint train	61.0	59.3	63.0	73.1	76.2

4.6. Ablation Study

In this section, we study several components of our method, including YOLOX proposals, proposal propagation, and CrowdHuman joint training. Table 5 summarizes the effect of components on DanceTrack validation and test

Table 5. Summary of cumulative improvements on DanceTrack.

	val HOTA	test HOTA
MOTR [1]	51.7	54.2
+ Implementation (Sec. 4.2 MOTR)	54.8	/
+ Propagate boxes	57.1	/
+ YOLOX & CrowdHuman (Tab. 6)	63.7	/
+ Query denoise (Tab. 9)	64.5	69.9
+ Extra association, val set, test ensemble		73.4

Table 6. Ablation study of CrowdHuman joint training and YOLOX proposal on the DanceTrack validation set.

CrowdHuman	YOLOX	HOTA	DetA	AssA
		57.1	66.2	49.5
	✓	60.7	74.8	49.6
✓		56.7	73.7	43.9
✓	✓	**63.7**	**76.6**	**53.2**

Table 7. Ablation study on propagating anchors *vs.* center points and learnable *vs.* sine-cosine positional encoding.

Propagate	Box embedding	HOTA	DetA	AssA
Point	Learnable	61.2	76.6	49.2
Point	Sine-cosine	60.5	76.6	48.0
Box	Learnable	63.8	76.9	53.1
Box	Sine-cosine	63.7	76.6	53.2

Table 8. Effect of using the confidence score of YOLOX proposals and different methods for encoding confidence score.

Score embedding	HOTA	DetA	AssA
Not applied	63.0	77.3	51.5
Linear projection	63.4	77.4	52.1
Sine-cosine	**63.7**	76.6	53.2

sets. The improvements are consistent across both sets.

YOLOX Proposal For a more thorough study of the benefits of using the YOLOX proposal, we test the effect of YOLOX proposals under two settings: with and without CrowdHuman joint training. Table 6 shows that using YOLOX predictions as proposal queries *consistently improves all three metrics* (HOTA, DetA, and AssA) regardless of whether the CrowdHuman dataset is used. The YOLOX proposals significantly improve association accuracy (AssA) by 9.3% when trained jointly with the CrowdHuman dataset. Using the pretrained object detector YOLOX alone outperforms that of joint training with the CrowdHuman dataset (HOTA 56.7 *vs.* 60.7).

Both using YOLOX proposals and CrowdHuman joint training improve detection accuracy as expected. However, using CrowdHuman pseudo-videos seems to have a negative impact on the training of association, as indicated by the 5.6% drop in AssA. This might be caused by the gap between the two datasets: the CrowdHuman pseudo-videos bias the training towards enabling learnable detect queries to handle more difficult detections, and the human motion of pseudo-videos created by affine transformations are different from that of DanceTrack. It is worth noticing that using YOLOX proposals in turn helps CrowdHuman joint training. Our method of using YOLOX proposals makes detection easier for MOTR, thus alleviating the bias toward detection and the conflict between the detection and association tasks. As a result, with the YOLOX proposals, joint training with CrowdHuman can further improve rather than hurt the tracking performance.

Proposal Propagation Here, we show the effect of propagating proposals (center point as well as width and height) from the current frame to the subsequent frame. The baseline for comparison is the propagation of reference point as applied in MOTR [43] and TransTrack [28]. It means that only the center point from the previous frame is employed as the query reference point. In addition, we explore the effect of replacing the learnable positional embedding of queries with the sine-cosine positional encoding of anchors (or reference points). From Table 7, we can easily find that propagating four-dimensional proposals (boxes) instead of center points yields much better association performance. It indicates that MOTRv2 benefits from the width and height information from the bounding box prediction of the previous frame for associating instances. In contrast, the sine-cosine positional encoding barely helps association compared to the original design of using learnable positional embedding in Deformable DETR [49]. Therefore, using the anchor boxes instead of points is not only critical for introducing YOLOX detection results but also sufficient for providing the MOTR decoder with localization information.

Score Encoding As mentioned in sec. 3.4, the proposal queries are the sum of two parts: (1) encoding of the confidence scores; (2) a shared learnable query embedding. We explored two ways to encode the confidence score of YOLOX proposals, namely linear projection and sine-cosine positional encoding. For linear projection, we use a simple weight matrix of size $1 \times D$ to expand the score scalar to a D-dimensional score embedding. Additionally, we test not using confidence scores at all, *i.e.*, only using a shared query embedding for proposal queries. Table 8 shows that not using score embedding performs the worst, which means the confidence score provides important information for MOTR. Further, both learnable embedding and sine-cosine encoding work well, and using sine-cosine encoding works better for the association.

Query Denoising For fast convergence in training, we introduce query denoising [16] (QD) as an auxiliary task for DanceTrack and MOT17. Table 9 shows that query de-

Table 9. Effect of query denoising on the DanceTrack validation set. The definition of noise scale λ_1, λ_2 follows DN-DETR [16].

λ_1	λ_2	HOTA	DetA	AssA
No QD		63.7	76.6	**53.2**
0.4	0.4	63.1	77.7	51.5
0.1	0.05	**64.5**	**78.7**	53.0

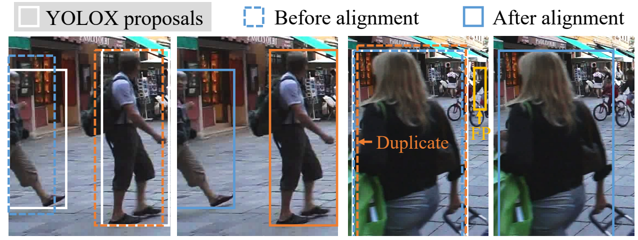
Table 10. Effect of track query alignment on MOT17 valhalf.

Prediction	Anchor	Removal	MOTA	IDF1
			75.9	75.1
		✓	83.0	76.7
	✓		84.3	79.0
	✓	✓	86.3	79.5
✓			77.0	75.4
✓		✓	84.1	77.0
✓	✓		84.9	79.2
✓	✓	✓	**86.8**	**79.7**

noising with the default noise scale (0.4) hurts the association performance. We attribute this to the gap between detection and tracking, as artificial noise is usually larger in scale compared to the cross-frame motion of instances. Our choice of noise range achieves a 2.1% improvement in DetA. Query denoising improves the detection performance and further improves the HOTA metric by 0.8%.

Track Query Alignment To take full advantage of accurate object detection from YOLOX in crowd scenarios, we further introduce track query alignment for enhancing MOTRv2 specifically on the MOT17 [15, 21] and MOT20 [10] datasets. We first calculate the intersection-over-union (IoU) matrix between the MOTR-predicted boxes and YOLOX proposals. Then, we perform Hungarian matching on the IoU matrix to find the best-matched pairs and keep all matched pairs of boxes with an IoU over 0.5. After that, we propose three independent alignment strategies: the matched YOLOX boxes can replace (1) MOTR box *predictions* of that frame and (2) the track query *anchors* for detecting the corresponding instances in the next frame. Further, (3) unmatched MOTR predictions can be *removed* from prediction as they are likely to be false positives. Figure 5 illustrate the effects of these alignments. Note that these alignments only apply to anchor or prediction boxes and do not change the propagation of query embedding, which preserves the end-to-end nature of our method.

We test the three methods on MOT17 using the first half of each training sequence for training and the remaining for validation. All alignments are applied during training and the ablation study of alignment methods is performed during inference. The results are shown in Table 10. Among the three methods, aligning anchors is the most beneficial



(a) Anchor and prediction alignment (b) Unmatched boxes removal

Figure 5. Illustration of Track Query Alignment: (a) imprecise MOTR localization is replaced with corresponding YOLOX proposal boxes for better prediction and anchor localization; (b) false positive detections and duplicate track queries can be eliminated by removing the unmatched boxes.

for detection and tracking performance, as it boosts MOTA by 8.4% and IDF1 by 3.9% when used alone (row 1 *vs.* row 3). Aligning anchors with the corresponding YOLOX proposals mitigates the accumulation of localization errors during anchor propagation, thereby improving both detection and association accuracy (see Figure 5(a)). Removing MOTR predictions that do not match any YOLOX boxes can improve detection performance under all settings. It further improves MOTA by 2.0% in addition to anchor alignment (row 2 *vs.* row 4) (see Figure 5(b)). Finally, frame-by-frame prediction alignment, as an intuitive approach, can be used to further improve MOTA and IDF1.

5. Discussion

In this paper, we propose MOTRv2, an elegant combination of MOTR tracker and YOLOX detector. YOLOX generates high-quality object proposals that help MOTR detect new objects more easily. This reduces the complexity of object detection, allowing MOTR to concentrate on the association process. MOTRv2 breaks through the common belief that end-to-end frameworks are not suitable for high-performance MOT and explains why previous end-to-end MOT frameworks have failed. We hope it can provide some new insights on end-to-end MOT for the community.

Limitations Although using the YOLOX proposals greatly ease the optimization problem of MOTR, the proposed method is still data-hungry and does not perform well enough on smaller datasets. Furthermore, we observe a few duplicated track queries when, for example, when two individuals cross paths with one another. In such cases, one track query might end up following the incorrect subject, leading to two track queries on same individual (see Figure 5(b)). This observation could serve as a valuable hint for potential enhancements in the future. Another limitation is the efficiency. The bottleneck is mainly from the MOTR [43] part. Quantitatively, the YOLOX [11] detector runs at 25 FPS while MOTR runs at 9.5 FPS on 2080Ti. Adding these two components yields a speed of 6.9 FPS.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 2, 6

[2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019. 3, 6

[3] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 5

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. 1, 2

[5] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8100, 2022. 3

[6] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv preprint arXiv:2203.14360*, 2022. 1, 2, 3, 5, 6

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[8] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 3

[9] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, and Zicheng Liu. Transmot: Spatial-temporal graph transformer for multiple object tracking. *arXiv preprint arXiv:2104.00194*, 2021. 1, 2

[10] 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. 6, 8

[11] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 1, 2, 3, 5, 8

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

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

[15] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015. 2, 4, 5, 6, 8

[16] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 7, 8

[17] Siyuan Li, Martin Danelljan, Henghui Ding, Thomas E Huang, and Fisher Yu. Tracking every thing in the wild. In *European Conference on Computer Vision*, pages 498–515. Springer, 2022. 6

[18] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 2, 4

[19] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *IJCV*, 129(2):548–578, 2021. 5

[20] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 1, 3, 6

[21] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 2, 4, 5, 6, 8

[22] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, 2021. 5, 6

[23] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, 2016. 5

[24] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018. 5

[25] Bing Shuai, Andrew G Berneshawi, Davide Modolo, and Joseph Tighe. Multi-object tracking with siamese track-rcnn. *arXiv preprint arXiv:2004.07786*, 2020. 2

[26] Daniel Stadler and Jürgen Beyerer. Modelling ambiguous assignments for multi-person tracking in crowds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 133–142, 2022. 6

[27] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. *arXiv preprint arXiv:2111.14690*, 2021. 1, 4, 5

[28] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv: 2012.15460*, 2020. 1, 3, 5, 6, 7

[29] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, and Masayoshi Tomizuka. Sparse r-cnn: End-to-end object detection with learnable proposals. *arXiv preprint arXiv:2011.12450*, 2020. 3

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[31] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3876–3886, 2021. 6

[32] Shuai Wang, Hao Sheng, Yang Zhang, Yubin Wu, and Zhang Xiong. A general recurrent tracking framework without real data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13219–13228, 2021. 6

[33] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13708–13715. IEEE, 2021. 6

[34] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021. 3

[35] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021. 2

[36] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, 2020. 1, 2

[37] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter, 1995. 2

[38] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 2

[39] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *CVPR*, 2021. 5, 6

[40] Junfeng Wu, Yi Jiang, Wenqing Zhang, Xiang Bai, and Song Bai. Seqformer: a frustratingly simple model for video instance segmentation. *arXiv preprint arXiv:2112.08275*, 2021. 3

[41] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*, 2022. 2, 6

[42] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 4, 5, 6

[43] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision*, pages 659–675. Springer, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[44] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *arXiv preprint arXiv:2110.06864*, 2021. 1, 2, 3, 5, 6

[45] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, pages 1–19, 2021. 1, 2, 5, 6

[46] Zelin Zhao, Ze Wu, Yueqing Zhuang, Boxun Li, and Jiaya Jia. Tracking objects as pixel-wise distributions, 2022. 3, 6

[47] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, 2020. 5, 6

[48] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, 2022. 2

[49] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 3, 7