

Multi-object Tracking by Detection and Query: an efficient end-to-end manner

Shukun Jia^{1, 2}, Yichao Cao^{1, 2}, Feng Yang³, Xin Lu^{1, 2}, Xiaobo Lu^{1, 2},

¹School of Automation, Southeast University, Nanjing 210096, China

²Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, China

³School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

Abstract

Multi-object tracking is advancing through two dominant paradigms: traditional tracking by detection and newly emerging tracking by query. In this work, we fuse them together and propose the tracking-by-detection-and-query paradigm, which is achieved by a Learnable Associator. Specifically, the basic information interaction module and the content-position alignment module are proposed for thorough information Interaction among object queries. Tracking results are directly Decoded from these queries. Hence, we name the method as LAID. Compared to tracking-by-query models, LAID achieves competitive tracking accuracy with notably higher training efficiency. With regard to tracking-by-detection methods, experimental results on DanceTrack show that LAID significantly surpasses the state-of-the-art heuristic method by 3.9% on HOTA metric and 6.1% on IDF1 metric. On SportsMOT, LAID also achieves the best score on HOTA metric. By holding low training cost, strong tracking capabilities, and an elegant end-to-end approach all at once, LAID presents a forward-looking direction for the field.

Introduction

Multi-Object Tracking (MOT) is a vital task in computer vision. Given a video with tracking classes, MOT aims to recognize, localize and assign consistent identification numbers to targets over time. Fundamentally, it can be partitioned into the detection task and association task. How to manage the relationship between the two tasks has been a central theme throughout the development of the field. At present, the field is driven forward by two leading paradigms, the conventional tracking by detection (Bewley et al. 2016; Wojke, Bewley, and Paulus 2017; Zhang et al. 2022) and the emerging tracking by query (Zeng et al. 2022; Gao and Wang 2023), which are progressing in tandem.

The tracking-by-detection paradigm separates the two fundamental tasks apart. Objects are first detected spatially and then associated temporally. It is a clear and well-modularized framework where the association stage becomes the focus. Appearance information and motion patterns are two inherent tracking cues to be considered. Several issues yet reside when integrating the two cues. In the aspect of appearance information, a specific model or branch

is typically exploited to extract appearance features for re-identification (ReID). However, the ReID model needs independent training with extra efforts and its features maybe sub-optimal in the MOT setting (Seidenschwarz et al. 2023). While a branch would raise competition among the detection and association tasks in the major model (Zhang et al. 2021). Regarding to the motion patterns, assumptions are required by motion models to predict object positions. Its effectiveness is limited as they are largely simplified, failing to represent actual motion information. Finally, the two kinds of cues are transformed into an affinity matrix, based on which objects are grouped into trajectories. This is a heuristic process and requires sophisticated designs with hand-crafted hyper-parameters. The property causes the weakness to tackle complex scenarios containing various motion patterns, heavy occlusions etc. In addition, although the tracking-by-detection paradigm focuses on the association stage, complex multi-step settings scatter the paradigm into many cells. As a result, it lacks elegance and overall holism.

The tracking-by-query paradigm carries out the two fundamental tasks simultaneously (Zeng et al. 2022; Gao and Wang 2023). Its models are modified from Transformer-based detectors (Carion et al. 2020) and perform coherently with the help of query mechanism. Compared to tracking-by-detection methods, they achieve remarkable association capabilities. But they abandon the well-modularized framework and couples the two tasks together, which makes them intra-conflicted and cumbersome. As there are works alleviating the confliction problem (Zhang, Wang, and Zhang 2023; Yan et al. 2023; Yu et al. 2023), training the two tasks as a whole remains low efficiency because of the discrepancies between them. Specifically, the detection task focuses on single images while the association task requires consecutive frames to learn temporal cues, within which the spatial information is abundant for detection. Moreover, detectors could easily refer to strong data augmentations like Mosaic, Mixup, etc, to enhance the detection performance, which is nontrivial in the training of a tracker. Last, the coupling feature makes it isolated from the convenience when detectors already have satisfiable detection performance in the tracking scenarios.

Based on the preceding analysis, we cannot help but ask: Can we achieve excellent association capabilities and the elegant end-to-end approach of tracking-by-query mod-

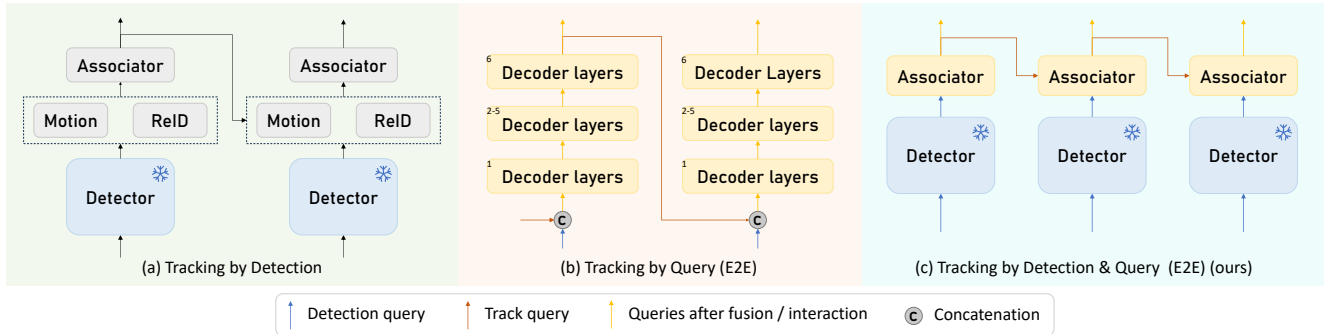


Figure 1: Comparison with two mainstream paradigms. LAID reformulates the MOT task into tracking by detection and query. Compared to traditional tracking-by-detection methods, LAID obtains excellent association capability and has the coherent end-to-end fashion. Compared to tracking-by-query methods, LAID enjoys the convenience of high-performance detectors and achieves remarkably higher training efficiency. The number of decoder layers indicates the index of decoders in tracking-by-query models.

els while still maintaining the clearly structured framework of tracking-by-detection methods? It is an open question. In this work, we answer it by adding a learnable associator upon pretrained detectors, persisting the tracking-by-detection paradigm. Meanwhile, the associator handles objects in the form of object query, which are directly decoded into predictions, following the tracking-by-query paradigm. On top of the two prerequisites, the rest of issues are tackled by the associator. Concretely, we solve them through the interaction and decoding steps. First of all, the Basic Information Interaction (BII) module is proposed to supply interactions among detection queries and track queries. Owing to the BII module primarily focuses on the content part of queries, the Content-Position Alignment (CPA) module is consequentially advocated to update the positional aspect, fostering the alignment of the two parts. Experiencing the BII and CPA modules, the fully interacted object queries are decoded into prediction results via a Transformer decoder layer. To sum up, the tracker with the Learnable Associator could capture complicated tracking cues and realize impressive performance through the Interaction and Decoding process, which is named as LAID.

LAID represents a novel tracking-by-detection-and-query paradigm. It is displayed and compared with previous paradigms in Figure 1. We evaluate LAID on large-scale datasets, DanceTrack and SportsMOT. With simple and effective methods, LAID surpasses the state-of-the-art heuristic tracking-by-detection method Hybrid-SORT (Yang et al. 2024) by 3.9% on HOTA metric and 6.1% on IDF1 metric. When compared to current end-to-end methods, LAID achieves competitive performance in a more efficient manner. The results of SportsMOT also demonstrate the effectiveness of LAID.

Overall, the contributions of this work are summarized as follows.

- We propose LAID to achieve MOT through a novel tracking-by-detection-and-query paradigm, combining low training cost, strong association capabilities and an elegant end-to-end fashion.

- We propose the BII module and the CPA module, guaranteeing the effectiveness of LAID.
- We acquire an impressive balance between tracking accuracy and training cost compared with mainstream MOT methods.

Related Works

Tracking by detection. This paradigm has dominated the MOT field for a long time. Methods are developed around motion patterns and appearance information. SORT (Bewley et al. 2016) adopts the Kalman Filter to predict the location. The Intersection-over-Union (IOU) of the predicted locations and detected boxes determines the matching results through the Hungarian algorithm. Based on SORT, Deep SORT (Wojke, Bewley, and Paulus 2017) introduces the appearance information to improve the robustness against misses and occlusions. Derived from the two methods, the tracking-by-detection paradigm has taken shape. OC-SORT (Cao et al. 2023) updates the motion model of SORT and breaks the limitations of the linear motion assumption. Deep OC-SORT (Maggiolino et al. 2023) adaptively integrates appearance information into the motion model of OC-SORT. JDE (Wang et al. 2020), FairMOT (Zhang et al. 2021) and Track-RCNN (Shuai et al. 2020) propose to jointly learn the detector and appearance embedding. CenterTrack (Zhou, Koltun, and Krähenbühl 2020) simultaneously localizes objects and predicts their offsets in the next frame. ByteTrack (Zhang et al. 2022) considers more low score detection boxes to improve the association ability. GHOST (Seidenschwarz et al. 2023) studies how to better utilize the ReID model in MOT settings, while FineTrack (Ren et al. 2023) explores to use diverse fine-grained representations. PuTR (Liu et al. 2024) upgrades heuristic association strategies to learnable Transformer modules. Although these methods have made noticeable progress, they still underperform in complex situations.

Tracking by query. These methods are emerging in the recent two years. They apply the query mechanism in MOT,

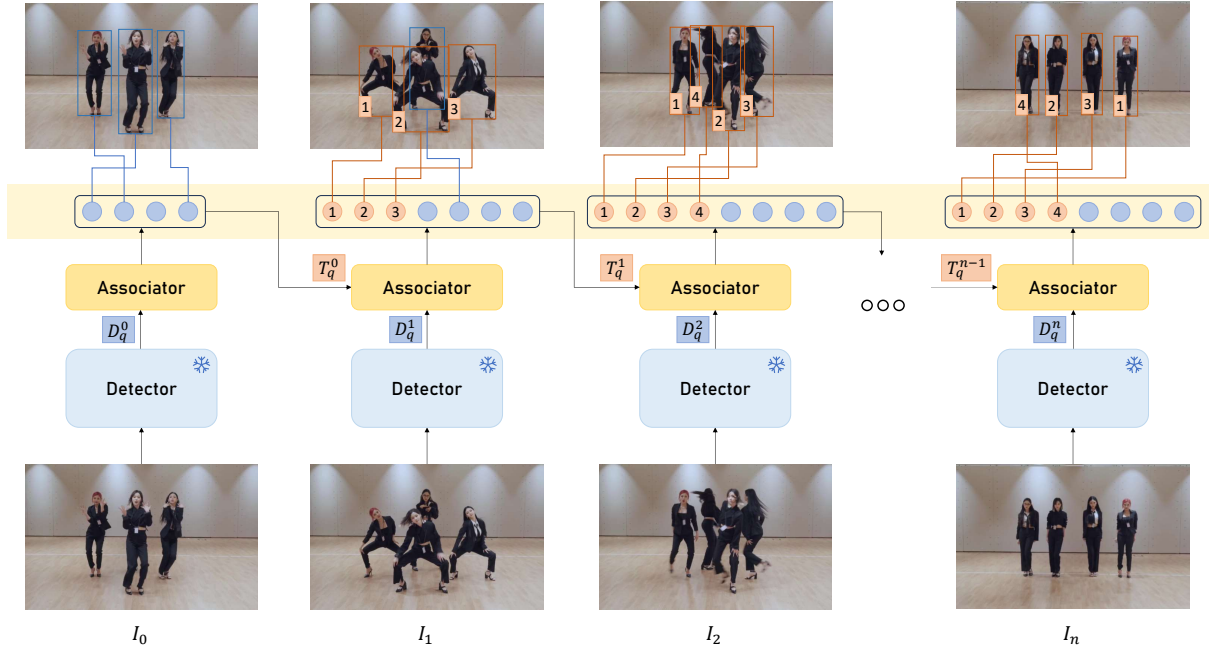


Figure 2: The overall framework of LAID. Detection queries are generated by pretrained detectors. They are responsible for detecting new-born objects. Track queries are initially copied from the detection queries of new-born objects. Afterwards, they are linked to these objects and propagated over time. In the associator, the two types of object queries get interacted and then directly decoded into tracking results. For the first frame I_0 , as track queries do not exist, detection queries are immediately decoded into final results.

where new objects are identified by detection queries and existing ones are connected to track queries. Based on the mechanism, TrackFormer (Meinhardt et al. 2022) explores the end-to-end trainable pipeline. But it still requires extra operations like Non-Maximum Suppression (NMS) operations and ReID features. MOTR (Zeng et al. 2022) first achieves fully end-to-end MOT with the accompanying techniques such as collective average loss and temporal aggregation network. After that, MeMOT (Cai et al. 2022) and MeMOTR (Gao and Wang 2023) get improvements by enhancing the utilization of temporal information. CO-MOT (Yan et al. 2023) and MOTRv3 (Yu et al. 2023) alleviate the conflict problem of MOTR through injecting more supervision on detection queries. Although alleviating the conflict issue, they are inefficient to be trained because of the inherent coupling of detection and association tasks. MOTRv2 (Zhang, Wang, and Zhang 2023) uses the predictions of YOLOX (Ge et al. 2021) as prior knowledge and leads the model to focus more on the association step. But it takes the whole MOTR model as the associator, which is inefficient.

Method

We clarify the components of LAID in this section. Following the outlined tracking-by-detection-and-query paradigm in the beginning, the Basic Information Interaction (BII) module, the Content-Position Alignment (CPA) module and the association decoder are consequentially exhibited. Details of training and inference are also expressed.

Preliminary

Given the frame I_t , the pretrained detector produces object embeddings and bounding boxes, constituting detection queries in our literature. Inspired by Conditional-DETR (Meng et al. 2021) and DAB-DETR (Liu et al. 2022), we set object embeddings as the content part and bounding boxes as the positional part. Similarly, track queries containing the two aspects are generated by the associator and linked to existing objects. The two kinds of object queries merely interact in the associator and are directly decoded into tracking results. The overview of the paradigm is displayed in Figure 2. The key ingredient is the trainable associator, with which high tracking performance is acquired and the end-to-end fashion is fulfilled. The structure of the associator is shown in Figure 3. In light of the fact that the associator is based on Transformer blocks, incorporating CNN-based detectors into the framework is slightly different from Transformer-based detectors, which is described in the appendix.

Basic Information Interaction Module

The Basic Information Interaction module is proposed to facilitate the information exchange among the content part of object queries. Simply and intuitively, it is achieved by the scaled dot-product attention:

$$\begin{aligned}
O_1 &= \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \right) \cdot V_1 \\
O_2 &= \text{norm} (O_1 + V_2) \\
O_3 &= \text{norm} (FFN(O_2) + O_2)
\end{aligned} \tag{1}$$

In Equation (1), Q and K represent the variable Query and Key. We make a modification on Value and split it into V_1 and V_2 . d is the dimension of these variables. On top of Equation (1), we update detection queries by setting:

$$\begin{aligned}
Q &= \widetilde{D}_q, K = \text{concat}(\widetilde{D}_q, \widetilde{T}_q) \\
V_1 &= \text{concat}(D_q, N_q), V_2 = D_q
\end{aligned} \tag{2}$$

While for track queries, we have:

$$\begin{aligned}
Q &= \widetilde{T}_q, K = \text{concat}(\widetilde{D}_q, \widetilde{H}_q) \\
V_1 &= \text{concat}(D_q, H_q), V_2 = T_q
\end{aligned} \tag{3}$$

In these equations, D_q , T_q and H_q denote the content part of detection query, track query and history track query, respectively. \widetilde{D}_q , \widetilde{T}_q and \widetilde{H}_q represent the full version of queries, including the content part and positional part:

$$\widetilde{E} = E + P_e(E_B), E \in \{D_q, T_q, H_q\} \tag{4}$$

where E_B are the corresponding bounding boxes and $P_e(E_B)$ are the positional encoding of them, constituting the positional part of queries.

The update of two types of queries has different considerations, thus Equation (2) and (3) are implemented by the BII with different parameters. To reduce the impact of background noise in the interaction process, we set the threshold τ_q to filter out low quality detection queries before the BII module. Additionally, two key points are worth noting. In the update of detection queries, the second item of V_1 is specifically set as noisy queries N_q , helping to crumble the detection queries that have high attention weights with track queries. Because these detection queries are likely linked to existing objects, which have been responsible by track queries. To enhance the learning ability of models, N_q are formed by hard negative samples. They are the low quality detection queries with the highest M scores. M is the number of track queries that are propagated from the last moment. In the update of track queries, we collect history track queries H_q to cope with missing objects. Through the BII module, track queries could be enhanced by H_q when the related objects disappear temporarily. H_q are collected by the Exponential Moving Average method:

$$H_q^t = w * T_q^t + (1 - w) * H_q^{t-1} \tag{5}$$

where w indicates the update weight of new information. \widetilde{H}_q share the same positional part with \widetilde{T}_q for better calculating attention weights.

Content-Position Alignment module

The Content-Position Alignment (CPA) module is advocated to update the positional part of object queries and align them to the content part. Considering that cross-attention

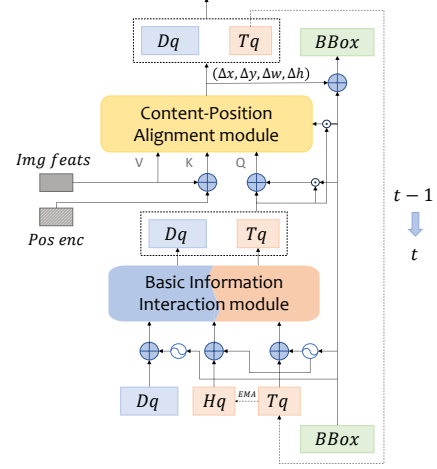


Figure 3: Illustration of the associator. The BII module is depicted with two different colors, indicating the updates of detection queries and track queries use the same module structure but do not share parameter weights.

modules of DETR-like detectors could refine the position information based on embeddings and update the embeddings simultaneously, the CPA module is straightforwardly constituted by these cross-attention modules. In this work, we exploit the modulated cross-attention from DAB-DETR as the CPA module. Finally, a group of auxiliary losses are introduced to supervise the CPA module during training. They share the same label assignment with the calculation of the final losses.

Association Decoder

Through the interaction via the above two modules, object queries are directly decoded into predictions, which is realized by the decoder layer of DAB-DETR. Compared to the matching strategies that group objects into trajectories based on affinity matrices, this manner forms a fully end-to-end fashion. Another benefit is the stronger association capability, which is demonstrated in the experiment section.

Training and Inference

During training, we follow MOTR to take label assignment and calculate the final loss. New-born objects are assigned to the outputs of detection queries via the bipartite matching. And existing objects are matched to the predictions of their linked track queries. In the inference mode, we set the threshold τ_n to discard negative predictions. For the detection queries whose confidence scores are larger than τ_n , its predictions are regarded as new-born objects, which are added to tracklets and equipped with a new ID number. For the results of track queries below τ_n , the related objects are marked as the inactive state. Inactive objects that remain for consecutive T frames will be removed from the tracklets. T is set as 20 in our experiments.

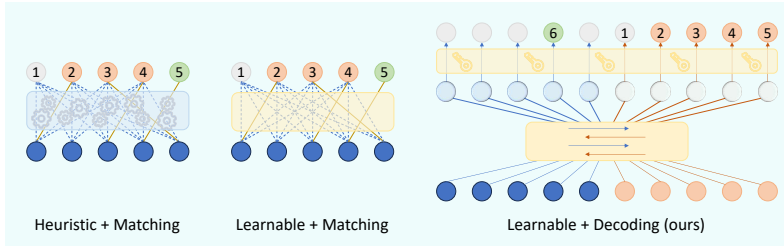


Figure 4: Comparisons among tracking-by-detection methods. LAID distinguishes traditional TBD methods from two aspects. First, it captures tracking cues through learning instead of heuristic algorithms. Second, predictions are decoded from the interacted object queries, instead of through matching on affinity matrix. The circles in top row represent the tracking results. Gray circles are negative results, while colored circles are positive ones. Specifically, circles in green, orange and blue denote new-born objects, tracked objects and detection results, respectively. The circles in the middle row are hidden states, which have undergone interactions and are about to be decoded into tracking results.

TBD/T&D	H/L	M/D	Representatives
TBD	H	M	SORT
TBD	L	M	PuTR
T&D	L	D	MOTR
TBD	L	D	LAID (ours)

Table 1: Categories of current MOT methods. TBD is tracking-by-detection. And T&D signifies tracking-by-query, where the tracking and detection are coupled and jointly undertaken. H/L indicates the association algorithm is heuristic or learnable. M/D represents the final results are derived from affinity matching or decoded from the model.

Methods	Total	Asso.	Training Time
MOTR	43.9	10.3	60h (8*V100)
MOTRv2	41.7	8.2	-
CO-MOT	36.4	8.2	-
MOTIP	58.9	18.4	36h (8*4090)
LAID (ours)	5.3	5.3	22h (2*4090)

Table 2: Efficiency comparison. ‘Total’ is the total number of trainable parameters and ‘Asso.’ implies the number of parameters in association modules. They are measured in millions. The training time refers to the time spent training on DanceTrack.

Discussions

The diversity of MOT methods provides multiple perspectives to depict them. To highlight the distinction of LAID, we label related works from three progressive aspects, which are listed in Table 1. In addition to the basic division of TBD and T&D, we further categorize methods into Heuristic (H) and Learnable (L), according to whether the tracking cues are heuristically integrated or adaptively learned. On top of tracking cues, methods can be subsequently indicated as M/D depending on the final results are generated through affinity matching or direct decoding.

To compare with other tracking-by-detection methods, present combinations of H/L and M/D are enumerated in Figure 4. H+M is the popular and plain scheme in this family. And they have been constantly developed up to now. Derived from that, L+M methods adopts learnable strategies to compute the similarity between detection results and trajectories, aiming to get robust tracking cues. But they still need individual matching algorithms and the tracking quality is unsatisfied in challenging situations. In comparison, LAID introduces H+D approach from the tracking-by-query paradigm and procures strong tracking capabilities, embodying a novel direction of the MOT field.

There is one prior work, MOTIP (Gao, Zhang, and Wang 2024), that also adopts the TBD+L+D scheme. But discrepancies still exist. MOTIP reformulates the association stage as an ID prediction task. Getting the detection results, it sim-

ply stacks six decoder layers to classify them into ID labels according to historical trajectories. In contrast, LAID proposes specific designs within the interaction between object queries, which is more efficient. In addition, LAID demonstrates that the present tracking-by-query paradigm could be straightforwardly decoupled through a simple and effective learnable association module.

Experiments

Datasets

LAID is mainly evaluated on two large-scale datasets, DanceTrack (Sun et al. 2022) and SportsMOT (Cui et al. 2023). DanceTrack is widely used in MOT because it provides sufficient data with high quality annotations. It contains similar appearance yet diverse non-linear motion patterns, severe occlusions and frequent clustering, posing great challenges in the association task. SportsMOT is a recently proposed dataset that centers on the applications of sports analysis. Fast and variable-speed motions on sports courts embody its challenge. In statistic, DanceTrack consists of 100 video sequences. Each sequence contains 1058 frames on average. SportsMOT includes 240 video sequences in total. The average number of frames in each sequence is 439.

Methods	HOTA	DetA	AssA	MOTA	IDF1
<i>T&D+L+D</i>					
MOTR	54.2	73.5	40.2	79.7	51.5
MOTRv2	69.9	83.0	59.0	91.9	71.7
MeMOTR	68.5	80.5	58.4	89.9	71.2
CO-MOT	69.4	82.1	58.9	91.2	71.9
<i>TBD+H+M</i>					
CenterTrack	41.8	78.1	22.6	86.8	35.7
FairMOT	39.7	66.7	23.8	82.2	40.8
QDTrack	54.2	80.1	36.8	87.7	50.4
FineTrack	52.7	72.4	38.5	89.9	59.8
ByteTrack*	47.7	71.0	32.1	89.6	53.9
OCSORT*	55.1	80.3	38.3	92.0	54.6
GHOST*	56.7	81.1	39.8	91.3	57.7
DSORT*	61.3	82.2	45.8	92.3	61.5
HSORT*	65.7	-	-	91.8	67.4
<i>TBD+L+M</i>					
PuTR*	55.8	-	-	91.9	58.2
<i>TBD+L+D</i>					
MOTIP	70.0	80.8	60.8	91.0	75.1
LAID (ours)*	69.6	81.1	59.9	89.9	73.5

Table 3: Comparisons on DanceTrack. Methods marked with * share the same pretrained detector YOLOX. The best scores on each metric are marked in **bold**. The second to fourth best scores are displayed in **red**, **blue**, and **green** font, respectively.

Metrics

Results of ablation studies are reported on the metrics: HOTA (Luiten et al. 2021), DetA, AssA, MOTA (Bernardin and Stiefelhagen 2008) and IDF1 (Ristani et al. 2016). HOTA is the geometric mean of DetA and AssA. DetA and MOTA emphasize detection performance, while AssA and IDF1 reflect association capability.

Implementation details

LAID is implemented via Pytorch. All experiments are conducted on two NVIDIA 4090 GPUs. On DanceTrack, we empirically set the threshold τ_q to 0.3, and set τ_n to 0.5. The model is trained in 12 epochs, with the initial learning rate as 1.2×10^{-4} and dropped by 10 at the 6th and 10th epoch respectively. On SportsMOT, whose detection is relatively easier and predictions tend to be high-confidence, the hyperparameters τ_q and τ_n are 0.4 and 0.6, respectively. The initial learning rate is the same as DanceTrack and the corresponding training schedule is [12, 16, 20]. On both datasets, LAID is trained with the publicly available detector YOLOX. In all experiments, we follow CO-MOT to set the clip-length as 5 and the sampling interval is randomly chosen from 1 to 10 for each iteration. Hyperparameters are not tuned specifically. We believe tuning them would further improve the performance. Ablation studies are conducted on DanceTrack with a pretrained DAB-DETR in smaller resolutions. More details are expressed in the appendix.

Comparison with other methods

We compare the efficiency of LAID with representative L+D models in the beginning. On DanceTrack, we compare LAID with other methods according to the categories listed in Table 1. Results on SportsMOT are also reported.

Efficiency. Thanks to the decoupled property of tracking by detection and the proposed associator, LAID owns notably higher training efficiency than other L+D methods. In Table 2, we list the number of parameters need to be trained in several representative models. Because LAID fetches pre-trained detectors from public sources and freezes them in the TBD framework, the total number of trainable parameters in LAID is significantly lower than in other methods. Although MOTIP follows the TBD structure as well, it needs to jointly train the detector and its ID predictor to obtain competitive performance. Besides, among all association modules, the number of trainable parameters in LAID is also the lowest, which is result of effective designs of the associator. In contrast, other methods simply stack multiple decoder layers to achieve association. Because of the above reasons, training LAID on DanceTrack takes the lowest cost in these methods.

Comparison with T&D+L+D methods. Retaining the L+D paradigm, comparison with these methods could strictly test the effect of turning TBD to T&D. From Table 3, LAID achieves competitive performance among these methods, demonstrating that the proposed learnable associator is effective to decouple the end-to-end framework into tracking-by-detection. As a benefit, LAID obtains prominently higher training efficiency after decoupling, which has been shown in Table 2. It is noted that MOTRv2 does not change the structure of MOTR essentially. It exploits the prior knowledge of detection from YOLOX and leads the cumbersome MOTR to focus on the association task, resulting in a higher training cost.

Comparison with TBD+H+M methods. With the rapid development of object detection, these methods could easily get superior detection performance. But the tracking cues that are explored by heuristic algorithms make them lack capabilities in association, especially in complicated cases. To eliminate the impact of basic detection quality, we adopt the same pretrained detector with previous methods. From Table 3, LAID outperforms the best model Hybrid-SORT-ReID by 3.9% on HOTA. It is also noted that LAID improves AssA more than 10% and IDF1 at least 6% with detection performance slightly sacrificed.

Comparison with the TBD+L+M method. LAID surpasses the TBD+L+M method PuTR by a large margin. Although PuTR uses learnable module to capture tracking cues. We infer the reason lies in the generation of tracking results. In matching strategies, tracking cues are fused in affinity matrix. While in LAID, they are captured in the interaction module. The additional decoding process in LAID helps the model to better understand scenarios and thus capture more robust tracking cues. However, the strict study and comparison between them is beyond the scope of this work.

Comparison on SportsMOT. Among various methods, the gaps on SportsMOT are less pronounced than on DanceTrack. From Table 4, LAID achieves competitive performance in the *train* setting. When the quantity of training

Methods	Paradigm	<i>train</i>					<i>train+val</i>					ΔH
		HOTA	DetA	AssA	MOTA	IDF1	HOTA	DetA	AssA	MOTA	IDF1	
ByteTrack*	TBD+H+M	62.8	77.1	51.2	94.1	69.8	64.1	78.5	52.3	95.9	71.4	1.3
OC – SORT*		71.9	86.4	59.8	94.5	72.2	73.7	88.5	61.5	96.5	74.0	1.8
PuTR*	TBD+L+M	73.0	-	-	95.1	74.2	-	-	-	-	-	-
MeMOTR	T&D+L+D	70.0	83.1	59.1	91.5	71.4	-	-	-	-	-	-
MOTIP	TBD+L+D	71.9	83.4	62.0	92.9	75.0	75.2	86.5	65.4	96.1	78.2	3.3
LAID (ours)*	TBD+L+D	71.7	82.5	62.4	89.2	72.7	75.5	87.2	65.5	94.4	76.0	3.8

Table 4: Comparisons on SportsMOT. The setting of *train* only includes the training subset into training and the setting of *train+val* further incorporates the validation subset. ΔH indicates the improvement of HOTA from the *train* setting to the *train+val* setting. Methods marked with * share the same pretrained detector YOLOX.

BII	CPA	HOTA	DetA	AssA
✓		54.9	66.0	46.1
	✓	52.9	65.6	43.1
✓	✓	58.9	71.5	48.9
self-attn	✓	57.6	71.1	47.0

Table 5: Contributions of each component of LAID. ‘self-attn’ means to replace the BII module with a general self-attention block of Transformer models.

BII on Det. queries	HOTA	DetA	AssA
track query	57.9	71.1	47.4
hard noisy query	58.9	71.5	48.9
easy noisy query	57.6	71.2	47.0
zeros	57.1	71.3	46.0
ones	58.1	71.1	47.8

Table 6: Different settings for the BII module on detection queries.

data increases, however, LAID gains the highest HOTA scores. Meanwhile, from the *train* setting to the *train+val* setting, LAID achieves the largest improvement on HOTA. It demonstrates that sufficient data could drive LAID to learn better tracking cues, which is a characteristic of LAID.

Ablation studies

LAID Components. We exhibit the contribution of the BII module and the CPA module in Table 5. It can be seen that the best scores could be reached when the two modules work together. Given the fact that the BII module is very similar to the self-attention blocks within general Transformer models, we take the replacement to test their performance. From Table 5, there is 1.3% gap on HOTA between the two settings, suggesting the specific designs of the BII module are effective to the MOT task.

Interaction on detection queries. As described above, the second item of V_1 in Equation (2) is set as noisy queries N_q to alleviate the confliction among detection queries and track queries. We try different choices of N_q in Table 6. The first setting does not use noisy queries but normal track queries. It gets 1% lower HOTA score than the setting of hard noisy queries, representing that the solution is convincing. We also try other alternatives of N_q , such as easy noisy

BII on Track queries		HOTA	DetA	AssA
Track query		57.9	72.1	46.9
History track query	0.9	58.3	72.0	47.5
	0.8	58.1	71.3	47.7
	0.7	58.9	71.5	48.9
	0.6	58.1	71.1	47.9
Track query and Hist query		57.3	71.6	46.2

Table 7: Different settings for the BII module on track queries.

queries, all zeros or all ones, but they get inferior performance. Besides, in these settings, the fluctuation is mainly reflected on the AssA metric, while the DetA scores roughly maintain at the same level. Because an improper setting of N_q can cause track queries to be disrupted by detection queries through Equation (2), negatively impacting the association performance.

Interaction on track queries. We explore various configurations for K and V_1 in Equation 3 to study the effect when track queries are updated by different information sources. Detection queries, being the primary source of information for track queries, are consistently included in each configuration. Results of Table 7 show that merely track queries and the combination of track queries and history track queries achieve inferior performance. It can be concluded that information of past moments will be overlooked when there are merely track queries. But putting them and history track queries together will bring redundant current information. The proper setting of w when updating history track query (Equation 5) is inductive to the best performance. To further analyze the effect of w , we test the scores by varying it from 0.9 to 0.6. The results are identical to our intuition that old information helps to association but current information contributes to detection.

Conclusion

In this work, we introduce LAID to establish a novel tracking-by-detection-and-query paradigm for MOT. LAID demonstrates impressive merits such as low training cost, strong association capabilities and an elegant end-to-end manner. However, it still has certain limitations. Firstly, it needs sufficient data to perform effectively. The performance will diminish when the data is scarce, Secondly, its

compatibility to work with different detectors is valuable to be studied in the future. Nonetheless, we believe LAID presents a promising and innovative direction for the field.

References

- 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 *2016 IEEE international conference on image processing (ICIP)*, 3464–3468. IEEE.
- 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.
- Cao, J.; Pang, J.; Weng, X.; Khirrodar, R.; and Kitani, K. 2023. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9686–9696.
- Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.
- Cui, Y.; Zeng, C.; Zhao, X.; Yang, Y.; Wu, G.; and Wang, L. 2023. Sportsmot: A large multi-object tracking dataset in multiple sports scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9921–9931.
- 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.
- Gao, R.; Zhang, Y.; and Wang, L. 2024. Multiple Object Tracking as ID Prediction. *arXiv preprint arXiv:2403.16848*.
- Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.
- Liu, C.; Li, H.; Wang, Z.; and Xu, R. 2024. PuTR: A Pure Transformer for Decoupled and Online Multi-Object Tracking. *arXiv preprint arXiv:2405.14119*.
- 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 preprint 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 *2023 IEEE International Conference on Image Processing (ICIP)*, 3025–3029. IEEE.
- Meinhardt, T.; Kirillov, A.; Leal-Taixé, 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.
- Meng, D.; Chen, X.; Fan, Z.; Zeng, G.; Li, H.; Yuan, Y.; Sun, L.; and Wang, J. 2021. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3651–3660.
- Ren, H.; Han, S.; Ding, H.; Zhang, Z.; Wang, H.; and Wang, F. 2023. Focus on details: Online multi-object tracking with diverse fine-grained representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11289–11298.
- Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; and Tomasi, C. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, 17–35. Springer.
- Seidenschwarz, J.; Brasó, G.; Serrano, V. C.; Elezi, I.; and Leal-Taixé, L. 2023. Simple cues lead to a strong multi-object tracker. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13813–13823.
- Shuai, B.; Berneshawi, A. G.; Modolo, D.; and Tighe, J. 2020. Multi-object tracking with siamese track-rcnn. *arXiv preprint arXiv:2004.07786*.
- 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.
- Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; and Wang, S. 2020. Towards real-time multi-object tracking. In *European conference on computer vision*, 107–122. Springer.
- Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, 3645–3649. IEEE.
- 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 preprint arXiv:2305.12724*.
- Yang, M.; Han, G.; Yan, B.; Zhang, W.; Qi, J.; Lu, H.; and Wang, D. 2024. Hybrid-sort: Weak cues matter for online multi-object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6504–6512.
- 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 preprint 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 *European Conference on Computer Vision*, 659–675. Springer.
- 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 *European conference on computer vision*, 1–21. Springer.
- 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: 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 *European conference on computer vision*, 474–490. Springer.

Appendix

In this appendix, we provide a detailed explanation of how LAID is built upon the YOLOX detector. Then, more training details are expressed. Finally, we reveal the working process of the Basic Information Interaction module by visualizing the attention weights.

Building LAID upon the YOLOX Detector

In the proposed associator, the final decoding process is implemented by the decoder of DAB-DETR, which needs encoder features to decode object queries into predictions. In the YOLOX (Ge et al. 2021) detector, however, both the encoder features and object queries are absent. To get encoder features, we simply map backbone features through an MLP module. As to object queries, they are essentially a representation of objects. We first fetch predictions from YOLOX, with the NMS threshold as 0.9 and the confidence score as 5×10^{-4} . Then, their object queries are sampled from backbone features according to the bounding boxes. Backbone features from the last three levels are leveraged. Finally, we pad these queries using a group of learnable parameters so that the total number of object queries is 300, aiming to improve the training stability.

More Training Details

Image augmentation has a direct impact on the final performance. In this work, we follow the augmentation strategy from CO-MOT (Yan et al. 2023). Besides, two versions of image resolution are adopted in this work. In the ablation studies, we adopt the version of smaller resolution for fast iteration. Images are pre-processed via scale augmentation, where the shortest side ranges from 480 to 800 pixels with a step of 32 pixels, and the longest side is at most 1333 pixels. The version of larger resolution is exploited to make a relatively fair comparison with other methods. Images are resized so that the shortest side is at least 608 and at most 992 pixels while the longest side is at most 1536.

Visualization of Attention Weights

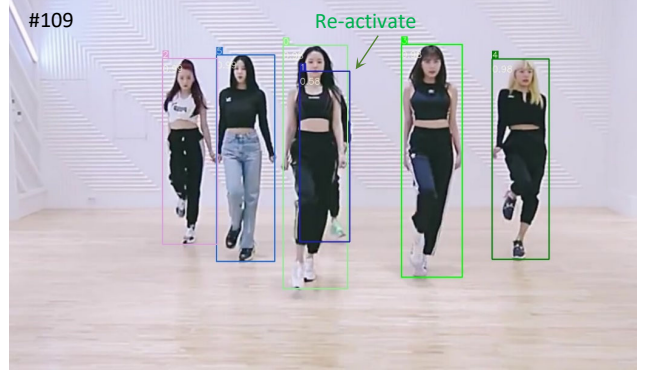
To disclose the actual function of the Basic Information Interaction (BII) module, we visually demonstrate its working process through two cases.

According to Equation (2) in the main paper, when updating detection queries, the attention weights are represented as $W_d(\widetilde{D}_q, \widetilde{D}_q)$ and $W_d(\widetilde{D}_q, \widetilde{T}_q)$. The former is comparable to the weights of self-attention among D_q , while the latter refers to the weights of noisy queries N_q , which are used to disrupt the detection queries and alleviate the confliction between detection queries and track queries. Similarly,

based on Equation (3), the attention weights are divided into $W_t(\widetilde{T}_q, \widetilde{D}_q)$ and $W_t(\widetilde{T}_q, \widetilde{H}_q)$, with which track queries are updated by detection queries and history track queries, respectively.

Two general cases are displayed in Figure 5 and 6. In Figure 5(b) and 6(b), gray cells are filled with large (top-2) attention weights, indicating that these detection queries are effectively disrupted by corresponding track queries, avoiding its final competition with track queries. From Figure 5(c) and 6(c), track queries are mainly updated by their related history track queries. Specially, as shown in Figure 5(a), there is an inactive track query (#1) that is re-activated. In this process, its detection query (#1) plays a more significant role than in typical situations, which is exhibited in Figure 5(c). From Figure 6(a), a new-born object (#6) is generated. Attention weights in Figure 6(b) reveal that the detection query (#6) is not disrupted and thus recognized as a new-born object.

In summary, the working process of the BII module is identical to our expectation. Last but not the least, the learned attention weights of the BII module are similar to the affinity matrices that are employed to match detections with trajectories in Matching-based methods. Different from those methods, LAID does not explicitly utilize these weights, but further makes interactions and decodes object queries based on the weights, which enables the model to understand scenarios more thoroughly and capture more robust tracking cues.



(a) The case where an inactive tracked object (#1) is re-activated.

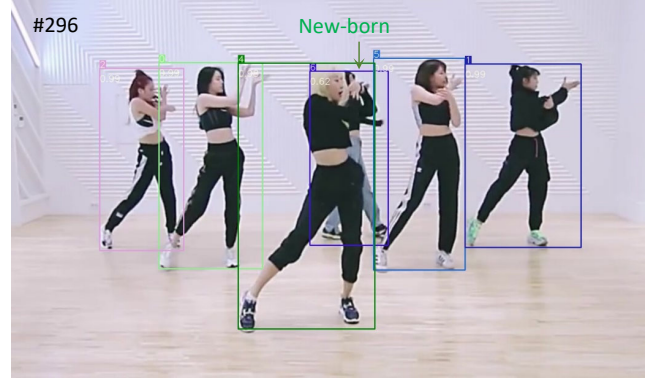
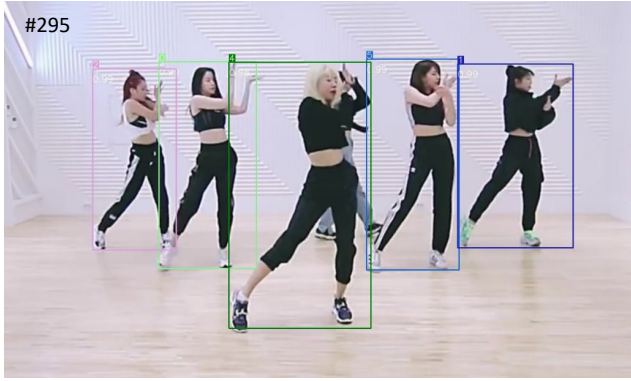
	0	1	2	3	4	5	0	1	2	3	4	5
0	0.154	0.0395	0.0524	0.0902	0.0425	0.0845	0.1436	0.0953	0.0637	0.0949	0.0522	0.0872
1	0.0675	0.1562	0.0714	0.0723	0.0682	0.0776	0.0659	0.1718	0.0613	0.0621	0.0564	0.0692
2	0.0645	0.0754	0.1468	0.0433	0.0386	0.1159	0.0767	0.148	0.1173	0.0432	0.029	0.1013
3	0.0938	0.0494	0.0314	0.1396	0.1126	0.0462	0.0936	0.104	0.0344	0.1302	0.1135	0.0511
4	0.049	0.0589	0.0237	0.1258	0.2102	0.0291	0.0571	0.1068	0.0239	0.1151	0.1667	0.0337
5	0.0931	0.0674	0.1002	0.0597	0.042	0.1117	0.0996	0.1319	0.0958	0.0601	0.0369	0.1014

(b) The illustration of $W_d(\widetilde{D}_q, \widetilde{T}_q)$ (the left part with the orange column header) and $W_d(\widetilde{D}_q, \widetilde{D}_q)$ (the right part with the blue column header). Gray cells indicate that detection queries are disrupted by their corresponding track queries.

	0	1	2	3	4	5	0	1	2	3	4	5
0	0.1713	0.0636	0.076	0.1258	0.0817	0.1138	0.0912	0.0464	0.0559	0.0634	0.0495	0.0614
1	0.0965	0.2001	0.0864	0.0985	0.1219	0.0958	0.022	0.1167	0.0435	0.0316	0.0543	0.0327
2	0.103	0.0559	0.2781	0.0654	0.0502	0.1635	0.0396	0.0542	0.0838	0.0289	0.0223	0.0553
3	0.1301	0.0561	0.0463	0.1778	0.1878	0.0692	0.0606	0.0436	0.0371	0.0627	0.0855	0.0433
4	0.084	0.053	0.0395	0.1539	0.3227	0.055	0.0294	0.0434	0.0244	0.0523	0.1157	0.0266
5	0.1479	0.0572	0.174	0.0916	0.0656	0.1609	0.0517	0.0536	0.0691	0.0391	0.0344	0.0549

(c) The illustration of $W_t(\widetilde{T}_q, \widetilde{H}_q)$ (the left part with the pink column header) and $W_t(\widetilde{T}_q, \widetilde{D}_q)$ (the right part with the blue column header). Yellow cells imply that track queries are mainly updated by their history track queries. The green cell means the track query (#1) is re-activated.

Figure 5: Illustration of the inference process on Frame #109. The attention weights of the BII module used to update the detection queries (Equation (2) in the main paper) and track queries (Equation (3) in the main paper) are shown in Figure 5(b) and Figure 5(c), respectively. Rows represent Q , while columns signify K . The indexes of tables are identical to the ID number of Figure 5(a). For each query in Q , the largest weight value is marked in red, while the second and third highest are marked in blue and green (optional). Headers with the blue, orange and pink backgrounds denote detection queries, track queries and history track queries, in turn.



(a) The case where a new-born object (#6) is recognized. Previously, the object was removed from the tracklets due to the long-time missing. In consequence, when it re-appears, it is recognized as a new-born object.

	0	1	2	4	5	0	1	2	4	5	6
0	0.1391	0.0305	0.1151	0.1025	0.0482	0.142	0.0288	0.1103	0.1226	0.0545	0.1063
1	0.037	0.2362	0.0256	0.0438	0.1392	0.0339	0.1702	0.0213	0.0588	0.1089	0.1251
2	0.1408	0.0306	0.2121	0.0481	0.0357	0.1302	0.0242	0.1463	0.0834	0.0378	0.1108
4	0.0718	0.0231	0.0354	0.2773	0.0579	0.0913	0.0342	0.0467	0.207	0.0797	0.0757
5	0.0601	0.0965	0.0336	0.1169	0.134	0.0601	0.0998	0.0323	0.1126	0.1268	0.1273
6	0.0871	0.0893	0.069	0.0532	0.0963	0.077	0.0741	0.0579	0.0759	0.0881	0.2322

(b) Illustration of $W_d(\widetilde{D}_q, \widetilde{T}_q)$ (the left part with the orange column header) and $W_d(\widetilde{D}_q, \widetilde{D}_q)$ (the right part with the blue column header). The green cell indicates the detection query (#6) is recognized as a new-born object.

	0	1	2	4	5	0	1	2	4	5	6
0	0.1889	0.0781	0.1617	0.1105	0.0993	0.0588	0.0415	0.0946	0.0558	0.0386	0.0722
1	0.0731	0.3521	0.0508	0.0509	0.1697	0.0269	0.1221	0.0279	0.0265	0.0382	0.0618
2	0.1798	0.0617	0.3015	0.0465	0.0735	0.0584	0.0259	0.1234	0.0346	0.029	0.0658
4	0.0763	0.0737	0.0484	0.2493	0.1287	0.0647	0.0507	0.0536	0.1482	0.0617	0.0447
5	0.0768	0.2169	0.0476	0.1298	0.1986	0.0356	0.0956	0.0372	0.0461	0.0427	0.0732

(c) Illustration of $W_t(\widetilde{T}_q, \widetilde{H}_q)$ (the left part with the pink column header) and $W_t(\widetilde{T}_q, \widetilde{D}_q)$ (the right part with the blue column header).

Figure 6: Illustration of the inference process on Frame #296. Unless specifically noted, the elements in this figure have the same meaning as those in Figure 5.