2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Multi-Object Tracking by Hierarchical Visual Representations

Jinkun Cao[1], Jiangmiao Pang[2] and Kris Kitani[1]
[1]Carnegie Mellon University [2]Shanghai AI Laboratory

***Abstract*— We propose a new visual hierarchical representation paradigm for multi-object tracking. It is more effective to discriminate between objects by attending to objects' *compositional* visual regions and contrasting with the background *contextual* information instead of sticking to only the *semantic* visual cue such as bounding boxes. This *compositional-semantic-contextual* hierarchy is flexible to be integrated in different appearance-based multi-object tracking methods. We also propose an attention-based visual feature module to fuse the hierarchical visual representations. The proposed method achieves state-of-the-art accuracy and time efficiency among query-based methods on multiple multi-object tracking benchmarks.**

I. Introduction

Discriminative visual representations can help avoid mismatches between different targets in appearance-based association for multi-object tracking. We propose a new visual representation paradigm by fusing visual information from different spatial regions in a hierarchy. We argue that, compared to the common paradigm of only using features from bounding boxes, the proposed hierarchical visual representation is more discriminative and no extra annotations are required.

In modern computer vision, we typically use bounding boxes or instance masks to define the area of an object of interest. Because the enclosed pixel area is bonded with a certain object category, such a representation is usually considered as *semantic*. However, we find that not just the *semantic* cues can make informative representations for visual recognition. We can generate more discriminative visual representations from the other two perspectives to define the existence of an object: *compoistional* and *contextual*. Compositional cues describe how the parts of a target look like and contrast cues describe how a target looks different from others. For example, as shown in Figure 1, multiple flamingo individuals are in almost undistinguishable appearance to us. But by focusing on the distinguishable parts of certain individuals, such as the shape of the wing red mark, we can easily spot the individual (*compositional*). We can also be more confident in distinguishing instances if we can compare all individuals across timesteps (*contrast*).

We thus build discriminative visual representations from three perspectives: *compositional*, *semantic*, and *contextual*. The *semantic* level, such as a tight bounding box or instance segmentation mask, defines the occupancy area of the object with certain visual existence and semantic concept. The compositional level suggests the salient visual regions of an object instance, with which, ideally, we can track it even without seeing its full body. The *contextual* information helps to highlight a subject via contrast with background pixels and other instances. For example, we often have a hard time

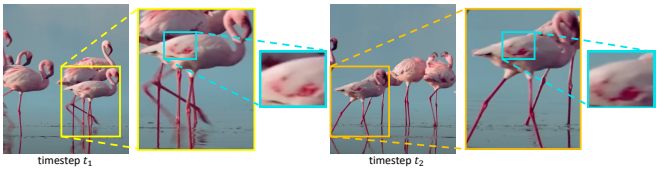

Fig. 1: With a close look at distinct compositional visual regions, we can recognize certain individuals much more easily.

determining whether two object instances are the same one. However, it is typically easier to determine whether one instance is more likely to be the same one than another. Motivated by the insight, we propose to represent an object by a three-level hierarchy, i.e., *Compositional*, *Semantic*, and *Contextual*.

We adopt the proposed visual hierarchy in video multi-object tracking to avoid the mismatch among different targets. We find that it is crucial how the representations from levels are leveraged together. The naive way of stacking or concatenating them does not show a significant performance advantage. Instead, we propose an attention-based module called CSC-Attention to fuse the features. The core idea of CSC-Attention is to leverage the attention-based mechanism to attend to the salient areas on the target subject body by contrasting to the background pixels close to it. Discriminating targets by the fused features, the multi-object tracker we construct is named CSC-Tracker . It leverages global association by a transformer to effectively track objects over time. Through experiments on multiple multi-object tracking datasets, CSC-Tracker achieves state-of-the-art accuracy among transformer-based methods with better robustness to noise, better time efficiency, and more economic computation requirements.

Our contributions are three-fold. First, we propose a visual hierarchy for more discriminative visual representations without additional annotations. Second, we propose an attention-based module to leverage the hierarchical features. Last, we build a transformer-based tracker with these two innovations and demonstrate its superior accuracy and time efficiency in a pure appearance-based fashion for multi-object tracking.

II. Related Works

Deep Visual Representation. We typically use a backbone network to extract features from a certain area, such as bounding boxes, as a visual representation for visual perception. However, the bounding box is noisy as it always contains pixels from the background or other object instances. For a more fine-grained visual representation, a common way is to use pre-defined regions, such as human head [36],

[31] or human joints [2], [44]. However, these choices require additional data annotations and specified perception modules. Without requiring additional annotations, multi-region CNN [16] proposes to stack the features from bounding box bins to build a compositional visual representation. However, this paradigm can not generate instance-level discriminative representation though it shows effectiveness in semantic-level recognition. Moreover, simply stacking features can't emphasize the discriminative visual regions.

Hierarchy Visual Representations. The term "hierarchical visual representations" has been used indiscriminately for (1) features fused from different resolutions of the same area, such as CNN feature pyramid [24], [20] and (2) features fused from different pixel areas. Our proposed hierarchical visual representations lie in the second genre. Our idea is inspired by David Marr's hierarchical modeling of the human body [26] (*computational*, *algorithmic*, and *implementational*) and the visual cognitive hierarchy [13] (*semantic*, *syntactic*, *physical*). Compared to the two visual hierarchies, the three-level hierarchy we propose (*compositional*, *semantic*, *contextual*) is focused on building discriminative visual representations for multi-object tracking. Also, in the area of re-identification, some previous works leverage part-based hierarchical features to build visual representation. But most of them typically require additional annotations for body parts [32]. The way they fuse the features from different regions [14] is not effective in multi-object tracking cases where the background noise in the target bounding box area is usually more severe with fast-moving targets and non-static cameras.

Query-based Multi-Object Tracking. Transformer [39] is introduced to visual perception [7] after its original application in natural language processing. Later, query-based multi-object tracking methods were proposed. The early methods [35], [27] associate objects locally on adjacent time steps. Some recent methods associate targets globally in a video clip [55], [49]. GTR [55] removes secondary modules such as positional encoding, making a clean baseline to evaluate feature discriminativeness. Most recent methods improve performance by gathering information over a long period [4], [49]. However, a downside is the high requirement of computation resources, e.g., 8xA100 GPUs [4]. Instead, the improvement of our method comes from the proposed hierarchical representation. We demonstrate its state-of-the-art effectiveness and efficiency among query-based methods.

III. Method

In this section, we first introduce the overall architecture of CSC-Tracker . Then we describe the proposed CSC-Attention module to fuse the features from the visual hierarchy. Finally, we elaborate on the training and inference of CSC-Tracker .

A. Overall Architecture

We follow the spatio-temporal global association paradigm [42], [55] to build CSC-Tracker , whose pipeline is shown in Figure 2. Now, we explain the three stages of it. Notations are conditional to a generic time step t, which is the last time step where the tracks have been finalized.

Detection and Feature Extraction. Given a video clip of T frames, i.e., $\mathcal{T} = \{t+1, ..., t+T\}$, we have the corresponding images $\mathcal{I} = \{I^{t+1}, ..., I^{t+T}\}$. Given a detector, we could derive the detections of the objects of interest on all frames in parallel, noted as $\mathcal{O} = \{O_1, ..., O_{N_t}\}$. N_t is the number of detections and $t_i \in \mathcal{T}$ $(1 \leq i \leq N_t)$ is the time step where the i-th detection, i.e., O_i, is detected. Then, we extract the features of each detected object by a backbone network.

Token Generation by CSC-Attention. We propose CSC-Attention (to be detailed in the following section) to generate feature tokens. By CSC-Attention, we will have the object CSC-tokens $\mathcal{Q}_t^{\text{det}} \in \mathbb{R}^{N_t \times D}$, where D is the feature dimension. If we aim to associate the new-coming detections with existing trajectories, we also need the tokens to represent the existing M_t trajectories, i.e., $\mathbf{T}_t^{\text{traj}} = \{Tk_1^{\text{traj}}, Tk_2^{\text{traj}}, ..., Tk_{M_t}^{\text{traj}}\}$. Instead of the resource-intensive iterative query passing [49] or long-time feature buffering [4], we leverage the CSC-tokens of objects on a trajectory to represent it. Within a horizon H, we represent a trajectory, Tk_j^{traj}, with the token $Q_j^{\text{traj}} \in \mathbb{R}^{H \times D}$ by combining the historical detection CSC-tokens. And all trajectory tokens are $\mathcal{Q}_t^{\text{traj}} = \{Q_1^{\text{traj}}, ..., Q_{M_t}^{\text{traj}}\}$.

Global Association. By cross-attention, we could get the association score between the set of detections and a trajectory, i.e. Tk_j^{traj}, as $S(Q_j^{\text{traj}}, \mathcal{Q}_t^{\text{det}}) \in \mathbb{R}^{H \times N_t}$. In practice, because we aim to associate between all M_t trajectories and N_t detections, we perform the cross-attention on all object queries and track queries at the same time, namely $S(\mathcal{Q}_t^{\text{traj}}, \mathcal{Q}_t^{\text{det}}) \in \mathbb{R}^{HM_t \times N_t}$. By averaging the score on the H steps in the horizon, we get the global association score $\mathbf{S}^t \in \mathbb{R}^{M_t \times N_t}$. Then, we normalize the association scores between a trajectory and objects from the same time step by softmax:

$$P(\mathbf{M}_{j,i}^t = 1 | \mathcal{Q}_t^{\text{det}}, \mathcal{Q}_t^{\text{traj}}) = \frac{\exp(\mathbf{S}_{j,i}^t)}{\sum_{k \in \{1,2,...,N_t\}} \mathbf{1}_{[t_k = t_i]} \exp(\mathbf{S}_{j,k}^t)}, \tag{1}$$

where the binary indicator function $\mathbf{1}_{[t_k = t_i]}$ indicates whether the i-th detection and the k-th detection are on the same time step. $\mathbf{M}^t \in \mathbb{R}^{(M_t+1) \times N_t}$ is the final global association matrix. Its dimension is of $(M_t + 1) \times N_t$ because each detection can be associated with an "empty trajectory" to start a new track. The query of the "empty trajectory" is represented by a token randomly drawn from a previous unassociated object. Also, after the association, unassociated trajectories will be considered absent on the corresponding frames. In such a fashion, we can train over a large set of detections and trajectories in parallel and conduct inference online by a sliding window. We use a uniform form for queries to represent both objects and trajectories. Thus, the global association can happen either among detections or between detections and trajectories. These two schemes of associations thus are implemented as the same and share all model modules. For online inference, we associate detections from the new-coming time step $(T = 1)$ and existing trajectories.

B. CSC-Attention

Now, we explain the attention mechanism to fuse the features from the ***C**ompositional-**S**emantic-**C**ontextual* visual hierarchy. We name it CSC-Attention (right-half of Fig. 2).

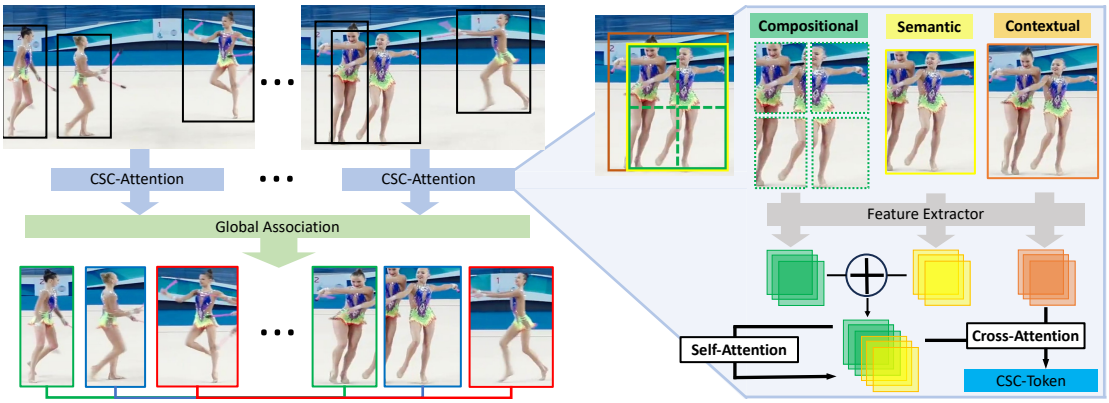

Fig. 2: The architecture of CSC-Tracker . The left half illustrates the overall architecture. The right half is the zoomed-in CSC-Attention module. Our contributions are (1) the visual hierarchy for feature extraction and (2) the CSC-Attention module for feature fusion.

Hierarchy Construction. There are different choices for constructing the hierarchy. To have a fair comparison with a close baseline [16], we use bounding box bins to represent object parts. Given a detection O, we divide the bounding box into 2×2 bins (to fit in GPU memory), making a set of body parts as $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$. On the other hand, from a global scope, there are other targets interacting with O which are highly likely to be mismatched in the association stage. We crop the union area enclosing O and all other targets having overlap with it. We note the union area as U. Till now, we have derived the triplet $\{\mathcal{P}, O, U\}$ as the raw material for the visual hierarchy.

Feature Fusion. Among the three levels, semantic information is necessary to define a visual boundary. Compositional and contextual cues serve as the enhancement to the final representation's discriminativeness. With the extracted regions $\{\mathcal{P}, O, U\}$, we use a shared feature extractor to get their features, i.e. compositional, semantic, and contextual features. To fuse the features, we first concatenate the compositional and semantic features. Then a self-attention module is applied to help attend to the discriminative regions. Finally, the contextual features and the self-attention output are processed by a cross-attention module to get the final CSC-tokens. Before being forwarded to the global association, the tokens would be projected to a uniform dimension of D.

C. Training and Inference

Training. We train the association module by maximizing the likelihood of associating detections belonging to the same trajectory as in Eq. 1. We calculate the association score on all T frames of the sampled video clip simultaneously and globally. The objective thus turns to

$$\max \prod_{q=t+1}^{t+T} P(\mathbf{M}^t_{j,\tau^j_q} = 1 | \mathcal{Q}^{\text{det}}_t, \mathcal{Q}^{\text{traj}}_t), \tag{2}$$

where τ^j_q is the ground truth index of the detection to be associated with the j-th trajectory on the q-th time step. By applying the objective to all trajectories, the training loss is

$$L_{\text{asso}} = -\sum_{j=1}^{M_t+1} \sum_{q=t+1}^{t+T} \log P(\mathbf{M}^t_{j,\tau^j_q} = 1 | \mathcal{Q}^{\text{det}}_t, \mathcal{Q}^{\text{traj}}_t). \tag{3}$$

On the other hand, trajectories can also be absent on some time steps because of occlusion or target disappearance. Therefore, Eq. 3 has included the situation of associating a trajectory with no detection, i.e. "empty". The token for an empty detection is an arbitrary negative sample. We also have a triplet loss to pull away the feature distance between negative pairs compared to that between positive pairs:

$$\begin{aligned} L_{\text{feat}} = \max(0, \min_{u=1}^{N_P} &||\text{Att}(f(F_{p_u}), f(F_O)) - f(F_O)||^2 - \\ &||\text{Att}(f(F_O), f(F^{bg}_U)) - f(F_O)||^2 + \alpha), \end{aligned} \tag{4}$$

where $f(\cdot)$ is the shared layers to project CNN features and $N_{\mathcal{P}}$ is the number of part patches ($N_{\mathcal{P}} = 4$ in our default setting). $\text{Att}(\cdot, \cdot)$ is the operation of cross attention. α is the margin to control the distance between positive and negative pairs. F_O and F_{p_u} $(1 \le u \le N_{\mathcal{P}})$ are the semantic and compositional features. F^{bg}_U is the features of the background area in the union area U. We obtain the background features by setting the pixels of O in the area of U to 0 and forward the masked union area into the shared feature encoder $f(\cdot)$. We design Eq. 4 to encourage (1) the feature encoder to pay more attention to the salient and distinct area on targets while less attention to the background area and (2) the features of the background area in the union box to be discriminative from the foreground object. Finally, the training objective is

$$L = L_{\text{asso}} + L_{\text{feat}} + L_{\text{det}}, \tag{5}$$

where L_{det} is an optional detection loss.

Inference. We realize online inference by traversing the video with a sliding window of stride 1. On the first frame, each detection initializes a trajectory. By averaging the detection-detection association score alongside a trajectory, we get the detection-trajectory association scores, whose negative value serves as the entries in the cost matrix for the association assignment. We adopt Hungarian matching to ensure one-to-one mapping. Only when the association score is higher than $\beta = 0.3$, the pair can be associated. All unassociated detections on the new-coming frames will start new tracks.

IV. Experiments

A. Experiment Setups

Datasets. We focus on pedestrian tracking in this paper as it is the most popular scenario and a line of previous works is available for comparison of association accuracy. On some other tracking datasets, such as TAO [10], tracking faces main

TABLE I: Results on MOT17 and MOT20 test sets with the private detections (FP and FN reported by $\times 10^4$).

Tracker	HOTA↑	AssA↑	MOTA↑	IDF1↑	FP↓	FN↓	IDs↓
MOT-17 Test							
FairMOT [51]	59.3	58.0	73.7	72.3	2.75	11.7	3,303
Semi-TCL [18]	59.8	59.4	73.3	73.2	2.29	12.5	2,790
CSTrack [19]	59.3	57.9	74.9	72.6	2.38	11.4	3,567
GRTU [40]	62.0	62.1	74.9	75.0	3.20	10.8	1,812
QDTrack [29]	53.9	52.7	68.7	66.3	2.66	14.7	3,378
MAA [33]	62.0	60.2	79.4	75.9	3.73	7.77	1,452
ReMOT [47]	59.7	57.1	77.0	72.0	3.32	9.36	2,853
PermaTr [38]	55.5	53.1	73.8	68.9	2.90	11.5	3,699
ByteTrack [50]	63.1	62.0	80.3	77.3	2.55	8.37	2,196
DST-Tracker [6]	60.1	62.1	75.2	72.3	2.42	11.0	2,729
UniCorn [46]	61.7	-	77.2	75.5	5.01	7.33	5,379
OC-SORT [5]	63.2	63.2	78.0	77.5	1.51	10.8	1,950
Deep OC-SORT [25]	64.9	65.9	79.4	80.6	1.66	9.88	1,023
MotionTrack [30]	65.1	65.1	81.1	80.1	2.38	8.17	1,140
SUSHI [8]	66.5	67.8	81.1	83.1	3.23	7.32	1,149
TransCt [45]	54.5	49.7	73.2	62.2	2.31	12.4	4,614
TransTrk [35]	54.1	47.9	75.2	63.5	5.02	**8.64**	3,603
MOTR [49]	57.2	55.8	71.9	68.4	**2.11**	13.6	**2,115**
TrackFormer [27]	-	-	65.0	63.9	7.44	12.4	3,528
GTR [55]	59.1	57.0	75.3	75.1	2.68	10.9	2,859
MeMOT [4]	56.9	55.2	72.5	69.0	3,72	11.5	2,724
CSC-Tracker	**60.8**	**60.7**	**75.4**	**75.7**	2,45	10,8	2,879
MOT-20 Test							
FairMOT [51]	54.6	54.7	61.8	67.3	10.3	8.89	5,243
CSTrack [19]	54.0	54.0	66.6	68.6	2.54	14.4	3,196
GSDT [41]	53.6	52.7	67.1	67.5	3.19	13.5	3,131
RelationT [48]	56.5	55.8	67.2	70.5	6.11	10.5	4,243
MAA [33]	57.3	55.1	73.9	71.2	2.49	10.9	1,331
ByteTrack [50]	61.3	59.6	77.8	75.2	2.62	8.76	1,223
OC-SORT [5]	62.1	62.0	75.5	75.9	1.80	10.8	913
Deep OC-SORT [25]	63.9	65.7	75.6	79.2	1.69	10.8	779
MotionTrack [30]	62.8	61.8	78.0	76.5	2.86	8.42	1,165
TransCt [45]	43.5	37.0	58.5	49.6	6.42	14.6	4,695
TransTrk [35]	48.5	45.2	65.0	59.4	**2.72**	15.0	3,608
MeMOT [4]	**54.1**	**55.0**	63.7	**66.1**	4,79	13.8	**1,938**
CSC-Tracker	53.0	51.1	**65.8**	64.4	3.64	**13.7**	3,948

difficulties at the detection stage instead of association. This causes uncontrollable noise to evaluate how discriminative the features are. For valid evaluation of visual representation distinguishness, we select three datasets, i.e., MOT17 [28], MOT20 [11] and DanceTrack [34]. DanceTrack has the largest data scale and provides an official validation set. DanceTrack contains targets mostly in the foreground but with heavy occlusion, complex motion patterns, and similar appearances. On DanceTrack, detection is not considered as the bottleneck and the model ability of appearance discrimination becomes the key for tracking.

Evaluation Metrics. The CLEAR evaluation protocol [3] is popular for multi-object tracking evaluation but is biased to single-frame association quality [23]. MOTA is the main metric of CLEAR [3] protocol. But it is also biased to the detection quality. To provide a more accurate sense of association accuracy, we emphasize the recent HOTA [23] metric set where the metric is calculated upon the video-level association between ground truth and predictions (by default in the form of bounding boxes). In the set of metrics, AssA emphasizes the association performance, and DetA stresses the detection quality. HOTA is the main metric by considering both detection and association quality. For the result tables, we use underlined numbers to indicate the overall best value and **bold** numbers for the best query-based methods. All query-based methods are listed in blue.

Implementation. We use ResNet-50 [17] as the backbone network, which is pretrained on Crowdhuman [31] dataset first. Though advanced detector [50] is demonstrated as a key to boosting tracking performance, we want our contribution to be more from the improvement of the association stage. Therefore, on MOT17, we align the implementation with the practice of GTR [55] to use the classic CenterNet [54], [53] as the detector to make a fair comparison. The CenterNet detector is pretrained together with the backbone on Crowdhuman. For the fine-tuning of association modules on MOT17, we use a 1:1 mixture of MOT17-train and Crowdhuman. We fine-tune with only the MOT20-train for evaluation on MOT20. For DanceTrack, we use its official training set as the only training set during finetuning. The image size is set to be 1280 $\times$ 1280 during training. The image size is 1560 for the longer edge during the test. During finetuning, the detector head is also finetuned. The training iterations are set to be 20k on MOT17/MOT20 and 80k on DanceTrack. We use BiFPN [37] for the feature upsampling. For the implementation of the transformer, we use a stack of two layers of "Linear + ReLU" as the projection layers and one-layer encoders and decoders. We use AdamW [22] optimizer for training whose base learning rate is set to be 5e-5. The length of the video clip is $T = 8$ for training and $T = 24$ for inference in a sliding window for a fair comparison with GTR [55]. We use 4 $\times$ V100 GPUs as the default training device but we will see that even using only one RTX 3090 GPU for training, our method still achieves comparable performance. The training takes 4 hours on MOT17 or MOT20 and 11 hours on DanceTrack.

TABLE II: Benchmarking results on DanceTrack test set.

Tracker	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
CenterTrack [53]	41.8	78.1	22.6	86.8	35.7
FairMOT [51]	39.7	66.7	23.8	82.2	40.8
QDTrack [29]	45.7	72.1	29.2	83.0	44.8
TraDes [43]	43.3	74.5	25.4	86.2	41.2
ByteTrack [50]	47.3	71.6	31.4	89.5	52.5
OC-SORT [5]	55.7	81.7	38.3	92.0	54.6
Deep OC-SORT [25]	61.3	82.2	45.8	92.3	61.5
DST-Tracker [6]	51.9	72.3	34.6	84.9	51.0
SUSHI [8]	63.3	80.1	50.1	88.7	63.4
TransTrk[35]	45.5	75.9	27.5	88.4	45.2
MOTR [49]	54.2	73.5	40.2	79.7	51.5
GTR [55]	48.0	72.5	31.9	84.7	50.3
CSC-Tracker (Ours)	**55.5**	**77.3**	**43.1**	**89.5**	**54.0**

B. Benchmark Results

For benchmarking, we only report the performance of online tracking algorithms as offline post-processing [12], [52] gives unfair advantages and blurs the discussion about visual representation discriminativeness. We first benchmark on MOT17 and MOT20 in Table I. On MOT17, CSC-Tracker achieves the highest HOTA and AssA score among transformer-based methods. MOT20 is a more challenging dataset with crowded pedestrian flows. Though CSC-Tracker shows better performance than MeMOT [4] on MOT17, its performance is inferior on MOT20. This is probably related to the long-time heavy and frequent occlusion on MOT20. To solve this problem, the long temporal buffer of historical object appearance in MeMOT shows effectiveness. However, MeMOT requires 8$\times$A100 GPUs for training to support such a long buffering (22 frames v.s. 8 frames by CSC-Tracker

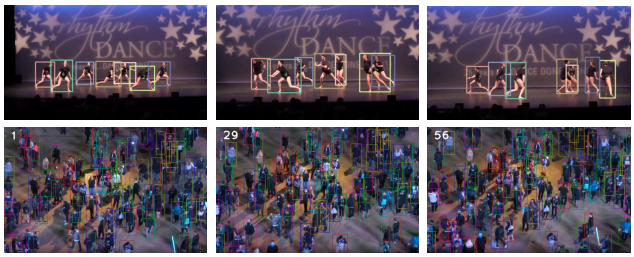

Fig. 3: Upper line: Results from DanceTrack-test set where targets have occlusion, crossover and similar appearance. Bottom line: Results on a MOT20-test video where the pedestrians are in the crowd and heavily occluded.

) and uses COCO [21] dataset as the additional pretraining data, which makes it not an apple-to-apple comparison.

We also benchmark on DanceTrack-test in Table II. CSC-Tracker achieves state-of-the-art performance among transformer-based methods. Also, CSC-Tracker shows advanced time efficiency. For example, training on MOT17 takes MOTR [49] 2.5 days on 8×V100 GPUs while only 4 hours on 4×V100 GPUs for our proposed method. The inference speed is 6.3FPS for MOTR while 21.3FPS for our method on the same machine (V100 GPU). Compared to GTR [55], CSC-Tracker achieves a more significant outperforming on DanceTrack than on MOT17. As other variables and design choices are strictly controlled, it suggests our proposed visual hierarchy representation is more powerful than the naive bounding box features when the occlusion is heavier.

Given the aforementioned results, we have demonstrated CSC-Tracker to be the state-of-the-art among transformer-based methods with a lightweight design. More importantly, we show that the proposed hierarchical representation is more effective and efficient in discriminatively distinguishing objects. CSC-Tracker builds a new baseline for future research in this line of methods. The commonly adopted techniques of query propagation and iteration [27], [35], [49], deformable attention [35], [4] and long-time feature buffering [4] are all compatible to be integrated with CSC-Tracker . Compared to the overall state-of-the-art methods, such as OC-SORT [5] and SUSHI [8], CSC-Tracker still shows inferior performance. But their performance is reported with a more advanced detector, i.e. YOLOX [15]. This makes a fair comparison hard to present. But still, there is a performance gap between the SOTAs and the transformer-based methods. For inference speed, given detections on MOT17, OC-SORT runs at 300FPS and SUSHI runs at 21FPS while CSC-Tracker runs at 93FPS.

C. Ablation Study

We now ablate the contribution of key variables in the design and implementation to the performance of CSC-Tracker . Many previous works in the multi-object tracking community follow the practice of CenterTrack [53] on MOT17 [28] to use the latter half of training video sequences as the validation set. However, this makes the ablation study on the validation set not fair because the data distribution of the training set and validation set is so close that the performance gap reflected on the validation set might degrade or even disappear on the test set. Therefore, we turn to DanceTrack [34] for the ablation study as an independent validation set is provided. For the following tables, we highlight our default implementation choice in yellow , which corresponds to the entries previously reported on benchmarks to compare with other methods.

TABLE III: Ablation of video clip length for training.

T	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
6	51.0	70.7	33.4	81.4	51.4
8	51.9	71.4	34.0	81.9	**52.2**
10	52.4	71.7	34.5	81.8	51.4
12	**52.6**	**71.9**	**34.7**	**82.0**	51.7

TABLE IV: Ablation of video clip length for Inference.

T	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
8	50.2	70.7	32.9	81.1	51.2
16	51.6	71.2	33.6	81.5	51.7
24	**51.9**	**71.4**	**34.0**	81.9	**52.2**
32	51.7	71.2	33.9	**82.0**	51.9

Video Length. Table III and IV show the influence of video clip length in the training and inference stages respectively. The result suggests that training the association model with longer video clips can continuously improve performance. Limited by the GPU memory, we cannot increase the video clip length to longer than 12 frames here. On the contrary, during the inference stage, the sliding window size does not have a significant impact on the performance. Increasing the window size beyond a plateau will even hurt the performance.

Three levels in CSC-hierarchy. We study the contribution of each level of the CSC hierarchy in Table V. Here, only the semantic information is necessary for the evaluation with bounding box-based ground truth annotations and we can manipulate the other two levels in the CSC-hierarchy by not adding the corresponding feature in the generation of the CSC-Tokens. Here we note that adding the compositional and contextual features only brings subtle computation overhead as the required self-attention and cross-attention operation are highly in parallel. Compared to only using the *semantic* feature, CSC-Tracker achieves a significant performance improvement indicated by higher HOTA and AssA scores. Also, integrating the features of the union area shows better effectiveness than solely integrating the features of body parts. This is probably because the cross attention between object body and union areas can provide critical information to compare object targets with their neighboring objects, preventing potential mismatch. On the other hand, integrating the body part features can't explicitly avoid the mismatch with other instances. Fusing the features from all the levels turns out the best choice.

Input size. We try different parameter configurations in Table VI for the input clip length and image size. With only a single RTX 3090 GPU for training and inference, its performance is still comparable to the default setting with 4 × V100 GPUs. This makes the notorious computation barrier of transformer-based methods not that terrible anymore.

Detector. The highest priority for experiments is to validate the effectiveness of our proposed representations instead of racing on the leaderboard. For a fair comparison with the closest baseline GTR [55], we follow it to choose CenterNet [54] as the default detector. But CSC-Tracker

TABLE V: The ablation study about the contribution from *semantic*, *compositional*, and *contextual* features.

Semantic	Compo.	Context.	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
✓			47.8	69.1	30.1	80.8	49.1
✓	✓		49.6	69.3	31.3	81.2	50.4
✓		✓	50.5	70.6	32.6	81.5	51.2
✓	✓	✓	51.9	71.4	34.0	81.9	52.2

TABLE VI: Different implementation choices to fit multiple training device configurations.

Training Device	Train_len	Image Size	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
1x RTX 3090-24GB	6	1280 × 1280	50.9	71.0	33.3	81.3	51.2
1x V100-32GB	8	1560 × 1560	51.2	71.7	33.7	82.0	52.0
4x V100-32GB	8	1280 × 1280	51.9	71.4	34.0	81.9	52.2

TABLE VII: Ablation of detector models.

Detector	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
CenterNet	51.9	71.4	34.0	81.9	52.2
YOLOv4 [1]	52.6	73.8	34.5	84.0	53.4
YOLOX [15]	**53.5**	**74.7**	**35.1**	**85.1**	**54.7**

TABLE VIII: Ablation about feature fusion strategies.

Method	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑
Bbox only	47.8	69.1	30.1	80.8	49.1
Multi-Region CNN[16]	47.4	69.5	29.5	80.8	48.6
CSC-Attention	**51.9**	**71.4**	**34.0**	**81.9**	**52.2**

is a tracking-by-detection method, flexible to integrate with different detectors. We compare CenterNet with the other detectors, i.e., YOLOv4 [1] and YOLOX [15] (used by ByteTrack, OC-SORT, SUSHI, etc.) in Table VII. Advanced detectors can boost tracking performance.

Fusion strategy of hierarchical features. As a main contribution of this paper, we propose CSC-Attention module to fuse the features from the CSC-hierarchy. In a naive fashion, the multi-region CNN applies a *split-and-concatenate* strategy to fuse the features from different bins inside a bounding box. We conduct a comparison with the multi-region CNN [16] in Table VIII. Though multi-region CNN achieves improvement over the naive bounding box representation for object detection, this advantage is not observed anymore for multi-object tracking. Its performance gap with the features fused by CSC-Attention is even more significant than solely using the bounding box. This experiment suggests the effectiveness of the proposed three-level hierarchy and fusing them with the proposed CSC-Attention module.

D. Robustness to Detection Noise

With the enforcement of the part region (compositional) features, we expect CSC-Tracker to show better robustness to the noise in detections. The intuition is that even if the bounding box is not accurate, as long as a distinct part is recognized, the model should be able to track an object consistently. To validate it, we add noise to the detection positions and observe its influence on the tracking performance. We apply random shifting and random resizing to add noise. For random shifting, we have a 25% chance to shift the bounding box to the four directions independently, the shift stride is a random value in the range of $[0, \min(0.2d, 20)]$, where d is the bounding box width or height. We resize the bounding box width or height independently with a ratio of α_w and α_h, both of which are random values in the range of [0.9, 1.1]. The results on Dancetrack-val are shown in Table IX. Compared to the motion-based baseline OC-SORT and the full-box-only baseline GTR, CSC-Tracker shows better robustness to the noise of detections as expected.

TABLE IX: Effect of detection noise (* indicates adding noise).

Method	HOTA↑	AssA↑	IDF1↑
OC-SORT [5]	52.1	35.3	51.6
OC-SORT*	49.5 (↓ 2.6)	31.3 (↓ 4.0)	48.5 (↓ 3.1)
GTR [55]	47.2	28.2	47.0
GTR*	45.0 (↓ 2.2)	26.7 (↓ 1.5)	45.6 (↓ 1.4)
CSC-Tracker	51.9	34.0	52.2
CSC-Tracker *	50.8 (↓ 1.1)	33.2 (↓ 0.8)	51.5 (↓ 0.7)

TABLE X: Time efficiency (MOT17-test).

Method	HOTA	training time	inference speed
Transtrack [35]	54.1	18 hrs	10FPS
Trackformer [27]	-	-	7.4FPS
MOTR [49]	57.2	63 hrs	6.5FPS
TransCenter [45]	54.5	-	11FPS
GTR [55]	59.1	4 hrs	22.4FPS
CSC-Tracker	60.8	4 hrs	21.3FPS

E. Time Efficiency

Time efficiency is a bottleneck of query-based methods, especially for those using graph network [9], long-history buffers [4] or temporal aggregation [49]. Collecting the methods that report the time efficiency or have open-sourced implementation, we report the required training time and inference speed in Table X by default settings on MOT17. The speed is tested on Nvidia V100 GPU and the training time is evaluated on 4xV100 GPUs. CSC-Tracker achieves the best accuracy with one of the best time efficiency for both training time and the inference speed.

V. Conclusion

In this paper, we propose to construct discriminative visual representations by a *compositional-semantic-contextual* visual hierarchy combining different visual cues to distinguish a target. To leverage them comprehensively, we propose a CSC-Attention to gather and fuse the visual features. These are the two main contributions of this paper. We have demonstrated that they are connected to show power. The designs are integrated into CSC-Tracker for multi-object tracking. The results on multiple datasets demonstrate its efficiency and effectiveness. We hope the study of this paper can provide new knowledge in the visual representation of objects and an advanced baseline model to solve multi-object tracking problems. The method is also more robust to the detection noises and computation-economic.

REFERENCES

[1] H.-Y. M. L. Alexey Bochkovskiy, Chien-Yao Wang, "Yolov4: Yolov4: Optimal speed and accuracy of object detection," *arXiv*, 2020.

[2] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "Posetrack: A benchmark for human pose estimation and tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5167–5176.

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

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

[5] J. Cao, X. Weng, R. Khirodkar, J. Pang, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," *arXiv preprint arXiv:2203.14360*, 2022.

[6] J. Cao, H. Wu, and K. Kitani, "Track targets by dense spatio-temporal position encoding," *arXiv preprint arXiv:2210.09455*, 2022.

[7] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[8] O. Cetintas, G. Brasó, and L. Leal-Taixé, "Unifying short and long-term tracking with graph hierarchies," *arXiv preprint arXiv:2212.03038*, 2022.

[9] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "Transmot: Spatial-temporal graph transformer for multiple object tracking," *arXiv preprint arXiv:2104.00194*, 2021.

[10] A. Dave, T. Khurana, P. Tokmakov, C. Schmid, and D. Ramanan, "Tao: A large-scale benchmark for tracking any object," in *European conference on computer vision*. Springer, 2020, pp. 436–454.

[11] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," *arXiv preprint arXiv:2003.09003*, 2020.

[12] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Transactions on Multimedia*, 2023.

[13] J. A. Fodor and Z. W. Pylyshyn, "Connectionism and cognitive architecture: A critical analysis," *Cognition*, vol. 28, no. 1-2, pp. 3–71, 1988.

[14] Y. Fu, Y. Wei, Y. Zhou, H. Shi, G. Huang, X. Wang, Z. Yao, and T. Huang, "Horizontal pyramid matching for person re-identification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8295–8302.

[15] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[16] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1134–1142.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] W. Li, Y. Xiong, S. Yang, M. Xu, Y. Wang, and W. Xia, "Semi-tcl: Semi-supervised track contrastive representation learning," *arXiv preprint arXiv:2107.02396*, 2021.

[19] C. Liang, Z. Zhang, Y. Lu, X. Zhou, B. Li, X. Ye, and J. Zou, "Rethinking the competition between detection and reid in multi-object tracking," *arXiv preprint arXiv:2010.12138*, 2020.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[22] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[23] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, no. 2, pp. 548–578, 2021.

[24] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Robust visual tracking via hierarchical convolutional features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2709–2723, 2018.

[25] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification," *arXiv preprint arXiv:2302.11813*, 2023.

[26] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[27] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," *arXiv preprint arXiv:2101.02702*, 2021.

[28] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[29] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 164–173.

[30] Z. Qin, S. Zhou, L. Wang, J. Duan, G. Hua, and W. Tang, "Motiontrack: Learning robust short-term and long-term motions for multi-object tracking," *arXiv preprint arXiv:2303.10404*, 2023.

[31] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," *arXiv preprint arXiv:1805.00123*, 2018.

[32] V. Somers, C. De Vleeschouwer, and A. Alahi, "Body part-based representation learning for occluded person re-identification," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 1613–1623.

[33] D. Stadler and J. Beyerer, "Modelling ambiguous assignments for multi-person tracking in crowds," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 133–142.

[34] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani, and P. Luo, "Dancetrack: Multi-object tracking in uniform appearance and diverse motion," *arXiv preprint arXiv:2111.14690*, 2021.

[35] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[36] R. Sundararaman, C. De Almeida Braga, E. Marchand, and J. Pettre, "Tracking pedestrian heads in dense crowd," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3865–3875.

[37] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.

[38] P. Tokmakov, J. Li, W. Burgard, and A. Gaidon, "Learning to track with object permanence," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 860–10 869.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[40] S. Wang, H. Sheng, Y. Zhang, Y. Wu, and Z. Xiong, "A general recurrent tracking framework without real data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 219–13 228.

[41] Y. Wang, K. Kitani, and X. Weng, "Joint object detection and multi-object tracking with graph neural networks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 708–13 715.

[42] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8741–8750.

[43] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 352–12 361.

[44] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose flow: Efficient online pose tracking," *arXiv preprint arXiv:1802.00977*, 2018.

[45] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "Transcenter: Transformers with dense queries for multiple-object tracking," *arXiv preprint arXiv:2103.15145*, 2021.

[46] B. Yan, Y. Jiang, P. Sun, D. Wang, Z. Yuan, P. Luo, and H. Lu, "Towards grand unification of object tracking," in *ECCV*, 2022.

[47] F. Yang, X. Chang, S. Sakti, Y. Wu, and S. Nakamura, "Remot: A model-agnostic refinement for multiple object tracking," *Image and Vision Computing*, vol. 106, p. 104091, 2021.
[48] E. Yu, Z. Li, S. Han, and H. Wang, "Relationtrack: Relation-aware multiple object tracking with decoupled representation," *arXiv preprint arXiv:2105.04322*, 2021.
[49] F. Zeng, B. Dong, T. Wang, X. Zhang, and Y. Wei, "Motr: End-to-end multiple-object tracking with transformer," *arXiv preprint arXiv:2105.03247*, 2021.
[50] Y. Zhang, P. Sun, Y. Jiang, D. Yu, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," *arXiv preprint arXiv:2110.06864*, 2021.
[51] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021.
[52] Y. Zhang, T. Wang, and X. Zhang, "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*, 2023, pp. 22 056–22 065.
[53] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *European Conference on Computer Vision*. Springer, 2020, pp. 474–490.
[54] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.
[55] X. Zhou, T. Yin, V. Koltun, and P. Krähenbühl, "Global tracking transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8771–8780.